RECENT TRENDS IN NETWORK SECURITY SYSTEM

Bhavana A.
Aishwary Awasthi

ALEXIS PRESS
JERSEY CITY, USA

RECENT TRENDS IN
NETWORK SECURITY SYSTEM

RECENT TRENDS IN NETWORK SECURITY SYSTEM

Bhavana A.

Aishwary Awasthi

ALEXIS PRESS

Published by: Alexis Press, LLC, Jersey City, USA
www.alexispress.us

First Published 2022

A catalogue record for this publication is available from the British Library

Library of Congress Cataloguing in Publication Data

Includes bibliographical references and index.

Recent Trends in Network Security System by *Bhavana A., Aishwary Awasthi*

ISBN 978-1-64532-883-4

CONTENTS

CHAPTER 1

NETWORK SECURITY ERROR IDENTIFICATION AND TRACKING

Ms. Bhavana A., Assistant Professor
Department of Computer Science and Engineering, Presidency University, Bangalore, India
Email Id- bhavana@presidencyuniversity.in

Essential components must be present for all forms of communication to be successful. First, a transmitter and a receiver two distinct entities must exist. There must be a necessity for sharing between these two. Second, a route over which the shareable object is sent must exist. The communication network is finally, a set of procedures or guidelines for interaction that must be established these three holds for all interaction types and structures. We will concentrate on these three elements of a computer network in this chapter. The viewer must be aware that from this point forward when we use the term "computer network," we mean the conventional computer network.

We will concentrate on these three elements of a computer network in this chapter. A computer network is what? The viewer must be aware that from this point forward when we use the word "computer network," we mean the conventional computer network. A distributed system made up of loosely linked machines and other devices is referred to as a computer network. Any two of these gadgets can interact with each other over a communication medium, and from this point forward we shall refer to them without losing generality as system components or transmission elements. There should be a set of communication guidelines or protocols that each connected device must adhere to for the group of linked devices to be regarded as a communicative network.

The network component of the hardware consists of a collection of nodes, or end systems, also known as hosts, and intermediary switching devices, such as connectors, overpasses, tunnels, and gateways. We'll refer to these as network elements to avoid generality loss. Resources can be owned by network parts locally or globally. All application programs and internet protocols that are used to synchronize, coordinate, and facilitate data sharing and exchange across network nodes are collectively referred to as network software. Sharing of expensive network resources is also made feasible by network software network components, and network applications.

Users collaborate so that ordinary users may communicate with one another and share resources that are not easily accessible locally on other platforms. The network components and their resources might use a variety of hardware technologies and the program may be as dissimilar as possible, yet the entire system must function as a single unit. Through the use of internetworking technology, heterogeneous networks may be seamlessly connected with a variety of underlying hardware and software regimes. Any computer communication network may operate without hiccups thanks to the low-level mechanisms given by the network components and the high-level communication features offered by the software that runs on the communicating components. Let's first glance at the various forms of networks before we explore how they function.

Computer Network Models:

The configuration models that make up a computer network are numerous. The centralized and decentralized models are the most prevalent among them[1]. In a centralized model, several

gadgets and computers are linked together and have communication capabilities all correspondence must, however, be routed through the master, a solitary central computer. Surrogates, or dependent computers, may have fewer local resources like memory, and the master in the center has authority over shareable global resources. The distributed network, in contrast to the centralized system, is made up of interconnected computers that are connected via a communication network made up of bridging components and communication channels. The computers themselves could be the owners of their local resources or they might ask another computer for resources[2], [3].

These devices are referred to by a variety of names, such as hosting, client, or node. A host is referred to as a server if it contains resources that the other hosts require. Exchanging information and resources are placed between any two interacting network nodes and are not managed by the main computer logically centralized model and a wireless mesh model.

Types of Network Security:

There are several sizes for computer networks. Every network consists of a collection of network resources and network components. The cluster's size influences the network's architecture. The local area network (LAN) and the wide area network (WAN) are the two basic types of networks large-scale networks (WAN).

1. **Local Area Networks (LANs)**

A computer network consists of several computers or network clusters and their resources that are restricted in a limited physical area, like a building floor, sharing communication protocols. A local neighborhood network is a group of buildings that are close to one another (LAN). Because all network components of a LAN are close to one another, data may travel more quickly via the communication channels. Additionally, because the communication pieces are close together, expensive, high-quality communicating elements may be employed to provide greater service and higher dependability.

2. **Wide Area Networks (WANs)**

On the other hand, a wide area network (WAN) is a system that consists of one or more groupings of system components and their capabilities, but the components of the groupings or the concentrations themselves are dispersed rather than restricted to a small region spanning a large geographic area, such as the entire nation, various countries, or the entire world, like the Internet, for instance. A WAN has benefits such as reaching a larger population and providing access to a variety of software and hardware resources that might not be present in a LAN. However, communication channels are sluggish and sometimes unreliable due to the vast geographic areas that WANs span.

3. **Metropolitan Area Networks (MANs)**

There is a network in the middle known as the metropolitan area network (MAN), which is somewhat wider than the Local network but not big enough to be referred to as a WAN. Cities or portions of cities with civic networks are a nice illustration of a man. Because they are frequently overshadowed by their cousin's Network to the left and WAN to the right, MANs are rarely discussed.

4. Data Communication Media Technology

The transmission technique and medium employed in the network have a significant impact on a network type's performance.

In order to achieve the relevant goals of maintaining the integrity, availability, and confidentiality of information system resources including hardware, software, and data, an automated information system is protected by computer security. The following description outlines the three main goals at the core of computer security:Confidentiality is a phrase that refers to two related ideas: Data1 confidentiality ensures that private or confidential information is not revealed to uninvited parties or made accessible to them. Privacy ensuring that people have control over what information about them may be gathered, held, and revealed, as well as by whom and to whom.

RFC 4949 defines data as "information in a specific physical representation, usually a sequence of symbols that have meaning; especially a representation of information that can be processed or produced by a computer" and information as "facts and ideas, which can be represented encoded as various forms of data." Typically, security literature does not distinguish much, and the same is true of this work. Integrity this word encompasses two related ideas:

Data integrity: Ensures that data and programs are modified only in a predetermined and approved way both when they are stored and when they are transferred as packets. System integrity ensuring that a system operates as intended, free from intentional or unintentional illegal modification of the system. Availability ensures that services are provided to authorized users and that systems operate quickly.

What is sometimes referred to as the CIA triad is made up of these three ideas. The three ideas represent the essential security goals for information and computer services, data, and both. Confidentiality, integrity, and availability are listed as the three security goals for information and information systems, respectively, in the NIST standard FIPS 199 Standards for Security Categorization of Federal Information and Information Systems. In terms of requirements and the definition of a loss of security in each category, FIPS 199 offers a good description of these three objectives:

Confidentiality maintaining lawful limitations on the access and disclosure of information, including safeguards for preserving individual privacy and proprietary data. Unauthorized information sharing is a breach of confidentiality. Integrity is the prevention of erroneous information alteration or deletion, as well as the assurance of information validity and nonrepudiation. Unauthorized alteration or destruction of data constitutes a loss of integrity.Ensuring prompt, dependable access to and use of information is known as availability. The interruption of usage or access to information or an information system is referred to as a loss of availability. Although the CIA triad has long been used to describe security goals, some in the security sector believe that other ideas are necessary to provide a whole picture. The following are two of the most often mentioned:Authenticity belief in the veracity of a transmission, a message, or the message originator; the quality of being genuine and able to be

confirmed and believed. This entails confirming that users are who they claim to be and that all input entering the system originated from a reliable source.

Accountability is the security objective that drives the need that an entity's activities can be traced back exclusively to that entity. This facilitates after-action recovery, legal recourse, fault isolation, non-repudiation, deterrent, and intrusion detection and prevention. We must be able to identify the source of a security breach since totally secure systems are not yet a reality. To enable subsequent forensic analysis to track security breaches or to assist in transaction disputes, systems must preserve records of their operations.

For these illustrations, we consider three tiers of potential damage on businesses or people in the event of a security breach (i.e., a loss of confidentiality, integrity, or availability). FIPS PUB 199 provides a definition of these levels:Low: It is possible that the loss may only have a little negative impact on an organization's operations, assets, or personnel. A limited adverse effect, on the other hand, means that, for instance, the loss of confidentiality, integrity, or availability might I cause a degradation in mission capability to the extent and duration that the organization is able to perform its primary functions, but the effectiveness of the functions is noticeably reduced, result in minor damage to organizational assets, result in minor financial loss, result in minor harm to individuals, or result in minor harm to organizational assets.

Moderate it is possible that the loss may have a significant negative impact on an organization's operations, assets, or personnel. A loss may have a serious adverse effect if it, among other things, significantly impairs the organization's ability to carry out its primary functions for an extended period of time, but the effectiveness of those functions is significantly reduced; significantly harms organizational assets; significantly increases financial loss; or significantly harms people without causing death or serious bodily injury [4], [5].

The loss can be anticipated to have a severe or catastrophic negative impact on an organization's operations, assets, or people. A severe or catastrophic adverse effect, on the other hand, denotes, for instance, that the loss might cause a severe degradation in or loss of mission capability to the point and duration that the organization is unable to perform one or more of its primary functions, result in major damage to organizational assets, result in major financial loss, result in severe or catastrophic harm to people, including loss of life or serious, life-threatening injuries, or result in anyThe Purdue University Information Technology Security and Privacy Office produced a security policy paper, which these examples are drawn from Students place a high value on the secrecy of student grade information as a valuable resource. The Family Educational Rights and Privacy Act governs how such information is disclosed in the United States (FERPA). Only students, their parents, and staff who need the knowledge to do their jobs should have access to grade information. Information about students' enrolment may be somewhat secret. While still protected by FERPA, this information is more widely disseminated, less likely to be misused than grades, and would cause less harm if made public. Lists of students, professors, or departments may be given a low secrecy grade or even no rating when it comes to directory information. On a school's website, this information is often publicly accessible to the general public.

Integrity the example of the allergy data kept in a database for a hospital patient serves to highlight many facets of integrity. The data should be reliable and up-to-date so the doctor can rely on it. Consider a scenario where a worker (like a nurse) with access to this data willfully falsifies the information in order to hurt the hospital. The database has to be swiftly restored to a reliable foundation, and it should be feasible to identify the individual who made the mistake and hold them accountable. Information on a patient's allergies is an asset with a high need for integrity. Inaccurate information might cause a patient considerable damage or even death, opening the institution up to enormous responsibility.

An example of an asset that may be given a modest degree of integrity requirement is a website that provides registered users with a forum where they can debate a certain subject. A registered user or a hacker might alter certain entries or alter the website's appearance. Potential harm is not serious if the forum serves solely to provide members with entertainment, generates little to no cash from advertising, and is not used for anything significant like study[6]. The webmaster might lose some time, money, and/or data.

An anonymous online survey is a prime illustration of a low integrity requirement. Many websites, including news organizations, provide their customers these polls with very minimal security measures. However, it is commonly known that such surveys are inaccurate and not based on science.The needed degree of availability rises with the importance of a component or service. Think of a system that offers authentication services for important applications, devices, and systems. Customers cannot use computer resources during a service disruption, and employees cannot access the resources they require to do essential activities. In terms of missed staff productivity and possible customer losses, the loss of the service results in a significant financial loss.A public university Web site is an example of an asset that would normally be categorized as having a moderate availability requirement; the Web site offers information for present and prospective students as well as benefactors. Although such a site is not essential to the university's information system, its absence will be embarrassing.

An application that searches the phone book online would be categorized as having a low availability need. There are other methods to get the information, such as a hardcopy directory or the operator, so the brief loss of the program won't be as annoying. Security is more complicated than a beginner would first think. The criteria seem to be simple; in fact, the majority of the key specifications for security services may be summed up in one word with the following definitions: secrecy, authentication, nonrepudiation, or integrity. However, the methods used to satisfy those criteria may be rather complicated, and comprehending them may need for nuanced thinking.Potential assaults on the security features must always be taken into account when creating a specific security mechanism or algorithm. In many instances, effective assaults are created by approaching the issue from a totally new perspective, thereby taking advantage of an unanticipated flaw in the system. As a result of point the methods employed to provide certain services are often illogical. A security mechanism is often complicated, and it may not be clear from the description of a given demand that such sophisticated safeguards are necessary. Extensive security measures only make sense when the threat's many facets are taken into account.

After devising many security measures, it is required to choose their application this is true in both a physical and logical sense (e.g., at what points in a network are certain security mechanisms required), such as what layer or levels of an architecture such as TCP/IP

Security measures usually go beyond a specific method or protocol additionally, they pose concerns regarding the production, dissemination, and security of any secret information that is required for participation such as an encryption key. Additionally, there can be a dependency on communications protocols, whose behavior might make it more difficult to create the security mechanism. For instance, any protocol or network that involves variable, unexpected delays may make any time limitations that must be imposed for a message's transit time from sender to recipient in order for the security mechanism to operate properly to be meaningless.

Finding flaws in computer and network security and trying to plug them is basically a game of wits between the culprit and the designer or administrator. The attacker has the significant benefit of only needing to identify one vulnerability, while the designer must identify and fix every flaw to obtain complete security.Users and system administrators have a natural inclination to undervalue the value of security investments up until a security failure occurs. Due to the short-term, overburdened environment of today, security demands routine, if not continual, monitoring, which is challenging. Rather from being an essential step in the design process, security is still much too often added to a system as an afterthought. A lot of users, even security administrators, believe that using information or operating an information system effectively and efficiently is hindered by tight security.

We will confront the aforementioned challenges in a variety of ways as we explore the different security risks and processes throughout this book. The manager in charge of security requires a systematic method for establishing security requirements and describing ways to achieving those criteria in order to analyze the security demands of an organization effectively and evaluate and choose different security products and policies. Even while this is challenging enough in a centralized data processing environment, using local and wide area networks only makes things worse.

Managers may organize the work of delivering security using the OSI security architecture. Aside from that, computer and communications manufacturers have included security features for their goods and services that connect to this organized specification of mechanisms and services since this architecture was created as an international standard. The OSI security architecture gives us a good, though abstract, overview of many of the ideas covered in this book. The emphasis of the OSI security architecture is on security methods, services, and attacks. Security attack any activity that jeopardizes the confidentiality of data that an organization owns.

A procedure or a device integrating such a process that is intended to identify, stop, or recover from a security assault is known as a security mechanism. A processing or communication service that improves the security of an organization's data processing systems and information transfers is referred to as a "security service." The services utilize one or more security mechanisms to offer the service and are designed to defend against security assaults.

Threat and assault are two words that are often used in literature to refer to the same idea. Definitions from the Internet Security Glossary, RFC 4949, The International Telecommunication Union's (ITU) Telecommunication Standardization Sector (ITU-T) is a UN-sponsored organization that creates recommendations for open systems interconnection and telecommunications standards (OSI). The OSI protocol architecture, which is detailed in Appendix L, served as the foundation for the development of the OSI security architecture. An grasp of the OSI protocol architecture is not necessary for the purposes of this chapter.

Both X.800 and RFC 4949 utilize the phrases passive attacks and active assaults to define security threats in a meaningful manner. A passive assault does not deplete system resources; instead, it tries to gather or utilize information from the system. An active assault tries to change system resources or interfere with their functionality. The characteristic of passive assaults is that they eavesdrop on or keep track of communications. The adversary wants to intercept the transmission of information in order to collect it[7]. The disclosure of message contents and traffic analysis are two examples of passive assaults. It is simple to understand how message contents are released. Sensitive or private information may be present in a phone call, an email, or a shared file. We want to keep an adversary from finding out what is being sent.

Traffic analysis, a different kind of passive assault, is more subtle. Imagine if we could conceal the information contained in messages or other information flow, preventing adversaries from using it against us even if they managed to intercept the transmission. Encryption is a typical method for hiding information. Even with encryption, an adversary could still be able to track the trend of these transmissions. The adversary may ascertain the whereabouts and identities of hosts that were in communication as well as their frequency and message length. It could be possible to infer the nature of the message from this information.

Due to the lack of data modification, passive assaults are exceedingly difficult to identify. Usually, the message traffic seems to be delivered and received normally, and neither the sender nor the recipient is aware that a third party has read the messages or tracked the traffic pattern. Nevertheless, it is possible to stop the success of these assaults, often through encryption. Therefore, preventing passive assaults is more important than detecting them when dealing with them. Masquerade, replay, message modification, and denial of service are the four subcategories of active assaults, which entail some alteration of the data stream or the generation of a fake stream.

When there is a situation, capacity, action, or event that might violate security and cause damage, there is a potential for a security violation. In other words, a threat is a potential risk that might take advantage of a weakness an intelligent act that is an intentional effort particularly in the sense of a method or technique to elude security services and breach the security policy of a system; this is an attack on system security that results from an intelligent threat.

There is a masquerade when one thing impersonates another thing one of the other active attack types is often present in a masquerade assault. For instance, authentication sequences may be recorded and replayed after a successful authentication sequence, allowing an authorized entity with limited rights to impersonate an entity with those privileges to get further privileges.

Modification of messages simply refers to the alteration of a piece of a lawful communication, as well as the delaying or rearrangement of messages in order to have an unauthorized. A notification that formerly said "Allow John Smith to view confidential file accounts" is changed to say "Allow Fred Brown to read confidential file accounts," for instance.

When a service is denied, it prohibits or hinders the regular usage or administration of communications infrastructure. This assault could be aimed against a particular person or thing; for instance, something might silence all signals going to a certain place (e.g., the security audit service). The interruption of a whole network, either by turning it off or by sending it too many messages to reduce performance, is another example of service denial. Attacks that are active exhibit the opposite traits of those that are passive. Although passive assaults are difficult to identify, there are ways to stop them from succeeding. On the other hand, since there are so many different possible physical, software, and network vulnerabilities, it is very difficult to completely avoid active assaults. Instead, the objective is to identify ongoing assaults and recover from whatever delays or disruptions they may have created. The detection may also aid in prevention if it has a deterrent impact.

According to the X.800 standard, a security service is one that is offered by the protocol layer of open systems that are interacting and assures acceptable security of the systems or of data transfers. RFC 4949, which offers the following definition, could provide a clearer one: a processing or communication service offered by a system to provide a certain level of protection for system resources; security services carry out security rules and are carried out by security mechanisms.

A communication's authenticity is ensured by the authentication service. The purpose of the authentication service is to reassure the receiver that the message is from the source that it purports to be from in the event of a single message, such as a warning or alarm signal. There are two factors at play when there is a continuous contact, such when a terminal is connected to a host. First, the service confirms that the two entities are legitimate at the moment of connection initiation, meaning that each is the entity that it represents. Second, the service must ensure that the connection is not tampered with in a manner that allows a third party to pretend to be one of the two legitimate parties in order to send or receive data that is not permitted.

Peer entity authentication enables the verification of a peer entity's identification inside an association. If two entities use distinct systems that implement the same protocol, such as two TCP modules in two communicating systems, they are regarded as peers. There is peer entity authentication available for many of the words used in security literature are not universally accepted. For instance, all facets of information security may sometimes be referred to as having integrity. Both identity verification and the different tasks outlined under integrity in this chapter are referred to as "authentication" at times. Our implementation here is compliant with RFC 4949 and X.800 the confirmation that the communicating entity is who it says it is.

Providing for the integrity of certain fields within user data of a data block transported across a connection, selective-field connection integrity determines if the specific fields have been changed, added, removed, or replayed. Single connectionless data block integrity is provided for,

and may take the form of data alteration detection. A constrained kind of replay detection might also be offered. Integrity with Selective-Field Connectionless enables the integrity of a few chosen fields inside a single connectionless data block; it does this by determining if the a few chosen fields have been changed offers defense against rejection of participation in all or a portion of a communication by one of the parties concerned.

Nonrepudiation and Origin Proof establish the sender of the communication as the designated party. Nonrepudiation, or destination proof, shows that the intended recipient of the communication really got it the Security Services (X.800) that are used during connection formation and, sometimes, data transmission. It makes an effort to provide assurance that a given entity isn't engaging in either a masquerade or an unapproved replay of a prior connection.

Data origin authentication enables the verification of a data unit's source. It doesn't provide defense against data unit duplication or change. Applications like electronic mail, in which the parties conversing have never met before, are supported by this kind of service. The capacity to restrict and regulate access to host systems and applications over communications channels is known as access control in the context of network security. In order for access privileges to be customized to the person, each entity attempting to get access must first be identified, or authenticated.

Transmitted data is shielded from passive assaults by confidentiality there are many tiers of security that may be assigned to data transmissions depending on their content. The most comprehensive solution safeguards all user data sent between two users over time. This extensive security, for instance, forbids the disclosure of any user data exchanged across a TCP connection between two computers. It is also possible to build more particular variations of this service, such as the security of a single message or even only certain message fields. These improvements are less helpful than the general strategy and can even be more difficult and costly to put into practice.

The shielding of traffic flow from analysis is the other part of secrecy. An attacker must be unable to see the source and destination, frequency, duration, or any other aspects of the traffic on a communications infrastructure in order to do this. Integrity, like confidentiality, may be applied to a single message, a stream of messages, or certain fields within a message. Again, complete stream protection is the best and most obvious solution.

When dealing with a stream of messages, a connection-oriented integrity service ensures that there is no replaying, duplication, insertion, alteration, or modification of the messages that have already been delivered. This service also includes coverage for data deletion. As a result, both message stream modification and denial of service are addressed by the connection-oriented integrity service. A connectionless integrity service, on the other hand, which deals with individual messages without taking into account any wider context, often offers security just against message tampering. We can distinguish between service with recovery and service without recovery. We are more interested in detection than prevention since the integrity service pertains to current assaults. If an integrity breach is found, the service may just notify the user of the violation; further software or human action may be needed to correct the problem. As we

shall discuss later, there are other methods for recovering from the loss of data integrity. In general, including automatic recovery techniques is a more appealing option [8]. A communication cannot be denied by the sender or the recipient due to nonrepudiation. As a result, when a communication is transmitted, the recipient may demonstrate that the message was indeed sent by the supposed sender. Similar to this, when a communication is sent, the sender may demonstrate that the purported recipient really got the message.

According to the performance requirements for the system, both X.800 and RFC 4949 define availability as the quality of a system or system resource being accessible and usable upon demand by an authorized system entity i.e., a system is available if it offers services in accordance with the system design whenever users request them. Availability may be lost or reduced as a consequence of many assaults. In order to avoid or recover from the loss of availability of components of a distributed system, some of these assaults are susceptible to automated countermeasures, such as authentication and encryption.

The availability of different security services is treated as a property by X.800. It seems appropriate to include an availability service in particular, however. A service that ensures a system's availability is an availability service. Denial-of-service attacks present security issues, which are addressed by this service. Access control services and other security services are necessary for appropriate administration and control of system resources. The security measures described in X.800 there are two types of mechanisms: those that are implemented in a particular protocol layer, like TCP or an application-layer protocol, and those that are not tied to any one protocol layer or security service in particular. The relevant Specific Security Mechanisms will address these mechanisms.

Some of the OSI security services could be implemented into the appropriate protocol layer the process of transforming facts using mathematical formulas into a form that is difficult to understand. A mathematical formula and one or more encryption keys are required for the data to be transformed and then recovered. A receiver of the data unit may use data attached to it or a cryptographic modification of it to confirm its origin and integrity and prevent tampering (e.g., by the recipient) a wide range of tools for enforcing resource access permissions several different methods for guaranteeing the integrity of a data unit or stream of data units. Mechanisms that do not belong to any one layer or security service in the OSI model something which is believed to be accurate in light of a few criteria such as those set out by a security policy [9], [10].

We won't go into further detail at this time other than to remark on the definition of decipherment. Reversible and irreversible decipherment techniques are distinguished by the X.800 standard. Simply put, a reversible decipherment method is an encryption technique that enables the encryption and subsequent decryption of data. Hash algorithms and message authentication codes are examples of irreversible decipherment techniques that are utilized in digital signature and message authentication applications.

It has proven impossible to create security design and implementation methodologies that consistently exclude security defects and stop all illegal acts, despite years of study and development. It is helpful to have a set of generally accepted design principles that may guide

the creation of protective mechanisms in the lack of such flawless methods. The following are key security design concepts, according to the National Centers of Academic Excellence in Information Assurance/Cyber Defense, which are supported by the US National Security Agency and the US Department of Homeland Security.

Complete mediation, open design, privilege separation, least privilege, least common mechanism, and least privilege isolation, encapsulation, modularity, layering, psychological acceptability, and least astonishment. The first eight enumerated guidelines were first put out in and have survived the test of time. Each premise is briefly discussed in this section. Economy of mechanism dictates that security features that are integrated into both hardware and software should be as simple and compact as feasible.

The rationale for this tenet is that small, relatively basic designs are simpler to properly test and validate. An opponent has many more possibilities to identify tiny flaws in a complicated design that may be hard to foresee beforehand. The likelihood that a mechanism has exploitable weaknesses increases with its complexity. Simple mechanisms often need less upkeep and have fewer vulnerabilities that may be exploited. Additionally, since configuration management problems are made simpler, changing or replacing a straightforward method requires less effort. This is maybe the hardest guideline to follow in real life. The challenge of security design is made more difficult by the ongoing desire for additional functionality in both hardware and software. The best that can be done to reduce needless complexity during system design is to keep this notion in mind.

Access choices need to be made based on permission rather than exclusion, according to fail-safe defaults. In other words, absence of access is the default scenario, and the protection system specifies the circumstances in which access is allowed. A system that explicitly grants permission is more likely to fail by denying it, which is a safe outcome that can be readily identified, than it does by working as intended. Contrarily, a design or implementation error in a mechanism that expressly denies access tends to fail by granting access, a failure that can go undiscovered for a while in typical usage. On client/server systems, the majority of file access systems and almost all protected services employ fail-safe defaults.

Every access must be validated against the access control mechanism in order for mediation to be complete. Access choices collected from a cache shouldn't be relied upon by systems. This concept dictates that, in a system intended to run continuously, care must be taken to ensure that changes in authority are propagated into any local memory that store access choices for later use. It seems that file access systems are an example of a system that abides by this rule. However, it's common practice not to check to determine whether permissions have changed after a user has opened a file. The system must utilize access control every time a user reads a field, record, or data item in a database in order to completely perform comprehensive mediation. Rarely is this labor-intensive strategy used.

Open design refers to the idea that a security mechanism's design should not be kept a secret. For instance, although encryption techniques should be available for public review, encryption keys must be kept confidential. The algorithms may then be examined by a large number of

specialists, giving consumers considerable trust in them. This is the guiding principle driving the NIST mission to standardize encryption and hash algorithms, which has resulted in the widespread deployment of NIST-approved methods.

According to, the practice of requiring numerous privilege characteristics in order to access a restricted resource is known as "separation of privilege." Multifactor user authentication, which involves the use of various methods, including a password and a smart card, to approve a user, is an excellent illustration of this. The phrase is now also used to describe any method of segmenting a program into pieces that are only granted the rights necessary for them to carry out a certain job. By doing this, the potential harm from a computer security assault is reduced.

This latter understanding of the notion includes the practice of transferring high-privilege activities to another process and operating that process with the higher privileges necessary to carry out its duties. Daily interfaces run in a process with less privileges. According to the principle of least privilege, each system process and user should operate with the fewest possible rights to do their tasks. The use of this idea may be seen in role-based access control. The many roles that people or processes might play can be recognized and defined by the system security policy. Only the permissions necessary for each position to carry out its responsibilities are granted. Each permission defines the permissible access to a certain resource such as connect access to a specific host and port, read and write access to a specific file or directory, etc.. The user or process should not be able to access the protected resource unless permission is expressly given. Any access control system, in principle, should only provide each user the rights that are permitted for them. The least privilege principle also has a time component[11]. For instance, specific privileges granted to system programs or administrators should only be used when required;

In order to provide security for all users, the design should reduce the functions that are shared by them. This idea makes it simpler to check if there are any negative security consequences since it lessens the number of unexpected communication pathways and the quantity of hardware and software that all users rely on. According to psychological acceptability, security measures should satisfy the demands of those who approve access while not unnecessarily interfering with users' capacity to do their jobs. Users may choose to disable security methods if they prevent resources from being used or accessed. Security measures should, if feasible, be visible to system users or, at best, cause a minimum amount of interference. Security measures must not only be unobtrusive and burdenless but also correspond to the user's mental concept of safety. Users are more prone to make mistakes if the protection protocols do not make sense to them or if they must convert their perception of protection into a protocol that is significantly different.

Three situations call for the use of the isolation principle. To avoid disclosure or manipulation, key resources data, processes, etc. should first be separated from public access systems. Organizations may desire to restrict the number of systems on which such data is housed and isolate them, either physically or conceptually, in circumstances where the sensitivity or criticality of the information is high. Making sure there is no physical link between an organization's important information and its public access information resources is one way to ensure physical isolation.

When establishing logical isolation solutions, layers of security services and processes between open systems and secure systems in charge of safeguarding vital resources should be built. Second, unless it is expressly required, each users' processes and files should be kept separate. All current operating systems include such isolation capabilities, enabling each user to have their own separated process area, memory space, and file space as well as security measures to prevent unwanted access. Finally, security measures should be segregated so that access to them is restricted. By use of logical access control, for instance, it may be possible to isolate cryptographic software from other components of the host system, safeguard it against manipulation, and conceal or change the keys.Encapsulation is one kind of isolation that is built on object-oriented functionality. The internal structure of a data item is only available to the procedures of the protected subsystem, and the procedures may only be invoked at defined domain entry points, so providing protection by enclosing a group of processes and data objects in their own domain. When discussing security, the term "modularity" encompasses both the construction of security functions as independent, secured modules and the usage of a modular architecture for the design and execution of mechanisms. The design objective in this case is to offer common security functions and services, such cryptographic operations, as common modules, as opposed to using distinct security modules. For instance, a lot of protocols and programs involve cryptographic operations. A more secure architecture is offered by creating a common cryptography module that may be used by various protocols and applications, as opposed to implementing such functionalities in each protocol or application.

The design and implementation work may then be concentrated on creating a single cryptographic module that is both secure and has procedures in place to prevent tampering. Regarding the usage of a modular architecture, each security mechanism should be able to enable the update of existing features or the migration to new technologies without necessitating a complete system redesign. The security architecture should be modular so that certain components may be updated without having to change the whole system. Layering is the employment of several, overlapping security measures to safeguard the operational, technological, and human elements of information systems. The system will still be secured if one protection method fails or is bypassed since there are other, overlapping security measures in place. The use of a layering strategy to create various barriers between an adversary and protected information or services will be evident throughout this book. Defense in depth is a term that describes this tactic often[12].A software or user interface should always react in a manner that is least likely to surprise the user, which is what the term "least astonishment" refers to. For instance, the authorization process must be clear enough to the user so that they may grasp intuitively how the security objectives relate to the security mechanism being used. We gave a general overview of the range of security risks and assaults that affect computer and network systems more information on the sorts of attacks and adversaries that pose security concerns is provided.Attack surfaces and attack trees are two ideas that are helpful in assessing and categorizing threats, and we go into further detail on them in this section code listening on open ports on external Web servers and other hosts. Internal services offered by a firewall programs that handle inbound data, email, XML, office documents, and bespoke data interchange formats particular to certain industries.

Network attack surface vulnerabilities on a business network, wide-area network, or the Internet fall under this category. This category includes network protocol flaws that may be exploited to launch intrusion attempts, interrupt communications, or cause a denial-of-service attack. Software attack surface this phrase alludes to flaws in operating system, utility, or application code. The Web server software subcategory has a specific emphasis. Human attack surface: Vulnerabilities caused by staff or visitors, such as social engineering, human mistake, and trusted insiders, fall under this category. An effective method for determining the scope and gravity of threats to a system is an attack surface analysis. Developers and security analysts become aware of the locations where security measures are needed via a methodical investigation of areas of vulnerability. Once an attack surface is identified, designers may be able to come up with strategies to reduce it, making it more challenging for the enemy to attack. Setting priorities for testing, bolstering security precautions, and altering the service or application are all aided by knowledge of the attack surface.

A collection of possible methods for exploiting security flaws are represented by an attack tree, which is a branching, hierarchical data structure. The root node of the tree represents the security incident that is the target of the attack, while the branches and sub nodes of the tree reflect the iterative and incremental methods in which an attacker may accomplish that aim. Each sub node specifies a sub goal, and every sub goal has the potential to have further sub goals of its own. The leaf nodes, which are the last nodes on the routes that branch out from the root, symbolize many methods to launch an assault. Other than leaves, every node is either an AND-node or an OR-node. All of the sub goals represented by a While-sub nodes must be accomplished in order for that node to represent a goal, and at least one sub goal must be accomplished in order for an OR-node to represent a goal. Alternative assaults may be contrasted by labeling branches with values that indicate attack qualities such as cost, difficulty, or other factors.

Utilizing the information on attack patterns to its fullest potential requires the usage of attack trees. Security advisories published by organizations like CERT have made it possible to build up a body of information regarding both broad attack tactics and particular attack patterns. The attack tree is a tool that security analysts may use to record security assaults in a way that highlights important vulnerabilities. The attack tree may help with both system and application design as well as the selection and potency of countermeasures. The attacker's goal is to compromise a user's account, which is the root of the tree[13]. The leaf nodes, which indicate the incidents that make up the assaults, are shown by the tree's coloured boxes. Keep in mind that every node in this tree aside from the leaf nodes is an OR-node.

REFRERNCES:

[1] Y. Li, G. qiu Huang, C. zi Wang, and Y. chao Li, “Analysis framework of network security situational awareness and comparison of implementation methods,” *Eurasip Journal on Wireless Communications and Networking*. 2019. doi: 10.1186/s13638-019-1506-1.

[2] X. Liu *et al.*, “Application of Temperature Prediction Based on Neural Network in Intrusion Detection of IoT,” *Secur. Commun. Networks*, 2018, doi: 10.1155/2018/1635081.

[3] S. C. de Alvarenga, S. Barbon, R. S. Miani, M. Cukier, and B. B. Zarpelão, "Process mining and hierarchical clustering to help intrusion alert visualization," *Comput. Secur.*, 2018, doi: 10.1016/j.cose.2017.11.021.

[4] M. A. M. Albashier, A. Abdaziz, and H. A. Ghani, "Performance analysis of physical layer security over different error correcting codes in wireless sensor networks," in *International Symposium on Wireless Personal Multimedia Communications, WPMC*, 2018. doi: 10.1109/WPMC.2017.8301806.

[5] R. F. Liao, H. Wen, J. Wu, H. Song, F. Pan, and L. Dong, "The Rayleigh Fading Channel Prediction via Deep Learning," *Wirel. Commun. Mob. Comput.*, 2018, doi: 10.1155/2018/6497340.

[6] M. V. Pawar and J. Anuradha, "Network security and types of attacks in network," in *Procedia Computer Science*, 2015. doi: 10.1016/j.procs.2015.04.126.

[7] H. Te Wu and C. W. Tsai, "An intelligent agriculture network security system based on private blockchains," *J. Commun. Networks*, 2019, doi: 10.1109/JCN.2019.000043.

[8] W. Han, Z. Tian, Z. Huang, L. Zhong, and Y. Jia, "System architecture and key technologies of network security situation awareness system YHSAS," *Comput. Mater. Contin.*, 2019, doi: 10.32604/cmc.2019.05192.

[9] J. Qin, M. Li, L. Shi, and X. Yu, "Optimal Denial-of-Service Attack Scheduling with Energy Constraint over Packet-Dropping Networks," *IEEE Trans. Automat. Contr.*, 2018, doi: 10.1109/TAC.2017.2756259.

[10] P. Kendrick, N. Criado, A. Hussain, and M. Randles, "A self-organising multi-agent system for decentralised forensic investigations," *Expert Syst. Appl.*, 2018, doi: 10.1016/j.eswa.2018.02.023.

[11] M. H. Manshaei, Q. Zhu, T. Alpcan, T. Basar, and J. P. Hubaux, "Game theory meets network security and privacy," *ACM Computing Surveys*. 2013. doi: 10.1145/2480741.2480742.

[12] J. He *et al.*, "Customized network security for cloud service," *IEEE Trans. Serv. Comput.*, 2020, doi: 10.1109/TSC.2017.2725828.

[13] J. Hu, S. Guo, X. Kuang, F. Meng, D. Hu, and Z. Shi, "I-HMM-Based Multidimensional Network Security Risk Assessment," *IEEE Access*, 2020, doi: 10.1109/ACCESS.2019.2961997.

CHAPTER 2

TRANSMISSION TECHNOLOGY

Dr. Senthilkumar S, Professor & HOD
Department of Computer Science and Engineering, Presidency University, Bangalore, India
Email Id- senthilkumars@presidencyuniversity.in

The signal to be utilized is determined by the medium via which information must be transferred. Certain mediums only accept analog transmissions. For some, analog and digital are permitted. Consequently, depending on the type of medium involved and other factors, the supplied information can be displayed as an analog or digital signal. In an analog format, data is carried through a variety of media, such as conductive material, twisting coaxial pair or cable, fiber optic, or wireless, as continuous electromagnetic radiation on an interval that represents things like speech and video. Soon, we'll talk about these mediums. However, in an electronic medium, information is transmitted as a digital signal, which is a series of voltage spikes that may be seen as a stream.

On the other hand, when data is delivered in an electronic medium, it does so as a digital signal, which is a series of output pulses that may be thought of as a continuous stream of binary bits. Data may spread both digitally and analogically, and it does so often in either an analog or digital representation. The act of transmitting data signals between network components involves both their propagation and processing. Encoding schemes refer to the idea of representing data for transmission as either an analog or digital signal. Then, a suitable transmission channel connecting all network components transmits the encoded data. Analog and digital encoding are also available. Analog encoding transmits analog signals that stand in for analog data like speech and sound waves.

Analog Encoding of Digital Data:

Remember that digital data is represented by 1s or 0s. Digital data must be encoded using a modulator and demodulation to create analog signals before being transferred over an analog channel, such as a telephone line, which has a finite amount of bandwidth. An ongoing oscillating wave typically a sine wave and a carrier signal with a fixed frequency are used in the encoding process. Intensity, speed, and phase shift are the three modulation properties of the carrier. The information stream is then modulated and demodulated by a computer, a modulation-demodulation pair, depending on any one of the three carrier properties or a combination. The resultant wave spans a variety of frequencies on both sides of the carrier, as seen below.Each binary value is represented through amplitude modulation by altering the carrier frequency's amplitude. Any other wavelength then indicates a1, and the lack of or low-frequency band may represent a 0. But this is a somewhat ineffective way exclusively used at low levels up to 1200 bps in speech grade lines as a modulation method. Frequency range also uses two distinct frequencies that are close to a frequency of an underlying carrier to represent the two binary values. A1 is represented by higher frequencies, and a 0 by lower frequencies. The system is less prone to mistakes.

Phase angle modulation shifts the carrier wave's phase while altering the carrier wave's time to encode data A1 is represented by a 180-degree phase shift.

Digital Encoding of Digital Data:

However, by delivering a separate clock signal, this is diminished. There are yet more representations, like the London and differential Manchester, which also encode the data's clock information. Why go through the trouble of encoding and transmission is a valid question. Over its cousin, analog encoding, it has several benefits.The most popular and straightforward method for transmitting digital signals, this encoding strategy uses two binary digits to represent two distinct voltages.

These voltages are typically 0 and 5 volts inside of a computer. Another method makes use of two representation codes including no return at a low point in which binary one is represented by negative voltage and binary zero by positive voltage, and no return to zero, inverted on ones (NRZ-I). Examples of these two codes may be found. In NRZ-L, a change in voltage level is employed to signal the information whenever a 1 occurs. The necessity for exact synchronization between the receiver and transmitter clocks is one issue with NRZ signaling systems. Why go through the trouble of digital encoding and transmission is a valid question. Over its cousin, analog encoding, it has several benefits. They consist of the following:

a) Falling prices for digital circuits the following benefits of using digital transmission techniques: Better integration of voice, video, text, and picture
b) Reduction of noise and other signal damage due to the usage of repeaters
c) Best channel capacity utilization

The volume of transferred data may frequently be far more than the capacity of the network medium during data transmission. When this occurs, it could be feasible to share a transmission amongst several signal carriers medium. This process is known as multiplexing. Time-division multiplexing (TMD) and frequency-division multiplexing are the two methods for achieving multiplexing (FDM). All data channels in FDM are initially converted to analog. Each analog signal is then modulated by a distinct and distinctive carrier frequency since several signals may be carried on a single carrier. This allows for recovery during the multiplexing process. After that, the carriers for the frequencies are combined. The decoder can choose at the receiving end[1], [2].

Any type of communication must have a medium via which it may be carried out, as we have seen above. As a result, communication between network pieces in a network requires a medium. No network can work without a transmission channel, the transmitting components would not be connected. The effectiveness of the network is significantly influenced by the transmission medium. A network's transmission medium has a significant impact on the distinctive quality, reliability, and overall performance of the network. The capacity of a network to handle expected network traffic, reliability of the network's operation, the network for example in terms of the area covered, and transmission rate are all influenced by the transmission medium. There are two types of network transmission media wired and wireless.

Transmission technology refers to the methods and systems used to transfer information, data, or signals from one point to another. This could include the transfer of data over wired or wireless networks, the distribution of broadcast signals over radio or television networks, or the transmission of power from one location to another[3]. In this article, we will discuss the various types of transmission technology and their applications.

Wired transmission technology is the oldest and most established form of transmission technology. It refers to the transfer of information, data, or signals over a physical cable or wire. Some examples of wired transmission technology include:

1. **Twisted Pair Cables:** These cables consist of two copper wires twisted together to reduce interference from other signals. They are commonly used for telephone and internet connections.
2. **Coaxial Cables:** These cables consist of a central copper conductor surrounded by insulation, a copper shield, and a protective outer layer. They are commonly used for cable television connections.
3. **Fiber Optic Cables:** These cables use light to transmit information, data, or signals. They consist of a thin glass or plastic fiber that is surrounded by insulation and protective layers. They are commonly used for high-speed internet connections and long-distance communication.

Wired transmission technology has several advantages, including high reliability, fast transfer speeds, and low interference. However, it also has some disadvantages, including the need for physical cables or wires, which can be expensive to install and maintain.

Wireless Transmission Technology

Wireless transmission technology refers to the transfer of information, data, or signals without the use of physical cables or wires. It relies on electromagnetic waves to transmit information through the air. Some examples of wireless transmission technology include:

1. **Radio Waves:** Radio waves are used to transmit signals for radio and television broadcasting, as well as for wireless communication.
2. **Microwaves:** Microwaves are used for long-distance communication, such as satellite communications and radar systems.
3. **Infrared Waves:** Infrared waves are used for short-range communication, such as remote controls for televisions and other devices.

Wireless transmission technology has several advantages, including the ability to transmit information without the need for physical cables or wires, which can be more cost-effective and easier to install. However, it also has some disadvantages, including the potential for interference and the limited range of transmission.

Power transmission technology refers to the transfer of electrical power from one location to another. This could include the transmission of power from a power plant to a city or from a wind farm to a substation. Some examples of power transmission technology include:

1. **Power Lines:** Power lines are used to transmit electrical power over long distances. They consist of overhead cables or underground cables.
2. **Transformers:** Transformers are used to change the voltage of electrical power to make it more suitable for transmission over long distances.
3. **Substations:** Substations are used to distribute electrical power to smaller areas, such as neighborhoods or industrial facilities.

Power transmission technology has several advantages, including the ability to transfer large amounts of electrical power over long distances. However, it also has some disadvantages, including the potential for power loss and the environmental impact of power lines. Transmission technology plays a critical role in our modern world. It allows us to transfer information, data, or signals over long distances and to distribute electrical power to large areas. Wired transmission technology, wireless transmission technology, and power transmission technology are just a few examples of the many types of transmission technology that are used today. Each type has its advantages and disadvantages, and the choice of transmission technology will depend on the specific application and the needs of the user.

Wired transmission technology has been in use for many years and is a reliable and secure way of transmitting data. One of the primary advantages of wired transmission technology is its high reliability. Since data is transmitted over a physical cable or wire, the risk of interference is minimal. This makes wired transmission technology ideal for applications that require a high degree of reliability, such as telecommunications and computer networking.

Another advantage of wired transmission technology is its fast transfer speeds. With the right infrastructure in place, data can be transmitted at very high speeds over wired networks. This makes wired transmission technology ideal for applications that require high-speed data transfer, such as video streaming and large file transfers. However, there are some disadvantages to wired transmission technology. One of the most significant disadvantages is the need for physical cables or wires, which can be expensive to install and maintain. This is particularly true for fiber optic cables, which require specialized installation and maintenance equipment. Additionally, the physical limitations of cables and wires can limit the range of transmission.Wireless transmission technology is becoming increasingly popular due to the growth of mobile devices and the internet of things. One of the primary advantages of wireless transmission technology is its flexibility. Since it does not rely on physical cables or wires, wireless transmission technology can be deployed in a wide variety of environments. This makes it ideal for applications that require a high degree of flexibility, such as mobile computing and home automation.Another advantage of wireless transmission technology is its ease of use. Unlike wired transmission technology, there is no need for physical connections, which can be time-consuming to set up and maintain. Wireless transmission technology also allows for more natural integration with mobile devices, which can be used to control various devices and access the internet.

However, there are some disadvantages to wireless transmission technology. One of the most significant disadvantages is the potential for interference. Since wireless signals travel through the air, they can be affected by various environmental factors, such as walls, other devices, and weather conditions. This can lead to a reduced range of transmission and slower transfer speeds.

Power transmission technology is used to distribute electrical power from power plants to homes and businesses. One of the primary advantages of power transmission technology is its ability to transmit large amounts of electrical power over long distances. This is done through power lines, which are typically either overhead cables or underground cables. Overhead cables are more common and consist of tall towers that support the cables. Underground cables are more expensive but offer the advantage of being less visible and less affected by weather conditions[4], [5].

Another advantage of power transmission technology is its ability to control voltage. Electrical power is typically generated at a high voltage, which is not suitable for distribution over long distances. Power transmission technology uses transformers to reduce the voltage and make it more suitable for transmission. Transformers are used at substations, which distribute power to smaller areas, such as neighborhoods or industrial facilities.

However, there are some disadvantages to power transmission technology. One of the most significant disadvantages is the potential for power loss. Electrical power is lost as it travels through power lines due to resistance, which can result in lower voltages and less efficient distribution. Additionally, the environmental impact of power transmission technology is a concern, as power lines and other infrastructure can impact natural habitats and wildlife.

Transmission technology is essential for modern life, as it enables us to transmit data and distribute electrical power over long distances. Wired transmission technology, wireless transmission technology, and power transmission technology are just a few examples of the many types of transmission technology that are in use today. Each type of transmission technology has its advantages and disadvantages, and the choice of technology will depend on the specific application and the needs of the user. As technology continues to advance, new forms of transmission technology will likely emerge, offering even greater flexibility and efficiency.

Fiber optic transmission technology is a type of wired transmission technology that uses glass or plastic fibers to transmit data. One of the primary advantages of fiber optic transmission technology is its fast transfer speeds. Fiber optic cables can transmit data at speeds of up to 100 gigabits per second, making them ideal for applications that require high-speed data transfer, such as internet and data centers.

Another advantage of fiber optic transmission technology is its high bandwidth. Fiber optic cables can transmit data over long distances without the need for amplification or regeneration, making them ideal for long-distance communication. They are also resistant to electromagnetic interference, which can cause problems with other types of wired transmission technology.

However, there are some disadvantages to fiber optic transmission technology. One of the most significant disadvantages is the cost. Fiber optic cables are more expensive than other types of cables, and the installation and maintenance costs can be high. Additionally, fiber optic cables are fragile and can be damaged by bending or twisting, which can lead to transmission problems.

Satellite transmission technology is a type of wireless transmission technology that uses satellites to transmit data over long distances. One of the primary advantages of satellite transmission technology is its global coverage. Satellites can transmit data to any location on the planet, making them ideal for applications that require global coverage, such as telecommunications and broadcasting.

Another advantage of satellite transmission technology is its ability to transmit data to remote or inaccessible locations. Satellites can transmit data to areas that are not served by other types of transmission technology, such as rural areas or areas that are affected by natural disasters. However, there are some disadvantages to satellite transmission technology. One of the most significant disadvantages is the potential for signal degradation. Since satellite signals must travel through the atmosphere, they can be affected by environmental factors, such as weather conditions and interference from other wireless signals. Additionally, the cost of launching and maintaining satellites can be high.

Bluetooth transmission technology is a type of wireless transmission technology that is used to connect devices such as mobile phones, tablets, and computers to other devices, such as speakers and headphones. One of the primary advantages of Bluetooth transmission technology is its low power consumption. Bluetooth devices use very little power, making them ideal for use in mobile devices, such as smartphones and tablets.

Another advantage of Bluetooth transmission technology is its ease of use. Bluetooth devices can be easily paired with other Bluetooth devices, allowing users to quickly and easily connect their devices to other devices, such as speakers and headphones. However, there are some disadvantages to Bluetooth transmission technology. One of the most significant disadvantages is the limited range. Bluetooth devices have a limited range of about 30 feet, which can be a problem in large spaces, such as conference rooms or lecture halls. Additionally, the potential for interference from other wireless signals can lead to reduced signal quality transmission technology plays a vital role in our daily lives, from enabling us to communicate with others to distributing electrical power to our homes and businesses. Wired and wireless transmission technology both have their advantages and disadvantages, and the choice of technology will depend on the specific application and the needs of the user. As technology continues to advance, new forms of transmission technology will likely emerge, offering even greater flexibility, efficiency, and connectivity.Power transmission technology refers to the methods used to transport electrical power from power plants to homes and businesses. There are two primary types of power transmission technology: overhead transmission and underground transmission. Overhead transmission technology uses large, tall structures called transmission towers to support electrical cables. Overhead transmission is less expensive than underground transmission and is easier to repair and maintain. However, it can be susceptible to weather-related damage and can be unsightly in some areas.

Underground transmission technology uses cables that are buried underground to transport electrical power. Underground transmission is more expensive than overhead transmission and is more difficult to repair and maintain. However, it is less susceptible to weather-related damage and is more aesthetically pleasing.

Wireless power transmission technology is a type of power transmission technology that uses electromagnetic fields to transfer power wirelessly between devices. One of the primary advantages of wireless power transmission technology is its convenience. Devices can be charged wirelessly, without the need for cables or charging ports. This makes it ideal for use in mobile devices, such as smartphones and tablets. Another advantage of wireless power transmission technology is its potential to reduce electronic waste. Since devices can be charged wirelessly, there is less need for cables and charging ports, which can reduce electronic waste and lead to a more sustainable future.

However, there are some disadvantages to wireless power transmission technology. One of the most significant disadvantages is the potential for energy loss. Wireless power transmission can be less efficient than traditional wired transmission, resulting in energy loss and reduced power output. Additionally, wireless power transmission can be affected by environmental factors, such as distance and interference from other electromagnetic fields transmission technology plays a critical role in our daily lives, from enabling us to communicate with others to distributing electrical power to our homes and businesses[6]. Wired and wireless transmission technology both have their advantages and disadvantages, and the choice of technology will depend on the specific application and the needs of the user.As technology continues to advance, new forms of transmission technology will likely emerge, offering even greater flexibility, efficiency, and connectivity. In the future, we may see new developments in wireless power transmission, advancements in fiber optic technology, and the widespread adoption of renewable energy sources. Whatever the future may hold, transmission technology will continue to play a vital role in shaping our world. Radio frequency (RF) transmission technology is a type of wireless transmission technology that uses radio waves to transmit data. One of the primary advantages of RF transmission technology is its long-range. RF transmission technology can transmit data over long distances, making it ideal for applications that require long-distance communication, such as broadcasting and wireless networking.Another advantage of RF transmission technology is its ability to penetrate walls and other obstacles. RF signals can pass through walls and other objects, allowing devices to communicate with each other even if they are not in direct line of sight. However, there are some disadvantages to RF transmission technology. One of the most significant disadvantages is the potential for interference. RF signals can be affected by interference from other wireless signals, which can lead to reduced signal quality and connectivity issues. Additionally, RF signals can be affected by environmental factors, such as weather conditions and terrain.Near field communication (NFC) transmission technology is a type of wireless transmission technology that allows devices to communicate with each other when they are within a few centimeters of each other. One of the primary advantages of NFC transmission technology is its low power consumption. NFC devices use very little power, making them ideal for use in mobile devices, such as smartphones and tablets.

Another advantage of NFC transmission technology is its ease of use. NFC devices can be easily paired with other NFC devices, allowing users to quickly and easily connect their devices to other devices, such as speakers and headphones. However, there are some disadvantages to NFC transmission technology. One of the most significant disadvantages is the limited range. NFC devices have a limited range of about 4 centimeters, which can be a problem in larger spaces. Additionally, the potential for interference from other wireless signals can lead to reduced signal quality.

Microwave transmission technology is a type of wireless transmission technology that uses microwave signals to transmit data. One of the primary advantages of microwave transmission technology is its high bandwidth. Microwave signals can transmit data at high speeds, making them ideal for applications that require high-speed data transfer, such as internet and data centers. Another advantage of microwave transmission technology is its long-range. Microwave signals can travel long distances without the need for amplification or regeneration, making them ideal for long-distance communication.

However, there are some disadvantages to microwave transmission technology. One of the most significant disadvantages is the potential for interference. Microwave signals can be affected by interference from other wireless signals, which can lead to reduced signal quality and connectivity issues. Additionally, the potential for atmospheric interference, such as rain and snow, can also lead to reduced signal quality[7], [8].

Infrared transmission technology is a type of wireless transmission technology that uses infrared signals to transmit data. One of the primary advantages of infrared transmission technology is its security. Infrared signals cannot pass through walls and other objects, making them more secure than other types of wireless transmission technology. Another advantage of infrared transmission technology is its low power consumption. Infrared devices use very little power, making them ideal for use in mobile devices, such as smartphones and tablets.

However, there are some disadvantages to infrared transmission technology. One of the most significant disadvantages is the limited range. Infrared devices have a limited range of about 1 meter, which can be a problem in larger spaces. Additionally, the potential for interference from other sources of infrared radiation, such as sunlight, can lead to reduced signal quality transmission technology is a broad and diverse field that includes a wide range of wired and wireless transmission technologies. Each type of transmission technology has its advantages and disadvantages, and the choice of technology will depend on the specific

In continuation of our discussion on transmission technology, we will further explore some of the advanced technologies that are being developed in this field. Visible Light Communication (VLC) technology is a type of wireless transmission technology that uses light waves to transmit data. VLC technology uses the visible spectrum of light to transmit data, and the data is transmitted through the modulation of light signals.

One of the primary advantages of VLC technology is its high-speed data transfer. VLC technology can transmit data at very high speeds, up to several gigabits per second, making it

ideal for high-speed data transfer applications. Additionally, VLC technology is highly secure since it cannot pass through walls or other obstacles.

Another advantage of VLC technology is its low power consumption. VLC devices use very little power, making them ideal for use in mobile devices, such as smartphones and tablets. However, there are some disadvantages to VLC technology. One of the most significant disadvantages is the limited range. VLC devices have a limited range of about 10 meters, which can be a problem in larger spaces. Additionally, the potential for interference from other light sources, such as sunlight, can lead to reduced signal quality.

Li-Fi transmission technology is a type of wireless transmission technology that uses light waves to transmit data. Li-Fi technology is a variation of VLC technology, but it uses the entire spectrum of light to transmit data. One of the primary advantages of Li-Fi technology is its high-speed data transfer. Li-Fi technology can transmit data at very high speeds, up to several gigabits per second, making it ideal for high-speed data transfer applications. Additionally, Li-Fi technology is highly secure since it cannot pass through walls or other obstacles.

Another advantage of Li-Fi technology is its low power consumption. Li-Fi devices use very little power, making them ideal for use in mobile devices, such as smartphones and tablets. However, there are some disadvantages to Li-Fi technology. One of the most significant disadvantages is the limited range. Li-Fi devices have a limited range of about 10 meters, which can be a problem in larger spaces. Additionally, the potential for interference from other light sources, such as sunlight, can lead to reduced signal quality.

Ultra-Wideband (UWB) transmission technology is a type of wireless transmission technology that uses a very large frequency range to transmit data. UWB technology uses a frequency range of 3.1 GHz to 10.6 GHz to transmit data. One of the primary advantages of UWB technology is its high-speed data transfer. UWB technology can transmit data at very high speeds, up to several gigabits per second, making it ideal for high-speed data transfer applications[9], [10].

Another advantage of UWB technology is its ability to penetrate walls and other obstacles. UWB signals can pass through walls and other objects, allowing devices to communicate with each other even if they are not in direct line of sight. However, there are some disadvantages to UWB technology. One of the most significant disadvantages is the potential for interference. UWB signals can be affected by interference from other wireless signals, which can lead to reduced signal quality and connectivity issues. Additionally, UWB signals can be affected by environmental factors, such as weather conditions and terrain.

Transmission technology is a rapidly evolving field that is continually producing new and innovative technologies. Each type of transmission technology has its advantages and disadvantages, and the choice of technology will depend on the specific application and requirements. As technology continues to evolve, we can expect to see even more advanced transmission technologies that will revolutionize the way we communicate and transfer data.

REFERENCES:

[1] Y. Zou and J. Lv, "Information security transmission technology in Internet of things control system," *Int. J. Online Eng.*, 2018, doi: 10.3991/ijoe.v14i06.8707.

[2] X. Xu, P. Dong, Y. Liu, and H. Zhang, "Progress in Automotive Transmission Technology," *Automot. Innov.*, 2018, doi: 10.1007/s42154-018-0031-y.

[3] W. T. Sung and S. J. Hsiao, "IoT network security and applications via long range technology," *Sensors Mater.*, 2020, doi: 10.18494/SAM.2020.2569.

[4] G. Arcia-Garibaldi, P. Cruz-Romero, and A. Gómez-Expósito, "Future power transmission: Visions, technologies and challenges," *Renewable and Sustainable Energy Reviews*. 2018. doi: 10.1016/j.rser.2018.06.004.

[5] S. Zhou, F. Rong, Z. Yin, S. Huang, and Y. Zhou, "HVDC transmission technology of wind power system with multi-phase PMSG," *Energies*, 2018, doi: 10.3390/en11123294.

[6] X. Tao, K. Kong, F. Zhao, S. Cheng, and S. Wang, "An efficient method for network security situation assessment," *Int. J. Distrib. Sens. Networks*, 2020, doi: 10.1177/1550147720971517.

[7] G. Liu *et al.*, "Wireless Electric Energy Transmission through Various Isolated Solid Media Based on Triboelectric Nanogenerator," *Adv. Energy Mater.*, 2018, doi: 10.1002/aenm.201703086.

[8] Q. Huang *et al.*, "Secure free-space optical communication system based on data fragmentation multipath transmission technology," *Opt. Express*, 2018, doi: 10.1364/oe.26.013536.

[9] Y. Liu, H. Tian, Z. Liu, and X. Qin, "Aspects of ultra-high voltage half-wavelength power transmission technology," *Glob. Energy Interconnect.*, 2018, doi: 10.14171/j.2096-5117.gei.2018.01.012.

[10] E. Le Taillandier De Gabory, K. Matsumoto, and H. Takeshita, "Advances in Power-Efficient SDM Transmission Technologies," in *Asia Communications and Photonics Conference, ACP*, 2018. doi: 10.1109/ACP.2018.8596122.

CHAPTER 3

THE OSI SECURITY ARCHITECTURE

Dr. Pravinthraja, Associate Professor
Department of Computer Science and Engineering, Presidency University, Bangalore, India
Email Id- pravinth.raja@presidencyuniversity.in

The OSI Security Architecture is a framework that defines how security mechanisms should be implemented in a computer network. It is part of the OSI (Open Systems Interconnection) reference model, which is a seven-layer model that describes how network protocols should be organized and interact with each other. The OSI Security Architecture extends this model by defining security mechanisms that can be applied at each layer to protect the network and the data that is transmitted over it. In this article, we will explain the OSI Security Architecture in detail, including its seven layers and the security mechanisms that can be applied at each layer.

1. Layer 1: Physical Layer the Physical Layer is the first layer of the OSI model and it deals with the physical aspects of network communication, such as cables, connectors, and other hardware. At this layer, security mechanisms are limited to physical access control, which means that access to network devices should be restricted to authorized personnel only. This can be achieved by using physical locks, biometric authentication, or other similar mechanisms.
2. Layer 2: Data Link Layer the Data Link Layer is responsible for the transfer of data between adjacent network nodes. At this layer, security mechanisms include MAC (Media Access Control) filtering and VLAN (Virtual Local Area Network) segregation. MAC filtering allows network administrators to limit access to the network based on the MAC addresses of the devices trying to connect. VLAN segregation, on the other hand, allows network administrators to create virtual LANs that are logically separated from each other, even though they may share the same physical network.
3. Layer 3: Network Layer the Network Layer is responsible for routing data packets between different networks. At this layer, security mechanisms include packet filtering, which involves analyzing the headers of incoming and outgoing packets and determining whether they should be allowed to pass or not. This can be done using access control lists (ACLs) or firewalls, which are devices that are specifically designed to filter network traffic.
4. Layer 4: Transport Layer the Transport Layer is responsible for providing reliable data transfer between applications running on different network nodes. At this layer, security mechanisms include TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) port filtering, which can be used to limit the types of traffic that are allowed to pass through the network. This can be useful in preventing attacks that rely on exploiting vulnerabilities in specific application protocols.

5. Layer 5: Session Layer the Session Layer is responsible for managing the communication sessions between network nodes. At this layer, security mechanisms include session encryption and decryption, which involves encrypting the data that is exchanged between the network nodes and then decrypting it at the other end. This can be achieved using symmetric or asymmetric encryption algorithms, such as AES (Advanced Encryption Standard) or RSA (Rivest-Shamir-Adleman)[1], [2].
6. Layer 6: Presentation Layer the Presentation Layer is responsible for transforming data into a format that can be understood by the application running on the other end of the network. At this layer, security mechanisms include data compression and encryption, which can be used to reduce the size of the data being transmitted and protect it from unauthorized access.
7. Layer 7: Application Layer the Application Layer is responsible for providing network services to applications running on different network nodes. At this layer, security mechanisms include digital signatures and message authentication codes (MACs), which can be used to verify the identity of the sender and ensure that the message has not been tampered with in transit. This can be achieved using cryptographic hash functions, such as SHA-256 (Secure Hash Algorithm 256).

Overall, the OSI Security Architecture provides a comprehensive framework for implementing security mechanisms in a computer network. By applying security measures at each layer of the network stack, it is possible to create a highly secure network that is resistant to various types of attacks. However, it is important to any type of communication must have a medium via which it may be carried out, as we have seen above. As a result, communication between network pieces in a network requires a medium. No network can work without a transmission channel, the transmitting components would not be connected. The performance of the network is significantly influenced by the transmission medium. A network's transmission medium has a significant impact on the distinctive quality, reliability, and overall performance of the network. The capacity of a network to handle expected network traffic, dependability of the network's availability, size of the network in terms of the distance covered, and transmission rate are all influenced by the transmission medium. There are two types of network transmission media: wired and wireless.

In fixed networks, every network component is physically connected via wired transmission means. Physical media can take many different forms, but the most popular ones include copper wires, twisted pairs, coaxial cables, and optical fibers. Due to their low resistance to electrical currents, copper cables have historically been employed in communication because they enable signals to go longer. However, electromagnetic activity in the surroundings can interfere with copper cables, thus they must constantly be insulated. A pair of wires known as a "twisted pair" is made up of two insulated copper wires that have been repeatedly twisted around one another. The insulated, twisted copper wires work as a full-duplex communication channel when connected. Twisting the wires lessens the cable's susceptibility to electromagnetic fields.

Twisting the wires lessens the cable's sensitivity to electromagnetic interference and the emission of radio frequency disturbances that might interfere with surrounding electronics' electrical

components and wires. More than one pair of twisted wires may be grouped in a protective covering to improve the capacity of the transmission medium. Twisted pairs were frequently employed in telephone networks due to their low cost, simplicity of installation, and great voice data quality. Twisted pair technology has been replaced by newer technologies because of its poor upward scalability in terms of transmission rate, distance, and capacity in LANs dual-conductor cables with a common inner conductor are known as coaxial cables[3], [4].

Coaxial cables are dual-conductor cables with an outside conductor around the insulation and a shared interior conductor in the cable's core that is shielded by insulation. Because they share an inner conductor, these cables are known as coaxial. Typically, the inner core conductor is formed of solid copper wire, however, it there be constructed using stranded wire as well. The outer conductor, which is often formed of braided wires but can also be made of metallic foil or both, surrounds the inner conductor in a protective tube. Another outside layer known as the sheath further shields this outer conductor from harm.

Coaxial cables may be used over a greater area than twisted pairs. There are two varieties of coaxial cables: thinner, a cheap, lightweight, and flexible cabling medium; and thick, a heavy, stiff, and difficult-to-install wire which, compared to the investor to invest, is thicker, more durable, and capable of carrying more signals farther. An optical beam can travel via fiber optics, a tiny medium comprised of glass and plastic. Due to its ability to support exceptionally high bandwidths and lack of issues with electromagnetic interference that coaxial cables have, this cable is the most suitable for data transfer. Additionally, it can sustain cable runs of many meters. However, the two drawbacks of fiber-optic cables are their expense and complex installation.

Basic fiber optics comprises a core comprised of tiny glass or plastic strands. A layer made of glass or plastic known as cladding acts as a protective barrier. Despite using the same materials, the cladding Material like the core has various characteristics that enable them to reflect core rays that tangentially strike them. The actual cladding is protected by a plastic jacket. The jacket guards against damage from the outside, such as bending and abrasions, to the inner fiber. Data signals are first converted into light signals before being sent across fiber optic lines. An injection laser diode or a light-emitting diode (LED) at the source emits the transmitted light (ILD).

The receiver receives the instruction as long as there is no obstruction in the way of the light's transmission. The most effective usage of infrared is in a small, enclosed space, such as a television remote control device within 100 feet of it a television. Infrared is a reasonably quick technology that can support large bandwidths of up to 10 Megabytes per second in a small space like this. Radio at High-Frequency High-frequency electromagnetic radio waves, often known as RF transmissions, are produced by the transmitter and are picked up by the receiver during a radio connection. Mobile computer components may communicate across a smaller area than infrared since the radio frequency band's range is longer than that of infrared. This eliminates the need for both the transmitter and the receiver to be situated in a straight line.

1. **Intermediate/Extended Network:** The two fixed LAN components of this wireless network are connected by a wireless component. The bridge can be tying together LANs in two close structures or even farther away.
2. **Mobile Network:** This completely wireless network links up two different network components. A mobile device that utilizes cellular or satellite technologies to connect to the fixed home network often makes up one of these components.

Basic media including infrared, laser beam, narrow-band, and spread-spectrum radio, microwave, and satellite communication are used to connect these three different types of wireless networks. Infrared In an infrared transmission, one network element sends pulses of infrared light to the receiving network element over a distance, carrying coded instructions.

Infrared In an infrared transmission, one network element sends pulses of infrared light to the receiving network element over a distance, carrying coded instructions. Whenever there is nothing in the way of the transmitted light, the receiver receives the directive. The most successful usage of infrared is in a small, enclosed space, up to 100 feet away, such as when a television remote controls a television. Infrared is a reasonably quick technology that can support large bandwidths of up to 10 Mbps in a small space like this. Radio at High-Frequency electromagnetic radio waves, often known as RF transmissions, are produced by the transmitter and are picked up by the receiver during a radio connection. Due to the radio frequency band's wide frequency range.

Radio at High-Frequency electromagnetic radio waves, often known as RF transmissions, are produced by the transmitter and are picked up by the receiver during a radio connection. Mobile computing is possible because the radio frequency band's range is wider than infrared's. Without the transmitter and receiver being located in a direct line of sight, components can communicate across a small distance since the signal can bounce off light.

The three ideas represent the essential security goals for information and computer services, data, and both. NIST Standards for Security, for instance confidentiality, integrity, and availability are listed as the three security goals for information and information systems in Categorization of Federal Information and Information Systems (FIPS 199). In terms of criteria and the definition of a loss of security in each category, FIPS 199 offers a good description of these three goals[2], [5].

2 RFC 2828 defines data as "information in a specific physical representation, usually a sequence of symbols that have meaning; especially a representation of information that can be processed or produced by a computer" and information as "facts and ideas, which can be represented (encoded) as various forms of data." Typically, security literature does not distinguish much, and the same is true of this work.

Confidentiality upholding lawful limitations on the access and disclosure of information, including safeguards for safeguarding individual privacy and private data. Unauthorized information sharing is a breach of confidentiality. Integrity refers to safeguarding against erroneous information alteration or deletion, as well as maintaining information validity and nonrepudiation. Unauthorized alteration or destruction of data constitutes a loss of integrity.

Availability making sure that information is timely and trustworthy to utilize. The interruption of usage or access to information or an information system is referred to as a loss of availability.

Although the CIA trio is often used to outline security goals, some people in the security industry believe that more ideas are required to provide a whole picture. The two that are most often stated are: Authenticity confidence in the veracity of a transmission, a message, or message originator. Authenticity is the quality of being genuine and being able to be confirmed and trusted. This entails confirming that users are who they claim to be and that all input entering the system originated from a reliable source.

Accountability the security objective that necessitates the need for an entity's activities to be individually traceable to that entity this facilitates after-action recovery, legal recourse, fault isolation, non-repudiation, deterrent, and intrusion detection and prevention. We must be able to identify the source of a security breach since totally secure systems are not yet a reality.

To enable subsequent forensic analysis to track security breaches or to assist in transaction disputes, systems must preserve records of their operations. We now provide a few examples of applications that demonstrate the aforementioned criteria. For these illustrations, we employ three tiers of effect on businesses or people in the event of a security breach (i.e., a loss of confidentiality, integrity, or availability). In FIPS 199, several levels are described:

A limited adverse effect, on the other hand, means that, for instance, the loss of confidentiality, integrity, or availability might cause a degradation in mission capability to the extent and duration that the organization is able to perform its primary functions, but the effectiveness of the functions is noticeably reduced, result in minor damage to organizational assets, result in minor financial loss, result in minor harm to individuals, or result in minor harm to organizational assets.

The Purdue University Information Technology Security and Privacy Office produced a security policy paper, which these examples are drawn from. Moderate it is possible that the loss may have a significant negative impact on an organization's operations, assets, or personnel. A loss may have a serious adverse effect if it results in one of the following: a significant degradation in mission capability, such that the organization can still perform its primary functions for a significant period of time, but those functions are significantly less effective; significant damage to organizational assets; significant financial loss; or significant harm to individuals that does not involve loss of life or serious injury.

High: It is possible that the loss may have a significant or catastrophic negative impact on an organization's operations, assets, or personnel. A severe or catastrophic adverse effect, on the other hand, denotes, for instance, that the loss might I cause a severe degradation in or loss of mission capability to the point and duration that the organization is unable to perform one or more of its primary functions, result in major damage to organizational assets, result in major financial loss, result in severe or catastrophic harm to people, including loss of life or serious, life-threatening injuries, or result in any

Confidentiality students place a high value on the secrecy of student grade information as a valuable resource. The Family Educational Rights and Privacy Act governs how such information is disclosed in the United States (FERPA). Only students, their parents, and staff who need the knowledge to do their jobs should have access to grade information. Information about students' enrolment may be somewhat secret. While still protected by FERPA, this information is more widely disseminated, less likely to be misused than grades, and would cause less harm if made public. The level of secrecy for directory information, such as student, teacher, or departmental listings, may be minimal or even zero. On a school's website, this information is often publicly accessible to the general public.

Integrity the example of the allergy data kept in a database for a hospital patient serves to highlight many facets of integrity. The data should be reliable and up-to-date so the doctor can rely on it. Consider a scenario where a worker like a nurse with access to this data willfully falsifies the information in order to hurt the hospital. The database has to be swiftly restored to a reliable foundation, and it should be feasible to identify the individual who made the mistake and hold them accountable. Information on a patient's allergies is an asset with a high need for integrity. Inaccurate information might cause a patient considerable damage or even death, opening the institution up to enormous responsibility.

An example of an asset that may be given a modest degree of integrity requirement is a website that provides registered users with a forum where they can debate a certain subject. A registered user or a hacker might alter certain entries or alter the website's appearance. Potential harm is not serious if the forum serves solely to provide members with entertainment, generates little to no cash from advertising, and is not used for anything significant like study. The webmaster might lose some time, money, and/or data.

An anonymous online survey is a prime illustration of a low-integrity demand. Many websites, including news organizations, provide their customers these polls with very minimal security measures. However, it is commonly known that such surveys are inaccurate and not based on science. Availability the needed degree of availability rises with the importance of a component or service. Think of a system that offers authentication services for important applications, devices, and systems. Customers cannot use computer resources during a service disruption, and personnel cannot access the resources they need to complete crucial jobs. Due to decreased staff productivity and probable client losses, the loss of the service results in a large financial loss[6], [7].

A public university Web site is an example of an asset that is often categorized as having a moderate availability requirement; the Web site offers information for present and potential students as well as benefactors. Although such a site is not essential to the university's information system, its absence will be embarrassing. An application that searches the phone book online would be categorized as having a low availability need. There are other methods to get the information, such as a hardcopy directory or the operator, so the brief loss of the program won't be as annoying. Security for computers and networks is both exciting and challenging. Some of the causes are as follows:

1. Security is more complicated than a beginner would first think. The requirements seem to be simple; in fact, the majority of the crucial criteria for security services can be summed up in one word, such as secrecy, authentication, nonrepudiation, and integrity. However, the methods used to satisfy those criteria may be rather complicated, and comprehending them may need for nuanced thinking.
2. Potential assaults on the security features must always be taken into account when creating a specific security mechanism or algorithm. In many instances, effective assaults are created by approaching the issue from a totally new perspective, thereby taking advantage of an unanticipated flaw in the system.
3. As a result of point 2, the methods employed to provide certain services are often illogical.
4. A security mechanism is often complicated, and it may not be clear from the description of a given demand that such sophisticated safeguards are necessary. Extensive security measures only make sense when the threat's many facets are taken into account.
5. After devising many security measures, it is required to choose their application. This applies logically [e.g., at what layer or levels of an architecture such as TCP/IP (Transmission Control Protocol/Internet Protocol) should mechanisms be put] as well as physically (e.g., at what points in a network are specific security measures required).
6. Security measures usually go beyond a specific method or protocol. Additionally, they pose concerns regarding the production, dissemination, and security of any secret information that is required for participation (such as an encryption key). Additionally, there can be a dependency on communications protocols, whose behavior might make it more difficult to create the security mechanism. For instance, any protocol or network that involves variable, unexpected delays may make any time limitations that must be imposed for a message's transit time from sender to recipient in order for the security mechanism to operate properly to be meaningless.
7. Finding flaws in computer and network security and trying to plug them is basically a game of wits between the culprit and the designer or administrator.
8. The attacker has the significant benefit of only needing to identify one vulnerability, while the designer must identify and fix every flaw to obtain complete security.
9. Users and system administrators have a natural inclination to undervalue the value of security investments up until a security failure occurs.
10. Due to the short-term, overburdened environment of today, security demands routine, if not continual, monitoring, which is challenging.
11. Rather from being an essential step in the design process, security is still much too often added to a system as an afterthought.
12. Many users (and even security managers) consider robust security to be a barrier to the effective and user-friendly usage of information or the administration of an information system.

We will confront the aforementioned challenges in a variety of ways as we explore the different security risks and processes throughout this book. The manager in charge of computer and network security requires a systematic method for establishing security requirements and

describing ways to achieving those requirements in order to analyze the security needs of a business, evaluate, and choose different security solutions and policies.

Even while this is challenging enough in a centralized data processing environment, using local and wide area networks only makes things worse. Such a methodical methodology is described in ITU-T4 Recommendation X.800, Security Architecture for OSI.The International Telecommunication Union's (ITU) Telecommunication Standardization Sector (ITU-T) is a UN-sponsored organization that creates recommendations for open systems interconnection and telecommunications standards (OSI). The OSI protocol architecture, which is detailed in Appendix D, served as the foundation for the development of the OSI security architecture. A grasp of the OSI protocol architecture is not necessary for the purposes of this chapter.

Security attacks, of the responsibility of coordinating security provision. Aside from that, computer and communications manufacturers have included security features for their goods and services that connect to this organized specification of mechanisms and services since this architecture was created as an international standard.

The OSI security architecture gives us a good, though abstract, overview of many of the ideas covered in this book. The emphasis of the OSI security architecture is on security methods, services, and attacks. These are briefly defined as any activity that jeopardizes the security of data that belongs to an organization is referred to as a security assault.

A procedure (or a device integrating such a process) that is intended to identify, stop, or recover from a security assault is known as a security mechanism. A processing or communication service that improves the security of an organization's data processing systems and information transfers. The services utilize one or more security mechanisms to offer the service and are designed to defend against security assaults.

Both X.800 and RFC 2828 employ passive attacks and active attacks as a good way to categorize security threats. A passive assault does not deplete system resources; instead, it tries to gather or utilize information from the system. An active assault tries to change system resources or interfere with their functionality. When there is a situation, capacity, action, or event that might violate security and cause damage, there is a potential for a security violation. In other words, a threat is a potential risk that might take advantage of a weakness an attack on system security caused by a sophisticated attacker. That is, an intelligent act that intentionally tries to circumvent security measures and breach a system's security policy, particularly in the sense of a method or methodology.

It is simple to understand how message contents are released. Sensitive or private information may be present in a phone call, an email, or a shared file. We want to keep an adversary from finding out what is being sent. Imagine if we could conceal the information contained in messages or other information flow, preventing adversaries from using it against us even if they managed to intercept the transmission. Encryption is a typical method for hiding information. Even with encryption, an adversary could still be able to track the trend of these transmissions. The adversary may ascertain the whereabouts and identities of hosts that were in communication as well as their frequency and message length.

Due to the lack of data modification, passive assaults are exceedingly difficult to identify. In most cases, the message traffic seems to be delivered and received normally, and neither the sender nor the recipient is aware that a third party has read the messages or tracked the traffic pattern. Nevertheless, it is possible to stop the success of these assaults, often through encryption. Therefore, preventing passive assaults is more important than detecting them when dealing with them.

Masquerade, replay, message modification, and denial of service are the four subcategories of active assaults, which entail some alteration of the data stream or the generation of a fake stream. There is a masquerade when one thing impersonates another thing. One of the other active attack types is often present in a masquerade assault. For instance, authentication sequences may be recorded and replayed after a successful authentication sequence, allowing an authorized entity with limited rights to impersonate an entity with those privileges to get further privileges[8], [9].

Modification of messages simply refers to the alteration of a piece of a lawful communication, as well as the delaying or rearrangement of messages in order to have an unauthorized effect. A message that was originally intended to say "Allow John Smith to view confidential file accounts" gets changed to say "Allow Fred Brown to read confidential file accounts," for instance.

When a service is denied, it prohibits or hinders the regular usage or administration of communications infrastructure. This assault could be aimed against a particular person or thing; for instance, something might silence all signals going to a certain place (e.g., the security audit service). Disrupting a whole network either by turning it off or by sending it too many messages to improve performance is another example of service denial stop them from succeeding. On the other hand, since there are so many different possible physical, software, and network vulnerabilities, it is very difficult to completely avoid active assaults. Instead, the objective is to identify ongoing assaults and recover from whatever delays or disruptions they may have created. If detection acts as a deterrent, it could also help with prevention.

According to the X.800 standard, a security service is one that is offered by the protocol layer of open systems that are interacting and assures acceptable security of the systems or of data transfers. RFC 2828, which offers the following definition, may provide a clearer one: The provision of a processing or communication service by 14 Security services carry out security policies and are carried out by security mechanisms in order to provide a certain kind of protection to system resources.

Many of the words used in security literature are not universally accepted. For instance, all facets of information security may sometimes be referred to as having integrity. Both identity verification and the different tasks outlined under integrity in this chapter are referred to as "authentication" at times. Our use here is compliant with RFC 2828 and X.800 the avoidance of illegal use of a resource i.e., this service regulates who has access to a resource, under what circumstances access may happen, and what those with access are permitted to do preventing unlawful exposure of data

Ensures the integrity of all user data on a connection and attempts to retrieve any data that has been modified, added, deleted, or replayed across an entire data sequence. Integrity of Connections without Recovery Similar as above, but just offers detection; no recovery is offered. Providing for the integrity of certain fields within user data of a data block transported across a connection, selective-field connection integrity determines if the specific fields have been changed, added, removed, or replayed. Single connectionless data block integrity is provided for, and may take the form of data alteration detection. A constrained kind of replay detection might also be offered.

Providing for the integrity of certain fields inside a single connectionless data block, selective-field connectionless integrity determines if the specific fields have been changed protects against the refusal of participation in all or a portion of a communication by one of the parties concerned. Origin Proof and Nonrepudiation ensure that the communication was transmitted by the designated person. Nonrepudiation, or destination proof, shows that the intended recipient of the communication really got it.

A communication's authenticity is ensured by the authentication service. The purpose of the authentication service is to reassure the receiver that the message is from the source that it purports to be from in the event of a single message, such as a warning or alarm signal. There are two factors at play when there is a continuous contact, such when a terminal is connected to a host. First, the service verifies the authenticity of the two entities at the moment of connection commencement that is, that each is the entity that it claims to be. Second, the service must ensure that the connection is not tampered with in a manner that allows a third party to pretend to be one of the two legitimate parties in order to send or receive data that is not permitted.

Peer entity authentication enables the verification of a peer entity's identification inside an association. If two entities use distinct implementations of the same protocol (such as two TCP modules in two interacting systems), they are regarded as peers. Peer entity authentication is offered for usage during the connection's formation or data transmission phase. It makes an effort to provide assurance that a given entity isn't engaging in either a masquerade or an unapproved replay of a prior connection.

Data origin authentication: enables the verification of a data unit's source. It doesn't provide defense against data unit duplication or change. Applications like electronic mail, in which the parties conversing have never met before, are supported by this kind of service. The capacity to restrict and regulate access to host systems and applications over communications channels is known as access control in the context of network security. In order for access privileges to be customized to the person, each entity attempting to get access must first be identified, or authenticated.

Transmitted data is shielded from passive assaults by confidentiality. There are many tiers of security that may be assigned to data transmissions depending on their content. The most comprehensive solution safeguards all user data sent between two users over time. This extensive security, for instance, forbids the disclosure of any user data exchanged across a TCP connection between two computers. It is also possible to build more particular variations of this service,

such as the security of a single message or even only certain message fields. These improvements are less helpful than the general strategy and can even be more difficult and costly to put into practice.

The shielding of traffic flow from analysis is the other part of secrecy. An attacker must be unable to see the source and destination, frequency, duration, or any other aspects of the traffic on a communications infrastructure in order to do this integrity, like confidentiality, may be applied to a single message, a stream of messages, or certain fields within a message. Again, complete stream protection is the best and most obvious solution.

A connection-oriented integrity service handles a stream of messages and ensures that there is no replaying, duplicate, insertion, modification, or change to the messages after they have been delivered. This service also includes coverage for data deletion. As a result, both message stream modification and denial of service are addressed by the connection-oriented integrity service. On the other hand, a connectionless integrity service merely offers protection against message tampering and solely deals with individual messages without taking into account any wider context.

We can distinguish between service with recovery and service without recovery. We are more interested in detection than prevention since the integrity service pertains to current assaults. If an integrity breach is found, the service may just notify the user of the violation; further software or human action may be needed to fix the problem. As we shall discuss later, there are other methods for recovering from the loss of data integrity. The more desirable option is usually to include automatic recovery procedures.

A communication cannot be denied by the sender or the recipient due to nonrepudiation. As a result, when a communication is transmitted, the recipient may demonstrate that the message was indeed sent by the supposed sender. Similar to this, when a communication is sent, the sender may demonstrate that the purported recipient really got the message.

According to the performance requirements for the system, both X.800 and RFC 2828 define availability as the quality of a system or system resource being accessible and usable upon demand by an authorized system entity (i.e., a system is available if it offers services in accordance with the system design whenever users request them). Availability may be lost or reduced as a consequence of many assaults. In order to avoid or recover from the loss of availability of components of a distributed system, some of these assaults are susceptible to automated countermeasures, such as authentication and encryption.

The availability of different security services is treated as a property by X.800. It seems appropriate to include an availability service in particular, however. A service that ensures a system's availability is an availability service. Denial-of-service attacks present security issues, which are addressed by this service. Access control services and other security services are necessary for appropriate administration and control of system resources[10], [11].

The mechanisms are separated between those that are implemented in a specific protocol layer, such TCP or an application-layer protocol, and others that are not unique to any certain protocol

layer or security the process of transforming facts using mathematical formulas into a form that is difficult to understand. An algorithm and one or more encryption keys are required for the data to be transformed and then recovered.A receiver of the data unit may use data attached to it or a cryptographic modification of it to confirm its origin and integrity and prevent tampering (e.g., by the recipient)a wide range of tools for enforcing resource access permissions.

Data Reliability several different methods for guaranteeing the integrity of a data unit or stream of data units. Exchange of information with the aim of ensuring an entity's authenticity via authentication the act of inserting bits into data stream gaps to thwart traffic analysis techniques. Routing Control permits the choice of specific, physically secure pathways for certain data and permits routing adjustments, particularly when a security breach is detected using a dependable outsider to guarantee certain characteristics of a data exchange

Mechanisms that do not belong to any one layer or security service in the OSI model dependable functionality anything which is considered accurate in light of a certain criterion (e.g., as established by a security policy). We won't go into detail about these methods now; instead, we'll address them where they belong in the book. We'll just make a brief observation on what and decipherment is. Reversible and irreversible decipherment techniques are distinguished by the X.800 standard. An encryption procedure that enables data to be encrypted and then later decrypted is known as a reversible decipherment mechanism. Hash algorithms and message authentication codes are examples of irreversible decipherment techniques that are utilized in digital signature and message authentication applications.

REFERENCES:

[1] P. Line, C. Committee, and I. Communications, *IEEE Standard for Smart Energy Profile Application Protocol*. 2018.

[2] S. Sinha, *Beginning ethical hacking with Kali Linux: Computational techniques for resolving security issues*. Berkeley, CA, CA: Apress, 2018. doi: 10.1007/978-1-4842-3891-2.

[3] M. G. Moreira Santos and P. A. Alcívar Marcillo, “Security in the data link layer of the OSI model on LANs wired Cisco,” *J. Sci. Res. Rev. Cienc. e Investig.*, 2018, doi: 10.26910/issn.2528-8083vol3isscitt2017.2018pp106-112.

[4] L. B. Gurajada, S. D. Rajaputra, I. Gogineni, and R. Shaik, “Security attacks in wireless sensor networks,” *Int. J. Eng. Technol.*, 2018, doi: 10.4018/978-1-61350-101-6.ch706.

[5] A. Sundararajan, A. Chavan, D. Saleem, and A. I. Sarwat, “A survey of protocol-level challenges and solutions for distributed energy resource cyber-physical security,” *Energies*, 2018, doi: 10.3390/en11092360.

[6] B. Mostefa and G. Abdelkader, “A survey of wireless sensor network security in the context of Internet of Things,” in *Proceedings of the 2017 4th International Conference on Information and Communication Technologies for Disaster Management, ICT-DM 2017*, 2018. doi: 10.1109/ICT-DM.2017.8275691.

[7] R. Shaik and S. S. Ahamad, "Security attacks and challenges of wireless sensor network's a review," *Int. J. Eng. Technol.*, 2018, doi: 10.14419/ijet.v7i2.29.13144.

[8] A. R. Chordiya, S. Majumder, and A. Y. Javaid, "Man-in-the-Middle (MITM) Attack Based Hijacking of HTTP Traffic Using Open Source Tools," in *IEEE International Conference on Electro Information Technology*, 2018. doi: 10.1109/EIT.2018.8500144.

[9] F. Waheed and M. Ali, "Hardening cisco devices based on cryptography and security protocols-part ii: Implementation and evaluation," *Ann. Emerg. Technol. Comput.*, 2018, doi: 10.33166/AETiC.2018.04.002.

[10] S. Wilson, N. Moustafa, and E. Sitnikova, "A digital identity stack to improve privacy in the IoT," in *IEEE World Forum on Internet of Things, WF-IoT 2018 - Proceedings*, 2018. doi: 10.1109/WF-IoT.2018.8355199.

[11] A. C. Zamfira and H. Ciocarlie, "Developing an ontology of cyber-operations in networks of computers," in *Proceedings - 2018 IEEE 14th International Conference on Intelligent Computer Communication and Processing, ICCP 2018*, 2018. doi: 10.1109/ICCP.2018.8516644.

CHAPTER 4

A MODEL FOR NETWORK SECURITY

Mr. Raghavendra T. S., Assistant Professor
Department of Computer Science and Engineering, Presidency University, Bangalore, India
Email Id- raghavendra@presidencyuniversity.in

Network security is an essential component of any organization's IT infrastructure. It refers to the policies, processes, and technologies that are designed to protect a network from unauthorized access, theft, and damage to its components. In this article, we will outline a model for network security that organizations can use to safeguard their networks against various cyber threats.

1. **Risk Assessment:** The first step in securing a network is to conduct a comprehensive risk assessment. This involves identifying potential risks and vulnerabilities in the network, assessing the likelihood and impact of these risks, and prioritizing them based on their severity. Risk assessment helps organizations identify potential threats and vulnerabilities and develop appropriate security measures to mitigate them.

2. **Network Design:** Once the risks and vulnerabilities have been identified, the next step is to design a secure network architecture. This involves developing a network topology that segregates the network into different zones and implementing various security measures, such as firewalls, intrusion detection systems, and access controls. The network design should be based on industry best practices and take into account the specific security needs of the organization.

3. **Access Control:** Access control is an essential component of network security, and it involves controlling who can access the network and what resources they can access. Access control can be implemented at various levels, including network, application, and data levels. Access controls should be based on the principle of least privilege, which means that users should only have access to the resources they need to perform their jobs.

4. **Encryption:** Encryption is a critical technology for securing network communications. It involves transforming plaintext data into ciphertext that can only be decrypted with a key. Encryption can be used to protect data in transit over the network, as well as data at rest on storage devices. The encryption algorithm should be chosen based on its strength, speed, and suitability for the specific use case.

5. **Monitoring and Detection:** Continuous monitoring and detection of network activity is critical for identifying potential security incidents and responding to them in a timely manner. Monitoring can be done at various levels, including network traffic, system logs, and user activity. Detection can be done using various tools, including intrusion detection systems, security information and event management (SIEM) systems, and endpoint detection and response (EDR) systems.

6. **Incident Response:** Incident response is the process of identifying, analyzing, and responding to security incidents. This includes containing the incident, analyzing the scope and impact, and remediating any damage. Incident response should be based on a predefined plan that outlines the roles and responsibilities of different stakeholders, the procedures for communication and coordination, and the steps for escalation and resolution.
7. **Testing and Review:** Regular testing and review of the network security model are critical for ensuring its effectiveness and identifying any weaknesses or vulnerabilities. This includes vulnerability scanning, penetration testing, and security audits. Testing and review should be done regularly and should be based on industry best practices and regulatory requirements[1], [2].

In conclusion, a model for network security should include a comprehensive risk assessment, a secure network design, access controls, encryption, continuous monitoring and detection, incident response, and regular testing and review. By implementing these measures, organizations can safeguard their networks against various cyber threats and ensure the confidentiality, integrity, and availability of their data and resources.

Network security is a vital concern for organizations in today's interconnected world, where the risk of cyber threats is constantly increasing. Hackers and other malicious actors are always looking for ways to exploit vulnerabilities in network infrastructure to steal sensitive data, disrupt operations, or cause other forms of damage. As a result, organizations must implement a model for network security that is robust, effective, and up-to-date. In this article, we will go into more detail on each of the components of the network security model outlined in the previous section.

1. Risk Assessment a risk assessment is a critical first step in developing a network security model. It involves identifying potential risks and vulnerabilities in the network infrastructure, assessing their likelihood and potential impact, and prioritizing them based on their severity. A risk assessment helps organizations to understand their security posture and make informed decisions about where to allocate resources for security measures.

Risk assessments can be conducted by internal security teams or by external consultants who specialize in this field. The assessment typically involves identifying potential threats, such as malware, phishing, or denial of service attacks, and vulnerabilities, such as outdated software or weak passwords. The assessment team will also evaluate the organization's security policies and procedures and assess their effectiveness.

2. Network Design Once the risks and vulnerabilities have been identified, the next step is to design a secure network architecture. This involves developing a network topology that segregates the network into different zones and implementing various security measures, such as firewalls, intrusion detection systems, and access controls. The network design should be based on industry best practices and take into account the specific security needs of the organization.

One of the most important principles in network design is the concept of defense in depth, which involves implementing multiple layers of security controls to protect the network. For example, a network might have a perimeter firewall to block unauthorized access, an intrusion detection system to identify potential attacks, and access controls to limit user access to sensitive resources.

3. Access Control Access control is another essential component of network security. It involves controlling who can access the network and what resources they can access. Access controls can be implemented at various levels, including network, application, and data levels. Access controls should be based on the principle of least privilege, which means that users should only have access to the resources they need to perform their jobs.

Access control mechanisms can include authentication and authorization, such as username and password combinations, multi-factor authentication, and role-based access control. It is also important to enforce access control policies across all devices and applications that are connected to the network, including mobile devices and cloud services.

4. Encryption is a critical technology for securing network communications. It involves transforming plaintext data into cipher text that can only be decrypted with a key. Encryption can be used to protect data in transit over the network, as well as data at rest on storage devices. The encryption algorithm should be chosen based on its strength, speed, and suitability for the specific use case.

For example, network traffic can be encrypted using protocols such as SSL/TLS or IPsec, while data at rest can be encrypted using tools such as BitLocker or Vera Crypt. Encryption keys should be managed securely and kept separate from the data they protect.

5. Monitoring and Detection Continuous monitoring and detection of network activity is critical for identifying potential security incidents and responding to them in a timely manner. Monitoring can be done at various levels, including network traffic, system logs, and user activity. Detection can be done using various tools, including intrusion detection systems, security information and event management (SIEM) systems, and endpoint detection and response (EDR) systems.

The purpose of monitoring and detection is to identify potential security incidents, such as unauthorized access attempts or malware infections, as soon as possible. The faster an incident is detected, the faster it can be contained and remediated, reducing the potential impact of the incident.

6. Incident Response Incident response is the process of responding to a security incident in a coordinated and effective manner. The goal of incident response is to minimize the damage caused by the incident and restore normal operations as quickly as possible. Incident response involves several key stages, including detection, containment, analysis, eradication, and recovery.

The first step in incident response is to detect the incident, which is typically done through monitoring and detection mechanisms. Once an incident has been detected, it is important to

contain the incident to prevent further damage. This can involve isolating affected systems, disconnecting from the network, and disabling compromised user accounts.

The next step is to analyze the incident to understand how it occurred and what data was affected. This can involve reviewing system logs, analyzing malware samples, and interviewing affected users. Once the incident has been analyzed, it is time to eradicate the threat by removing malware, patching vulnerabilities, and resetting compromised passwords.

After the threat has been eradicated, the focus shifts to recovery. This involves restoring affected systems and data to their pre-incident state. It is also important to conduct a post-incident review to identify lessons learned and make improvements to the security model.

7. Training and Awareness Training and awareness are critical components of network security. Employees are often the weakest link in the security chain, as they may not be aware of the risks associated with certain behaviors or may not know how to identify and report potential security incidents. Training and awareness programs should be developed and implemented to educate employees on how to protect sensitive data, identify potential threats, and respond to security incidents.

Training can include topics such as password security, phishing awareness, and data protection policies. It is also important to provide regular updates and reminders to employees to reinforce the importance of security best practices.

A comprehensive model for network security is essential for protecting organizations from cyber threats. The model should include components such as risk assessment, network design, access control, encryption, monitoring and detection, incident response, and training and awareness. Implementing a robust network security model can help organizations to prevent security incidents, respond effectively to incidents that do occur, and minimize the impact of incidents on the organization. By making network security a priority and regularly reviewing and updating the security model, organizations can stay ahead of the evolving threat landscape and protect their data and operations.

8. Continuous Improvement Network security is an ongoing process, and it is important to regularly review and update the security model to ensure that it remains effective. Threats and vulnerabilities are constantly evolving, so it is important to stay up-to-date on the latest security trends and technologies. Regular risk assessments can help to identify new or emerging risks, and adjustments can be made to the security model as needed.

In addition to monitoring the threat landscape, it is important to track and analyze security incidents to identify areas for improvement. Post-incident reviews can help to identify weaknesses in the security model and highlight areas for improvement. By regularly reviewing and updating the security model, organizations can ensure that they are well-positioned to prevent and respond to security incidents.

9. Compliance with applicable laws and regulations is a critical component of network security. Depending on the industry, organizations may be subject to a variety of regulatory requirements, such as HIPAA for healthcare organizations or PCI DSS for

payment card industry merchants. Compliance with these regulations can help to ensure that sensitive data is protected and that the organization is not subject to fines or legal action.

Compliance with regulations typically involves implementing specific security controls and policies. For example, HIPAA requires the implementation of administrative, physical, and technical safeguards to protect electronic protected health information (ePHI). Organizations that fail to comply with applicable regulations may face significant financial and reputational harm.

10. Vendor Management Finally, vendor management is an important component of network security. Many organizations rely on third-party vendors for a variety of services, such as cloud hosting, software development, or payment processing. It is important to ensure that these vendors have appropriate security controls in place to protect sensitive data.

Vendor management typically involves conducting due diligence on vendors before engaging in a business relationship, and regularly monitoring vendors to ensure that they remain in compliance with security requirements. Contracts with vendors should include clear language regarding data protection and security, and vendors should be required to report any security incidents or breaches in a timely manner.

A comprehensive model for network security should include risk assessment, network design, access control, encryption, monitoring and detection, incident response, training and awareness, continuous improvement, compliance, and vendor management. By implementing a robust security model that addresses each of these components, organizations can better protect their data and operations from cyber threats.

It is important to regularly review and update the security model to ensure that it remains effective in the face of evolving threats and to comply with applicable laws and regulations. Finally, vendor management is a critical component of network security, and organizations must ensure that third-party vendors have appropriate security controls in place to protect sensitive data.

It is necessary for a message to be sent from one party to another via an Internet service. The two parties, who constitute the transaction's principals, are required to cooperation is required for the trade to happen. A logical information channel is created by specifying a path over the Internet from source to destination and by the two principals working together to employ communication protocols (like TCP/IP)[3], [4].

When it is required or desired to secure the transmission of information from an adversary who may pose a danger to confidentiality, authenticity, etc., security considerations come into play. There are two elements common to all security methods: An alteration to the supplied data that relates to security. Examples include the insertion of a code depending on the contents of the message, which may be used to validate the identity of the sender, and the encryption of the message, which scrambles it so that it is unreadable by the adversary. Some top-secret knowledge that the two leaders discussed that, it is hoped, the rival was unaware of. An example

is the employment of an encryption key together with a transformation to scramble a message before transmission and unscramble it after receipt.

It could be necessary to use a reliable third party to ensure secure transmission. For instance, the confidential knowledge may be disseminated by a third party. The topic of symmetric encryption, which only requires one of the two principals to possess the secret information, is covered in Part Two leaders while securing it from competitors. Alternatively, a third party would be required to settle disagreements between the two protagonists over the veracity of a message transmission.

This broad model demonstrates the four fundamental activities involved in building a specific security service:

1. Create an algorithm to carry out the transformation that relates to security. An adversary should not be able to subvert the algorithm's goals.
2. Produce the secret data that will be combined with the algorithm.
3. Create strategies for the secret information's dissemination and exchange.
4. Specify a protocol that the two principals will follow to implement a specific security service using the secret information and the security algorithm.

This book's first through fifth parts are devoted to describing the various security tools and services that fall inside the framework. There are additional interesting security-related circumstances, too, that do not perfectly match this paradigm but are taken into account in this work. The majority of readers are aware of the dangers posed by hackers, who try to get into systems accessible across a network. A hacker is anybody who enjoys hacking into a computer system without having any malicious intentions. An employee who wants to cause trouble or a criminal who wants to use computer resources for personal gain might both be the invader (e.g., obtaining credit card numbers or performing illegal money transfers).

Another kind of unauthorized access is the insertion of logic into a computer system that takes advantage of flaws in the system and affects both application programs and utility applications, such editors and compilers. Programs may pose two different types of threats: Threats to information access include the interception of data or its modification on behalf of people who shouldn't have access to it. Software assaults include, for example, worms and viruses. A disk containing the undesirable logic cloaked in otherwise helpful software may be used to launch such assaults onto a system. They can also be injected into a system across a network, however network security is mainly concerned with the latter approach.

The security measures required to handle unauthorized access may be divided into two groups. One may refer to the first group as a gatekeeper function. It has screening logic that can identify and reject worms, viruses, and other similar assaults, as well as password-based login protocols that are intended to prevent access from being granted to anybody other than authorized users. The second line of defense consists of a range of internal controls that monitor activities and examine stored data in an effort to find the existence of unwelcome invaders once either an unauthorized person or unauthorized software obtains access. Part Six explores these concerns.

Physical Security Physical security is an often-overlooked component of network security, but it is just as important as the technical components. Physical security involves controlling access to the physical components of the network, such as servers, switches, and routers. Physical security measures can include surveillance cameras, access control systems, and alarm systems. Access to server rooms and other secure areas should be restricted to authorized personnel only, and equipment should be secured to prevent theft or tampering.

Cloud Security Many organizations are moving their data and applications to the cloud, but this shift can create new security risks. Cloud providers have their own security measures in place, but it is important for organizations to understand their own security responsibilities and implement additional measures to protect their data in the cloud. Cloud security measures can include encryption of data at rest and in transit, access controls, and monitoring and detection. It is also important to regularly review and update cloud security policies and procedures to ensure that they remain effective in the face of evolving threats.

Incident Response planning despite best efforts to prevent security incidents, they can still occur. Organizations should have an incident response plan in place to ensure that they can quickly and effectively respond to a security incident and minimize the damage.Incident response plans should include procedures for detecting, analyzing, containing, and remedying security incidents. They should also identify key stakeholders and outline communication plans to keep all parties informed during an incident.

Supply Chain Security The supply chain is an often-overlooked component of network security, but it is critical to ensuring the integrity of the network. Supply chain security involves ensuring that all components of the network, including hardware and software, are purchased from trusted vendors and that they are free from security vulnerabilities. Organizations should have a vendor management program in place to ensure that vendors are properly vetted and that their products and services meet the organization's security requirements. Supply chain security should be considered throughout the procurement process, from initial vendor selection to ongoing monitoring and evaluation[5], [6].

Continuous Improvement Network security is not a one-time project, but an ongoing process that requires continuous improvement. It is important to regularly review and update security policies and procedures to ensure that they remain effective in the face of evolving threats. Organizations should also regularly assess the effectiveness of their security measures and make changes as needed. This can include conducting vulnerability assessments and penetration testing, as well as monitoring and analyzing security logs and other data[7], [8].

Continuous improvement also involves staying up to date on the latest security threats and technologies. Organizations should stay informed about emerging threats and new security technologies and incorporate them into their security model as appropriate. Network security is a complex and multifaceted issue that requires a comprehensive approach. By addressing each of the components discussed above, organizations can better protect their data and operations from cyber threats. It is important to regularly review and update the security model to ensure that it remains effective in the face of evolving threats and to comply with applicable laws and

regulations. Finally, testing and validation, physical security, cloud security, incident response planning, supply chain security, and continuous improvement are all critical components of network security.

REFERENCES:

[1] Z. Ni, Q. Li, and G. Liu, "Game-Model-Based Network Security Risk Control," *Computer (Long. Beach. Calif).*, 2018, doi: 10.1109/MC.2018.2141032.

[2] M. A. Naagas, E. L. Mique, T. D. Palaoag, and J. S. Dela Cruz, "Defense-through-deception network security model: Securing university campus network from DOS/DDOS attack," *Bull. Electr. Eng. Informatics*, 2018, doi: 10.11591/eei.v7i4.1349.

[3] L. Zhou, W. Guo, J. Y. Wang, and J. T. Huang, "Network security evaluation model based on neural network algorithm," *Shenyang Gongye Daxue Xuebao/Journal Shenyang Univ. Technol.*, 2018, doi: 10.7688/j.issn.1000-1646.2018.04.12.

[4] X.-L. ZHAO, Y.-M. ZHANG, H. YA, X.-H. ZHANG, and Y.-N. YANG, "Multi-Layer, Multi-Dimensional and Multi-Granularity Network Model to Measure Network Security," *DEStech Trans. Comput. Sci. Eng.*, 2018, doi: 10.12783/dtcse/cimns2017/17401.

[5] N. Zhang, "Defensive strategy selection based on attack-defense game model in network security," *Int. J. Performability Eng.*, 2018, doi: 10.23940/ijpe.18.11.p9.26332642.

[6] R. Alguliyev, R. Aliguliyev, and F. Yusifov, "MCDM Model for Evaluation of Social Network Security Threats," *18th Eur. Conf. Digit. Gov. (ECDG 18)*, 2018.

[7] J. Zhu, "Wireless sensor network technology based on security trust evaluation model," *Int. J. Online Eng.*, 2018, doi: 10.3991/ijoe.v14i04.8590.

[8] S. Rehman and V. Gruhn, "An Effective Security Requirements Engineering Framework for Cyber-Physical Systems," *Technologies*, 2018, doi: 10.3390/technologies6030065.

CHAPTER 5

NETWORK TOPOLOGY

Mr. Prakash Metre, Assistant Professor
Department of Computer Science and Engineering, Presidency University, Bangalore, India
Email Id- prakashbmetre@presidencyuniversity.in

All types of computer networks, including LANs, MANs, and WANs, are built using a topology. The following are some of the most well-known topologies. In contrast to previous topologies, a mesh topology permits various access links between network components. Because there are several access lines connecting the network parts, network dependability is improved anytime one network when element fails, the network doesn't stop working; instead, it just finds a way around the problem and keeps going. In MAN, mesh topology is most frequently used.

The tree topology is a more typical kind of network topology. In the tree topology, network elements are arranged in a hierarchical form with the most dominating element serving as the tree's root and all other network elements sharing a common structure. In the parent-child connection, there are no closed loops, just like in regular trees. Accordingly, dealing with network element failures might be difficult depending on where the element is located within the hierarchy. For instance, if the root element in a deeply rooted tree fails, the network immediately ruptures.

Network client-server information exchange is one of the most crucial helpful areas in network performance where scripting plays a crucial role. A Common Gateway Interface, or CGI, is used to do this. CGI is a specification for a database schema that Information exchange requires the usage of servers, browsers, and software. CGI scripts are software created in any language that uses this protocol to transfer data from a Web server to a client's browser. In other terms, a CGI script acts as an external gateway application to communicate with client browsers and information servers like HTTP or Web servers. CGI scripts are fantastic because they enable Web servers to communicate and be dynamic with client browsers. CGI scripts are fantastic because they enable Web servers to be dynamic and interactive with client browsers as the server accepts user inputs and replies to them in a controlled and pertinent manner to please the user Without CGI, any information that users get from a server would not be packed according to their requests, but rather according to how the information is kept on the server.

CGI applications may be divided into two categories: scripts written in scripting languages like PERL, Java, and UNIX, and programmers written in programming languages like C/C++ and FORTRAN that can be compiled to create an executable module stored on the server shell. Unlike CGI applications, these CGI scripts don't require the information server to keep any related source code. The compilation of CGI scripts written in scripting languages differs from that of non-scripting languages. Instead, they are written code that is immediately interpreted and

executed by the interpreter on the information server or in the browser. This has the benefit that you can copy your script to any system using the same interpreter with little to no changes.

When run at the information server, CGI scripts and programmers both assist in organizing data for the server and the client. For instance, the server could wish to receive visitor data and utilize it to compile an appropriate for the client's product. GCI can also be used to dynamically set field descriptions on a form and notify the user in real-time of the data that has been entered and is still to be entered. Even after completion, the form could be sent back to the user for final editing before submission. In addition to automating a variety of services in search engines and directories, CGI scripts go beyond dynamic form filling to handle routine tasks like making downloads available and providing access[1], [2].

CGI scripts may be created in any programming language that an information server can use, as we previously said. Scripting languages like Perl, JavaScript, TCL, Apple script, UNIX shell, VBScript, and others are present in many of these languages. C/C++, FORTRAN, and non-script languages like Visual Basic are examples. The languages themselves are dynamic, thus new languages might emerge shortly. The same manners we employ when we meet a stranger are used to begin contact between a server and a client. Before making any demands, a trusting connection must be built first. This can be accomplished in a variety of ways. Some individuals introduce themselves with a formal "Hello, I'm...," followed by "I require..." The stranger then responds with "Hello, I'm..." and "Sure I can...." Others extend it further to embraces, kisses, and any other icebreakers that may be used. In the form of an acknowledgement of your initial hug, the stranger will let you know if they are ready to make a request. However, there is typically no acknowledgement of your presence if the stranger is not prepared to speak to you.

These etiquette patterns and regulations are followed by computers while they are conversing, and we refer to this process as a handshake. It is referred to as a three-way handshake for computers. The client initiates a three-way handshake by sending the client and server addresses are included in a packet known as an SYN (short for synchronize), along with some preliminary information for introductions. The server opens a communication socket with the same port number that the client requested upon receiving this packet through its open welcome port, which is where all subsequent communication with the client will take place. Following the creation of the communication socket, the server places the connection in a queue and notifies the client by delivering an SYN-ACK, or sync-ask, acknowledgement.

The SYN request for a service from the client is sent to the CGI script, a server-side language that lives on the server. The script then starts running, enabling direct communication between the server and the client. In this situation, the script can data between the client and server received and sent on demand. The client browser is unaware that a script is being run on the server. The server adds the required protocol data after receiving the script's output and sends the packet (or packets) back to the client's browser. The location of a CGI script in a three-way handshake because the server is located on a computer, the CGI scripts are located there.

One can only examine the result of the script after it executes on the server and sends the output using a browser on the client computer the user is on since a user on a client machine cannot execute the script in a browser on the server.We said that in the interaction between the client

and the server, the CGI script is on the server side. To reply to client requests, the server executes the scripts that are stored on the server. Consequently, there is

An interface that keeps the script and server apart. This interface, as seen in, consists of data supplied by the server to the script, including input variables taken from an HTTP header provided by the client, and data sent back to the server by the script. Environment variables and script command lines are used to transfer output data from the server to the script and from the script to the server. Command line inputs direct a script to do certain actions.

Network topology refers to the physical or logical arrangement of devices and connections in a computer network. It determines how data flows between devices and how communication is established between them. There are several types of network topologies, each with its own advantages and disadvantages.

1. **Bus Topology:** In a bus topology, all devices are connected to a single cable called the bus. Data is transmitted along the bus to all connected devices. This type of topology is simple and inexpensive to implement, but if the bus cable fails, the entire network goes down. It's also susceptible to collisions, which can slow down the network.
2. **Ring Topology:** In a ring topology, devices are connected in a ring or circular layout. Data is transmitted around the ring in one direction. Each device in the ring is responsible for transmitting data to the next device in the ring until the data reaches its destination. This topology is more reliable than a bus topology since each device acts as a repeater, but it's also susceptible to failure if any device fails.
3. **Star Topology:** In a star topology, devices are connected to a central hub or switch. The hub or switch manages the flow of data between devices. If a device fails, only that device is affected, and the rest of the network remains unaffected. However, this topology is more expensive to implement than a bus or ring topology.
4. **Mesh Topology:** In a mesh topology, each device is connected to every other device in the network. This type of topology provides redundancy, so if one connection fails, data can be routed through another path. However, it's the most expensive and complex topology to implement.
5. **Tree Topology:** In a tree topology, devices are arranged in a hierarchical tree structure. The root of the tree is connected to a switch or hub, and branches off to smaller sub-trees. This type of topology is useful for large networks that need to be divided into smaller subnetworks. However, it's more complex than a star or bus topology.
6. **Hybrid Topology:** A hybrid topology is a combination of two or more topologies. For example, a network might have a central hub connected to several switches in a star topology, with each switch connecting to several devices in a bus topology. This type of topology provides flexibility and can be tailored to the specific needs of a network[3], [4].

In addition to physical topologies, there are also logical topologies, which determine how data flows between devices. Two common logical topologies are:

1. **Broadcast Topology:** In a broadcast topology, data is transmitted to all devices on the network. This type of topology is useful for applications like video streaming or online gaming, where all devices need to receive the same data.
2. **Point-to-Point Topology:** In a point-to-point topology, data is transmitted between two specific devices. This type of topology is useful for applications like email or file sharing, where data is sent between two specific devices.

In summary, network topology is an important consideration when designing a computer network. The choice of topology depends on factors like the size of the network, the type of data being transmitted, and the level of redundancy required. While there are several types of topologies to choose from, each has its own strengths and weaknesses, and it's important to choose the one that best fits the needs of the network.

Topology is a crucial element of any network design, as it dictates how data flows between devices and how those devices are connected to each other. In this regard, a network's topology is similar to a map or a blueprint, as it provides an overview of the network's architecture and helps to guide network administrators in their decision-making.

One of the primary considerations when choosing a network topology is the network's size. For small networks with a few devices, a simple topology like a bus or star topology may suffice. These topologies are easy to set up and maintain, and they provide a basic level of connectivity between devices. However, as the network grows in size and complexity, more sophisticated topologies like mesh or tree topologies may be necessary.

Another factor to consider when choosing a topology is the type of data being transmitted on the network. Different applications and protocols have different requirements for how data is transmitted, and some topologies may be better suited to certain types of data than others. For example, a network that primarily handles video streaming or online gaming may benefit from a broadcast topology, which allows all devices on the network to receive the same data simultaneously.

A related consideration is the level of redundancy required in the network. Redundancy refers to the use of multiple paths for data transmission, which can help to ensure that data is delivered even if one path fails. Mesh topologies are often preferred for networks that require high levels of redundancy, as they provide multiple paths for data transmission between devices.

Security is another important consideration when choosing a network topology. Certain topologies, like bus or ring topologies, are more susceptible to security breaches than others. For example, in a bus topology, any device can "listen in" on data transmissions along the bus. In contrast, a star topology provides greater security by limiting access to the network to devices that are directly connected to the central hub or switch.

When designing a network topology, it's also important to consider the physical environment in which the network will be deployed. Factors like the distance between devices, the location of

network closets or server rooms, and the availability of power and cooling all play a role in determining the most appropriate topology for a given network. For example, in a large office building, a tree topology may be used to divide the network into smaller subnetworks on each floor, while a mesh topology may be used to provide redundancy between floors.

Finally, it's worth noting that a network's topology is not necessarily fixed or permanent. As the network's needs evolve over time, it may be necessary to reconfigure or even replace the existing topology. This is especially true for networks that are expanding rapidly or that are being updated to support new applications or protocols. In such cases, it's important to have a flexible network design that can adapt to changing requirements[5], [6].

Network topology is a critical element of any network design. It determines how data flows between devices, how those devices are connected to each other, and how the network is organized. When choosing a topology, network administrators must consider factors like the network's size, the type of data being transmitted, the level of redundancy required, the security of the network, and the physical environment in which the network will be deployed. By carefully considering these factors, network administrators can choose the most appropriate topology for their network, and ensure that it is capable of meeting the network's current and future needs.

Let's dive a bit deeper into the various network topologies and their advantages and disadvantages.

1. **Bus Topology**

In a bus topology, all devices on the network are connected to a single communication channel called the bus. Data is transmitted along the bus and received by all devices on the network, which can then filter out the data that is intended for them.

Advantages:

a) Easy to set up and maintain

b) Low cost

c) Efficient for small networks

Disadvantages:

a) Limited scalability

b) Susceptible to network congestion and collisions

c) Security vulnerabilities due to the fact that any device can "listen in" on data transmissions

2. **Star Topology**

In a star topology, all devices on the network are connected to a central hub or switch. Data is transmitted between devices via the hub, which acts as a mediator.

Advantages:

a) Easy to set up and maintain
b) Easy to add or remove devices without affecting the rest of the network
c) High level of security due to the centralized nature of the topology

Disadvantages:

a) Higher cost than bus topology
b) Hub represents a single point of failure
c) Limited scalability due to the fact that all devices must be connected to the hub

3. Ring Topology

In a ring topology, devices on the network are connected in a closed loop. Data is transmitted between devices in one direction around the ring.

Advantages:

a) Easy to set up and maintain
b) Efficient use of network resources due to the lack of collisions
c) Low cost

Disadvantages:

a) Single point of failure if any device on the ring fails
b) Limited scalability
c) Limited data transfer speed due to the nature of the ring

4. Mesh Topology

In a mesh topology, devices on the network are connected to multiple other devices, creating multiple paths for data transmission. Data is transmitted between devices via the shortest path.

Advantages:

a) High level of redundancy, as data can be transmitted via multiple paths
b) Highly scalable due to the decentralized nature of the topology
c) Can handle high volumes of data traffic

Disadvantages:

a) High cost due to the number of connections required
b) Difficult to set up and maintain
c) Difficult to diagnose and repair problems due to the complex nature of the topology

5. Tree Topology

In a tree topology, devices on the network are organized in a hierarchical structure, with parent and child nodes. Data is transmitted between nodes via the parent-child relationship.

Advantages:

a) Highly scalable due to the hierarchical nature of the topology

b) Can be used to create subnetworks for different departments or functions

c) Redundancy can be achieved by creating multiple paths between nodes

Disadvantages:

a) Higher cost than bus or star topology

b) Root node represents a single point of failure

c) Performance can be impacted if too many nodes are added to the tree

6. Hybrid Topology

A hybrid topology combines two or more of the above topologies to create a more flexible and resilient network. For example, a hybrid topology might combine a star topology with a mesh topology to create a network that is both highly scalable and highly redundant.

Advantages:

a) Offers the advantages of multiple topologies

b) Can be customized to meet specific network requirements

c) Flexible and adaptable to changing network needs

Disadvantages:

a) Can be more complex to set up and maintain than a single topology

b) Higher cost than a single topology

In summary, each network topology has its own set of advantages and disadvantages. The choice of topology will depend on factors such as the size and complexity of the network, the type of data being transmitted, the level of redundancy required, the security of the network, and the physical environment in which the network will be deployed. By carefully considering these here are some additional details about network topologies:

7. Point-to-Point Topology

In a point-to-point topology, two devices are connected directly to each other, with no intermediary devices. This type of topology is often used for simple networks or for connecting devices in a local area network (LAN).

Advantages:

a) Simple and easy to set up
b) Low cost
c) Efficient for small networks

Disadvantages:

a) Limited scalability
b) No redundancy
c) Not suitable for complex networks

8. Fully Connected Topology

In a fully connected topology, every device is connected to every other device on the network. This type of topology is highly redundant, as data can be transmitted along multiple paths. However, it is also very expensive and complex to set up.

Advantages:

a) High level of redundancy
b) Can handle high volumes of data traffic
c) No single point of failure

Disadvantages:

a) Very expensive
b) Very complex to set up and maintain
c) Not suitable for small or medium-sized networks

9. Partially Connected Topology

A partially connected topology is a hybrid topology that combines elements of a fully connected topology with other topologies, such as a star or ring topology. In a partially connected topology, some devices are connected to each other, while others are connected to a central hub or switch[7], [8].

Advantages:

a) Offers a balance between redundancy and cost
b) Can be customized to meet specific network requirements
c) Can handle high volumes of data traffic

Disadvantages:

a) More complex to set up and maintain than a single topology

b) Higher cost than a single topology

c) Limited scalability

10. Centralized Topology

In a centralized topology, all data and network resources are controlled by a single server or central hub. This type of topology is often used in enterprise-level networks, where a large number of devices must be managed and monitored.

Advantages:

a) High level of control and security

b) Easy to manage and monitor network resources

c) Efficient use of network resources

Disadvantages:

a) Single point of failure

b) Limited scalability

c) Higher cost than other topologies

11. Decentralized Topology

In a decentralized topology, network resources and data are distributed across multiple servers or nodes. This type of topology is often used in peer-to-peer networks, where all devices on the network have equal access to resources and can communicate directly with each other.

Advantages:

a) Highly scalable

b) No single point of failure

c) Can handle high volumes of data traffic

Disadvantages:

a) More difficult to manage and monitor network resources

b) Can be less secure than a centralized topology

c) May require more advanced technical expertise to set up and maintain

In conclusion, choosing the right network topology depends on a variety of factors, including the size and complexity of the network, the type of data being transmitted, the level of redundancy required, and the physical environment in which the network will be deployed. By carefully considering these factors, network administrators can choose a topology that is efficient, secure, and cost-effective for their particular needs[9], [10].

Sure, here are some more details about network topologies:

12. Mesh Topology

In a mesh topology, every device on the network is connected to every other device in a point-to-point arrangement. This provides a high level of redundancy, as data can be transmitted along multiple paths. However, it can be expensive and complex to set up and maintain, particularly for large networks.

Advantages:

a) High level of redundancy
b) Can handle high volumes of data traffic
c) No single point of failure

Disadvantages:

a) Very expensive and complex to set up and maintain
b) Not suitable for small or medium-sized networks
c) Higher power consumption due to multiple connections

13. Hybrid Topology

A hybrid topology is a combination of two or more topologies, such as a star-ring or a bus-star topology. This allows network administrators to customize the topology to meet the specific requirements of their network.

Advantages:

a) Offers a balance between redundancy and cost
b) Can be customized to meet specific network requirements
c) Can handle high volumes of data traffic

Disadvantages:

a) More complex to set up and maintain than a single topology
b) Higher cost than a single topology
c) Limited scalability

14. Ring Topology

In a ring topology, all devices on the network are connected in a circular arrangement, with each device connected to its two nearest neighbors. Data is transmitted around the ring in one direction, with each device acting as a repeater to amplify the signal.

Advantages:

a) Simple and easy to set up
b) Low cost
c) Efficient for small networks

Disadvantages:

a) Limited scalability

b) No redundancy

c) Not suitable for complex networks

15. Tree Topology

In a tree topology, devices are arranged in a hierarchical structure, with a main trunk or backbone connecting multiple branches. This type of topology is often used in large enterprise networks, as it provides a high level of scalability and redundancy.

Advantages:

a) Highly scalable

b) Provides redundancy through multiple branches

c) Can handle high volumes of data traffic

Disadvantages:

a) Complex to set up and maintain

b) Requires a significant amount of cabling

c) Single point of failure in the main trunk

REFERENCES:

[1] X. Diego, L. Marcon, P. Müller, and J. Sharpe, "Key Features of Turing Systems are Determined Purely by Network Topology," *Phys. Rev. X*, 2018, doi: 10.1103/PhysRevX.8.021071.

[2] G. Engels *et al.*, "Clinical pain and functional network topology in Parkinson's disease: a resting-state fMRI study," *J. Neural Transm.*, 2018, doi: 10.1007/s00702-018-1916-y.

[3] Y. Li, A. Hao, X. Zhang, and X. Xiong, "Network topology and systemic risk in Peer-to-Peer lending market," *Phys. A Stat. Mech. its Appl.*, 2018, doi: 10.1016/j.physa.2018.05.083.

[4] S. J. Pappu, N. Bhatt, R. Pasumarthy, and A. Rajeswaran, "Identifying Topology of Low Voltage Distribution Networks Based on Smart Meter Data," *IEEE Trans. Smart Grid*, 2018, doi: 10.1109/TSG.2017.2680542.

[5] A. Ledwoch, H. Yasarcan, and A. Brintrup, "The moderating impact of supply network topology on the effectiveness of risk management," *Int. J. Prod. Econ.*, 2018, doi: 10.1016/j.ijpe.2017.12.013.

[6] A. Nedic, A. Olshevsky, and M. G. Rabbat, "Network Topology and Communication-Computation Tradeoffs in Decentralized Optimization," *Proceedings of the IEEE*. 2018. doi: 10.1109/JPROC.2018.2817461.

[7] V. P. Pastore, P. Massobrio, A. Godjoski, and S. Martinoia, "Identification of excitatory-inhibitory links and network topology in large-scale neuronal assemblies from multi-electrode recordings," *PLoS Comput. Biol.*, 2018, doi: 10.1371/journal.pcbi.1006381.

[8] N. Qaqos, S. R. M. Zeebaree, and B. Hussan, "Opnet Based Performance Analysis and Comparison Among Different Physical Network Topologies," *Acad. J. Nawroz Univ.*, 2018, doi: 10.25007/ajnu.v7n3a199.

[9] Y. Li, K. Xie, L. Wang, and Y. Xiang, "The impact of PHEVs charging and network topology optimization on bulk power system reliability," *Electr. Power Syst. Res.*, 2018, doi: 10.1016/j.epsr.2018.06.002.

[10] P. E. Strandberg, T. J. Ostrand, E. J. Weyuker, D. Sundmark, and W. Afzal, "Automated test mapping and coverage for network topologies," in *ISSTA 2018 - Proceedings of the 27th ACM SIGSOFT International Symposium on Software Testing and Analysis*, 2018. doi: 10.1145/3213846.3213859.

CHAPTER 6

SYMMETRIC CYPHER MODEL

Dr. C. Kalairasan, Professor and Associate Dean
Department of Computer Science and Engineering, Presidency University, Bangalore, India
Email Id- kalairasan@presidencyuniversity.in

Symmetric Cipher Model is a cryptographic technique used to secure data transmission between two parties. It is also known as Shared Secret Cryptography or Secret Key Cryptography. The core idea behind Symmetric Cipher Model is to use a single secret key to both encrypt and decrypt the message. In other words, the same key is used for both encryption and decryption, and the key must be kept secret from any third party.

This model is widely used in various communication systems, such as electronic payment systems, online banking, messaging apps, and many more. This model is an essential aspect of modern-day communication systems and is used to ensure the confidentiality, integrity, and authenticity of the data transmitted.

The Symmetric Cipher Model consists of various components such as plaintext, encryption algorithm, secret key, cipher text, decryption algorithm, and key management system. In this article, we will discuss these components in detail.

Components of Symmetric Cipher Model

1. **Plaintext**

The plaintext is the original message that is to be transmitted between the two parties. It can be in any form, such as a text message, audio, video, or any other digital data. The plaintext is converted into cipher text through encryption algorithms, and the recipient receives the ciphertext, which is then decrypted using decryption algorithms.

2. **Encryption Algorithm**

The encryption algorithm is a set of mathematical operations that are applied to the plaintext to transform it into cipher text. The encryption algorithm is designed to be reversible so that the cipher text can be converted back into the plaintext using the decryption algorithm. There are many encryption algorithms such as Advanced Encryption Standard (AES), Data Encryption Standard (DES), and Rivest-Shamir-Adleman (RSA).

3. **Secret Key**

The secret key is the key used to encrypt and decrypt the message. The same key is used for both encryption and decryption. The security of the Symmetric Cipher Model depends on the security of the secret key. If the secret key falls into the hands of an unauthorized person, they can easily decrypt the cipher text and access the plaintext. Therefore, it is crucial to keep the secret key secret and secure.

4. **Cipher text**

The cipher text is the encrypted message that is sent to the recipient. It is produced by applying the encryption algorithm to the plaintext using the secret key. The cipher text is meaningless and unreadable without the decryption algorithm and the secret key[1], [2].

5. **Decryption Algorithm**

The decryption algorithm is the reverse of the encryption algorithm. It takes the cipher text and the secret key as input and produces the original plaintext. The decryption algorithm should be designed in such a way that only the intended recipient can decrypt the cipher text.

6. **Key Management System**

The key management system is responsible for the generation, distribution, and storage of secret keys. It is a critical component of the Symmetric Cipher Model, as the security of the system depends on the security of the secret key. The key management system ensures that the secret key is kept secret and secure and is only accessible to the authorized parties. The key management system also ensures that the keys are periodically changed to maintain the security of the system.

Types of Symmetric Cipher Model

There are two types of Symmetric Cipher Model - Stream Cipher and Block Cipher.

1. **Stream Cipher**

A Stream Cipher encrypts one bit or one byte at a time. It generates a stream of random bits that are combined with the plaintext to produce the ciphertext. The key stream is generated by a pseudo-random number generator (PRNG). The key stream is combined with the plaintext bit by bit using a bitwise XOR operation to produce the ciphertext. The decryption process is the reverse of the encryption process, and the key stream is generated using the same PRNG.

Stream Cipher is efficient and can be used for real-time data transmission. It is often used in wireless communication systems and satellite communication systems.

2. **Block Cipher**

A Block Cipher encrypts a block of fixed size, typically 64 or 128 bits. The plaintext is divided into blocks, and each block is encrypted separately using the same secret key. The encryption algorithm performs a series of mathematical operations on the plaintext block, using the secret key to produce the cipher text block. The decryption algorithm performs the reverse of the encryption algorithm, using the same secret key to convert the cipher text block back to the plaintext block.

Block Cipher is more secure than Stream Cipher, but it is slower and requires more processing power. It is often used in applications where security is the primary concern, such as online banking and secure communication systems.

The Symmetric Cipher Model is used in various applications, such as secure messaging apps, online banking, and e-commerce websites. Let's take the example of secure messaging apps to understand how the Symmetric Cipher Model works in real-world applications.

When a user sends a message through a secure messaging app, the message is first encrypted using the Symmetric Cipher Model. The plaintext message is encrypted using the encryption algorithm and the secret key to produce the cipher text message. The cipher text message is then sent to the recipient[3], [4].

When the recipient receives the cipher text message, the message is decrypted using the decryption algorithm and the same secret key. The cipher text message is converted back to the original plaintext message, which can then be read by the recipient.

The key management system ensures that the secret key used for encryption and decryption is kept secret and secure. The key is generated by the app and distributed to the sender and the recipient. The key is periodically changed to maintain the security of the system.

Challenges of Symmetric Cipher Model

The Symmetric Cipher Model has several challenges that need to be addressed to ensure the security of the system.

1. **Key Management**

The key management system is a critical component of the Symmetric Cipher Model. It is responsible for the generation, distribution, and storage of secret keys. If the secret key falls into the hands of an unauthorized person, they can easily decrypt the cipher text and access the plaintext. Therefore, it is crucial to keep the secret key secret and secure. The key management system ensures that the keys are generated securely and distributed only to the authorized parties.

2. **Key Distribution**

The distribution of the secret key is a significant challenge in the Symmetric Cipher Model. The key must be distributed securely to the sender and the recipient. If an unauthorized person intercepts the key during transmission, they can easily access the plaintext. Therefore, it is essential to ensure that the key is distributed securely and only to the authorized parties.

3. **Key Exchange**

The key exchange is a process of securely exchanging the secret key between the sender and the recipient. The key exchange process is vulnerable to attacks, such as the Man-in-the-Middle (MitM) attack, where an attacker intercepts the key exchange and replaces the key with their own key. To prevent such attacks, various protocols have been developed, such as the Diffie-Hellman key exchange protocol, which enables two parties to exchange keys securely over an insecure channel.

4. Key Length

The security of the Symmetric Cipher Model depends on the length of the secret key. The longer the key, the more secure the system. However, longer keys require more processing power, and the encryption and decryption process becomes slower. Therefore, there is a trade-off between security and speed, and the key length should be chosen carefully to maintain a balance between the two.

The Symmetric Cipher Model is a widely used cryptographic technique that is used to secure data transmission between two parties. It is used in various applications, such as online banking, secure messaging apps, and e-commerce websites, to ensure the confidentiality [5], [6].

Secret key: The encryption algorithm also requires the secret key. The value of the key is unrelated to either the plaintext or the algorithm. The result of the algorithm will vary based on the particular key being used at the moment. The key determines exactly what replacements and modifications the algorithm does.

You should be able to: Present an overview of the key ideas in symmetric cryptography after reading this chapter. Describe the differences between a brute-force assault and cryptanalysis. Recognize how a polyalphabetic substitution cipher works. Recognize how a monoalphabetic substitution cipher works.

The output message that has been scrambled. The plaintext and the secret key both play a role. Two separate keys will result in two distinct cipher texts for the same message. The cipher text is an incomprehensible stream of data that seems random. The decryption algorithm basically just reverses the encryption method. It generates the original plaintext using the secret key and the cipher text.

For safe usage of traditional encryption, the following two conditions must be met:

A reliable encryption algorithm is required. We would want the method to at least prevent an adversary who is aware of the algorithm and has access to one or more ciphertexts from deciphering the ciphertext or determining the key. The opponent should not be able to decipher the ciphertext or find the key, even if they have access to several ciphertexts and the plaintext that was used to create each ciphertext. This criterion is often expressed in a stronger manner.

The secret key must have been acquired by the sender and recipient in a safe manner, and both parties are required to keep the key safe. All communication using this key is readable if someone can find the key and understands the algorithm. On the basis of the ciphertext and the encryption/decryption technique information, we presum that it is impossible to decode a communication. In other words, the only thing we need to keep a secret is the key, not the algorithm.

Manufacturers can and have created low-cost chip implementations of data encryption methods since the algorithm need not be kept a secret. These chips are readily accessible and used in many different goods. The biggest security issue with symmetric encryption is maintaining the

key's confidentiality. A source generates a plaintext message, X = [X1, X2, c, XM]. The letters in a finite alphabet make up the M components of X.

Typically, the binary numbers 0 and 1 are utilized. A key using the formula K = [K1, K2, c, KJ] is produced for encryption. If the key is produced at the message source, it must also be sent through a secure channel to the destination.

The key might also be created and securely sent to both the source and the destination by a third party. The cipher text Y = [Y1, Y2, c, YN] is created by the encryption method using the message X and the encryption key K as input. This may be expressed as Y = E. (K, X) According to this notation, Y is generated by the encryption method E as a function of the plaintext X, with the precise function being defined by the key K's value.

An adversary who is monitoring Y but does not have access to K or X may try to retrieve K, X, or both. The opponent is deemed to be familiar with both the encryption (E) and decryption (D) algorithms. The goal of the attempt is to recover X by producing a plaintext estimate Xn if the adversary is solely interested in this specific message. However, it often happens that the adversary is also interested in being able to read future messages, in which case an effort is made to recover K by creating an estimate Kn.

Three separate characteristics of cryptographic systems are as follows:

1. The procedures utilized to convert plaintext into ciphertext. Each bit, letter, or group of bits or letters in the plaintext is translated into another element in a substitution method, and each encryption technique is based on the transposition principle, which rearranges the components in the plaintext. The primary need is to ensure that no information is lost (i.e., that all operations are reversible). Many phases of replacements and transpositions are involved in the majority of systems, also known as product systems.
2. The quantity of keys used. The system is referred to as symmetric, single-key, secret-key, or conventional encryption if the sender and receiver share the same key. The system is known as asymmetric, two-key, or public-key encryption if the sender and recipient employ distinct keys.
3. The mechanism in which the plaintext is handled. A block cipher creates an output block for every input block it analyzes, one block of elements at a time. A stream cipher continually processes the input elements, generating the output one element at a time.

Attacks using brute force and cryptanalysis

Attacking an encryption system often aims to recover the key being used rather than just obtaining the plaintext of a particular cipher text. There are two main methods for taking down a typical encryption system:

a) **Cryptanalysis:** Cryptanalytic assaults depend on the algorithm's design as well as maybe some general knowledge of the plaintext's properties or even certain examples of plaintext-cipher text pairings. This kind of attack uses the algorithm's features to try to figure out the key being used or a particular plaintext.
b) **Brute-force attack:** The attacker attempts each potential key on a piece of cipher text until a plaintext translation that can be understood is achieved. To succeed, on average, 50% of all potential keys must be tested. The outcome is disastrous if either sort of attack is successful in reducing the key: All communications that have been encrypted with that key in the past and present are now vulnerable.

Based on the quantity of information the cryptanalyst is aware oflists the many sorts of cryptanalytic assaults. When just the cipher text is accessible, the most challenging issue arises. In rare instances, not even the encryption algorithm is known, but generally speaking, we may assume that the adversary is aware of the encryption method. In these conditions, a brute-force assault that tries every key is a possibility. This becomes unworkable if the key space is extremely big. As a result, the opponent is forced to depend on an examination of the cipher text itself, usually via the use of different statistical tests. The opponent must have a broad understanding of the kind of plaintext that is disguised, such as English or French text, an EXE file.

Since the opponent has the least amount of information to deal with, the ciphertext-only attack is the simplest to counter. But often, the analyst is privy to additional details. One or more plaintext communications as well as their encryptions may be captured by the analyzer. Alternatively, the analyst could be aware that a communication would include certain plaintext patterns. For instance, a file that is encoded in the Postscript format always starts with the same pattern, and electronic funds transfer messages may include a standard header or banner, among other things. These are all examples of well-known plaintext. With this information, the analyst may be able to determine the key based on how the known plaintext is altered.

What may be referred to as a probable-word assault is closely similar to the known-plaintext attack. The adversary may not be familiar with the contents of the communication if they are dealing with the encryption of a generic prose message. The message, however, could be known in part if the adversary is searching for a particularly specific piece of information. For instance, if a complete accounting file is being communicated, the adversary may be aware of the location of certain key phrases in the file's header. Another example would be the presence of a copyright declaration in a predetermined point in the source code of a software created by Corporation X. A chosen-plaintext attack is feasible if the analyst can persuade the source system to include a message of their choosing in the system.

There are two additional definitions worth mentioning. No matter how much ciphertext is provided, an encryption technique is considered unconditionally secure if the ciphertext it produces cannot be uniquely decoded to reveal the matching plaintext. This means that, regardless of how much time an adversary has, it is simply not feasible for him or her to decode the ciphertext since the necessary information is missing. There is no encryption technique that is 100% secure, with the exception of the one-time pad method covered later in this chapter. Users

of encryption algorithms may only aim for an algorithm that satisfies one or both of the following requirements. The value of the encrypted data is less than the cost of cracking the encryption. The amount of time needed to decipher the encryption is longer than the information's useful lifespan. If any of the aforementioned two requirements is satisfied, an encryption method is considered to be computationally secure. Unfortunately, it is quite difficult to gauge the amount of work necessary to correctly cryptanalyze cipher text[7], [8].

All types of cryptanalysis for symmetric encryption techniques are designed to take advantage of the possibility that certain elements of the plaintext's structure or pattern may remain visible after encryption and be recognizable in the cipher text. As we look at several symmetric encryption algorithms in this chapter, it will become evident why. As we will see in Part Two, cryptanalysis for public-key schemes starts from a fundamentally different assumption, namely, that it could be able to infer one of the two keys from the other due to the mathematical features of the pair of keys.

In a brute-force assault, every key is tested until the cipher text can be deciphered and converted into plaintext. To succeed, on average, 50% of all potential keys must be tested. In other words, if there are X distinct keys, an attacker would often find the real key after X/2 attempts. It's crucial to understand that a brute-force approach involves more than just trying every conceivable combination of characters. The analyst must be able to identify plaintext as plaintext unless known plaintext is given. Even though English recognition would need to be automated, the outcome is obvious if the message is merely plain text in English. Recognition is more challenging if the text message has been compressed before encryption. Additionally, the challenge is considerably harder to automate if the message includes some more generic sort of data, such a compressed numerical file. As a result, additional information about the anticipated plaintext and a way to automatically distinguish plaintext from garble are required to augment the brute-force method.

We look at a few examples of what may be referred to as classical encryption methods in this and the next section. By studying these methods, we can demonstrate the fundamental symmetric encryption algorithms now in use as well as the kinds of cryptanalytic assaults that must be prepared for. Substitution and transposition are the two fundamental building elements of all encryption methods. In the next two parts, we look at them. Finally, we talk about a system that combines transposition and substitution.

A replacement approach involves changing the plaintext's letters with alternative letters, integers, or symbols. If the plaintext is thought of as a series of bits, substitution entails swapping out the bit patterns of the plaintext with those of the cipher text. Julius Caesar made the first and most basic known use of a substitution cipher. Each letter in the alphabet is swapped out for the letter three positions below in the alphabet as part of the Caesar

The third trait is important as well. The output of plaintext could not be discernible if the language of the plaintext is unknown. Additionally, the input could be shortened or compressed in some way, which complicates identification once again. As an example, the plaintext may not be recognized if this file is subsequently encrypted using a basic substitution cipher extended to

contain more than just 26 alphabetic letters after that. The Caesar cipher has just 25 potential keys, making it far from safe. By enabling an arbitrary substitute, the key space may be dramatically increased. We first define the word "permutation" before moving on. A permutation of a finite set of items S is a sequence in which every element in S appears precisely once. There are six permutations of S, such as abc, acb, bac, bca, cab, and cba if S = a, b, and c.

A collection of n items may be combined in n! Different ways since there are n ways to choose the first element, n ways to choose the second, n ways to choose the third, and so on. There are 26! or more than 4 * 1026 potential keys if, on the other hand, the "cipher" line may be any combination of the 26 alphabetic letters. This would appear to prevent the use of brute-force cryptanalysis methods since the key space is 10 orders of magnitude larger than the key space for DES. Because just one cipher alphabet is used per message to map from the plain alphabet to the cipher alphabet, this method is known as a monoalphabetic substitution cipher.

First, the relative frequency of the letters may be calculated and contrasted with a typical English frequency distribution. This method may be adequate on its own if the message were lengthy enough, but because it's not, we can't anticipate a precise match in this case. In any case, the following table lists the relative frequency of the letters in the ciphertext (in percentages):

It seems that the plain letters e and t are equal to the cipher letters P and Z, albeit it is unclear which is which. There are many methods to go forward at this stage. To evaluate whether the plaintext resembles a plausible "skeleton" of a message, we may make some impromptu assignments and begin to fill it in. A more methodical technique is to search for more regularities. For instance, it could be known that a text contains a certain term. Alternately, we may search for recurring cipher letter sequences and attempt to determine their plaintext counterparts.

Examining the frequency of diagrams two-letter combinations is a useful technique. The relative frequency of diagrams might. The most frequent of these diagrams. The most frequent diagram in our cipher text is ZW, which occurs three times. Thus, we establish a correlation between Z and t and W and h. Then, based on our prior supposition, we may compare P to e. Now take note of the sequence ZWP, which may be translated as "the" in the ciphertext. This trigram, which consists of three letters, occurs the most often in the English language, which would appear to point in the proper direction. Next, pay attention to the ZWSZ sequence in the first line. We do not know whether these four letters make up a whole word, but if they do, the word would have the form. Even though just four letters have been found, we now know a significant portion of the message. From here, a solution should be readily found via more frequency analysis and trial and error. The whole plaintext, with words separated by spaces, is as follows:

Due to the fact that they mirror the frequency information of the original alphabet, monoalphabetic ciphers are simple to crack. Homophones are many alternatives for a single letter that may be used as a defense. For instance, the homophones 16, 74, 35, and 21 might all be assigned to the letter e in a random or rotating manner, along with other cipher symbols. Single-letter frequency information is totally erased if the amount of symbols allocated to each letter is proportionate to its relative frequency. Using homophones, the famous mathematician

Carl Friedrich Gauss thought he had created an impenetrable encryption. However, even with homophones, each plaintext element only influences one ciphertext element, therefore multiple-letter patterns are unaffected.

To reduce how much of the plaintext's structure is preserved in the ciphertext, substitution ciphers primarily use two techniques: The two methods involve using several cipher alphabets and numerous plaintext encryption keys. We quickly go through each. The Play fair cipher, which interprets diagrams in the plaintext as single units and converts these units into cipher text diagrams, is the most well-known multiple-letter encryption cipher.

The Play fair method is based on the construction of a 5 by 5 letter matrix from a keyword. Here is an illustration from Dorothy Sayers' novel Have His Case that Lord Peter Wimsey.

The term to remember here is monarchy. The matrix is created by first filling in the letters of the keyword (after removing any duplicates) from left to right and from top to bottom, and then finishing the matrix with the remaining letters in alphabetical order. I and J are considered to be one letter. Following these guidelines, plaintext is encrypted two letters at a time:

1. To treat balloon as ba lx lo on, repeating plaintext characters that are in the same pair are separated by a filler letter, such as x.
2. The letter to the right of each pair of adjacent plaintext letters that are in the same row of the matrix is substituted, with the row's initial element following the final one in a circle. Ar, for instance, is encoded as RM.
3. The letter below a pair of plaintext letters in the same column replaces each of them, with the top element of the column then circularly following the last.

For instance, mu is encoded as CM. If not, the letter from its own row and the column that the other plaintext letter occupies is substituted for each pair of plaintext letters. As a result, becomes IM and has becomes BP (or JM, as the enciphered wishes). The Play fair cipher is a significant improvement over straightforward monoalphabetic ciphers.

Although Baron Playfair of St. Andrews, a friend of Sir Charles Wheatstone's, championed the encryption at the British Foreign Office, it was really Sir Charles Wheatstone who developed the cipher in 1854 such that it is more challenging to identify certain digrams. Additionally, compared to digrams, individual letter relative frequencies show a significantly wider range, making frequency analysis much more challenging. These factors contributed to the Playfair cipher's long-standing reputation of being impenetrable. The British Army employed it as their standard field system during World War I, and the U.S. Army and other Allied forces continued to use it extensively throughout World War II.The Play fair cipher is quite simple to crack despite the high degree of trust in its security since it mostly maintains the plaintext language's structural integrity. In most cases, a few hundred letters of cipher text are enough. The plaintext line displays a typical frequency distribution of the 26 alphabetic characters in regular text without considering capital or lowercase letters. Because the frequency values for individual letters are the same but alternative letters have been used in place of the original letters, this is also the frequency distribution of any monoalphabetic substitution cipher.

The number of instances of each letter in the text is tallied, then divided by the number of instances of the letter that appears the most often. We can observe from Figure 3.5's findings that the most common letter is e. Thus, e has a relative frequency of 1, t has a relative frequency of 9.056/12.702 0.72, and so on. In decreasing order of frequency, the dots on the horizontal axis correspond to the letters.

The frequency distribution that is produced when the text is encrypted using the Playfair encryption is also shown in Figure 3.6. The number of instances of each letter in the cipher text was once again divided by the number of occurrences of e in the plaintext to normalize the figure. The amount to which the frequency distribution of letters, which makes substitution easy to solve, is shown in the final graphic.

The ciphertext plot of frequencies would be flat if the frequency distribution information were entirely hidden during the encryption process, making cryptanalysis using ciphertext alone almost impossible. The Playfair cipher, as seen in the picture, has a flatter distribution than plaintext but still discloses enough structure for cryptanalysts to work with. The Vigenère cipher, which is covered later, is also shown in the story. Based on the findings given in [SIMM93], the Hill and Vigenère curves are shown on the plot.

The Hill encryption, created by mathematician Lester Hill in 1929, is another fascinating multiletter cipher a few linear algebraic terms before discussing the Hill cipher. We are talking about matrix arithmetic modulo 26 in this topic. See Appendix E if you need a review on matrix multiplication and inversion.

The equation M(M-1) = M-1 M = I, where I is the identity matrix, is used to determine the inverse M-1 of a square matrix M. I is a square matrix with zeros on all sides except the main diagonal, which runs from top left to lower right. Although the inverse of a matrix is not always possible, when it is, the previous equation is satisfied. For instance, A = 5 8 17 3 A-1 mod 26 = 9 2 1 15 AA-1 = (5 * 9) + (8 * 1) (8 * 15) (17 * 9) + (3 * 1) (17 * 2) + (3 * 15) = 53 130 156 79 mod 26 = 1 0 0 1

We start with the idea of determinant to describe how the inverse of a matrix is calculated. The sum of all the products that can be created by taking precisely one element from each row and one element from each column, with some of the product terms preceded by a negative sign, is the determinant for any square matrix (m × m). The determinant for a 2 * 2 matrix is equal to k11k22 - k12k21. The value of the determinant for a 3 * 3 matrix is given by k11k22k33 + k21k32k13 + k31k12k23 - k31k22k13 - k21k12k33 - k11k32k23. Although this cipher is a little more challenging to comprehend than the others in this chapter, it highlights a crucial aspect of cryptanalysis that will be helpful in a later chapter. On a first reading, this paragraph may be disregarded. If matrix A has a nonzero determinant, then is computed to represent the matrix's inverse ij = (det A)-1 (-1)i+j (Dji), where (Dji) is a subdeterminant created by removing the jth row and ith column from A, det(A) is the determinant of A, and (det A)-1 is the multiplicative inverse of (det A) mod 26. Using our previous example as a guide, det 5 8 17 3 = (5 * 3) - (8 * 17) = -121 mod 26 = 9

Because 9 * 3 = 27 mod 26 = 1, we can demonstrate that 9-1 mod 26 = 3 (see Chapter 2 or Appendix E). As a result, we calculate A's inverse as A = 5 8 17 3 A-1 mod 26 = 3 3 -8 -17 5 = 3 3 18 9 = 9 54 27 15 = 9 2 1 15. ALGORITHM HILL This encryption technique replaces m consecutive plaintext letters with m ciphertext letters in place of the plaintext letters. Each letter is given a numerical value (a = 0, b = 1, c = 25) in m linear equations that decide the replacement. The equations for the system are c1 = (k11p1 + k21p2 + k31p3) mod 26 and c2 = (k12p1 + k22p2 + k32p3) mod 26 for m = 3. (K13P1, K23P2, and K33P3) = c3 mod 26 Row vectors and matrices may be used to represent this: 6 (c1 c2 c3) = (p1 p2 p3)£ K11, K12, K13, K21, K22, K23, K31, K32, and K33 C = PK mod 26 or mod 26 where K is a 3 * 3 matrix representing the encryption key and C and P are row vectors of length 3 representing the plaintext and ciphertext, respectively. Operations are carried out mod. 26.

In some publications on cryptography, the plaintext and ciphertext are expressed as column vectors instead of row vectors, which are then inserted after the matrix. We follow Sage's tradition of using row vectors. Consider the plaintext "paymoremoney" as an example, then use the encryption key.

K = £ \s17 17 5 \s21 18 21 \s2 2 19≥

The vector (15 0 24) represents the first three letters of the plaintext, and (15 0 24)K = (303 303 531) mod 26 = (17 17 11) = RRL. Following this pattern, the final ciphertext is RRLMWBKASPDH, which covers the full plaintext.

Utilizing the inverse of the matrix K is necessary for decryption. Det K may be calculated to be 23; hence, (det K)-1 mod 26 is equal to 17. The inverse may then be calculated as follows: 7 K-1 = $ 4 9 15 15 17 6 24 0 17 This is shown as follows: £ 17 17 5 21 18 21 2 2 19 £ 4 9 15 15 17 6 24 0 17 = € 443 442 442 858 495 780 494 52 365 mod 26 = £ 100 010 001

It is clear that the plaintext may be retrieved if the matrix K-1 is applied to the ciphertext.The Hill system may be written down as follows: C = E(K, P) = PK mod 26 P = D(K, C) = CK-1 mod 26 = PKK-1 = P.

The Hill cipher's strength, like with Playfair's, is that it totally conceals single-letter frequencies. In fact, Hill conceals more frequency information when using a bigger matrix. Therefore, a 3 × 3 Hill cipher conceals frequency information for both single- and double-letter words.

The Hill cipher may be quickly cracked using a known plaintext assault, while being robust against ciphertext-only attacks. Let's say we have m plaintext-ciphertext pairings, each of length m, for a m * m Hill cipher. We provide the pairings Cj = Pj K for 1... j... m and for an unidentified key matrix K by labeling them as Pj = (c1j c1jc cmj) and Pj = (c1j c1jc pmj). Next, establish the two m*m matrices X = (pij) and Y = (cij). The matrix formula Y = XK may then be created. If X has an inverse, then K = X-1 Y may be found. In the event that X isn't invertible, further iterations of X may be created using more plaintext-ciphertext combinations up until an invertible X is attained.

Think about this instance. Assume that the ciphertext HCRZSSXNSP is produced after the plaintext "hillcipher" has been encrypted using a 2 * 2 Hill cipher. As a result, we are aware that

(7 8)K mod 26 = (7 2); (11)K mod 26 = (17 25); and so on. We have 7 using the first two plaintext-ciphertext pairings.

Multiple-letter ciphers utilizing multiple monoalphabetic replacements as one reads through the plaintext message is another option to improve on the straightforward monoalphabetic strategy.

This method is also known as a polyalphabetic substitution cipher. These characteristics are shared by all of these methods:

1. There is a set of connected monoalphabetic substitution rules.
2. The specific rule that is used for a given transformation is determined by a key.

The Vigenère cipher is one of the most popular and straightforward polyalphabetic ciphers. The 26 Caesar ciphers with shifts from 0 to 25 make up the associated monoalphabetic substitution rules in this system. A key letter, which is the ciphertext letter that stands in for the plaintext letter a, is used to identify each cipher. Thus, the key value 3.8 designates a Caesar cipher with a shift of 3.

The Vigenère tableau is a matrix that is often used to facilitate both understanding and application of this system. Box.com/Crypto7e has a paper that discusses this tableau in detail. Until the whole plaintext sequence has been encrypted. The encryption process may be described by the generic equation Ci = (pi + ki mod m) mod 26. This may be compared to the Caesar ciphers in essence, the associated key character determines which Caesar cipher is used to encrypt each plaintext character. Decryption is an extension of in a similar way.

$$pi = \text{mod } 26\,(Ci - ki \text{ mod } m) \quad (3.4)$$

A key that is the same length as the message is required to encrypt it. The advantage of this encryption is that it has several ciphertext letters one for each distinct letter of the keyword—for each plaintext letter. The information about letter frequency is thus hidden. Not all of the plaintext structure's information is lost, however. The frequency distribution for a Vigenère cipher using a 9-character keyword. The Playfair cipher is improved, but significant frequency information is left behind.

Sketching a strategy to crack this encryption is educational since it demonstrates some of the mathematical ideas that underlie cryptanalysis. Let's start by supposing that the adversary thinks the Vigenère cipher or monoalphabetic substitution was used to encrypt the ciphertext. A quick test may be performed to determine the situation. The statistical characteristics of the ciphertext should match those of the language of the plaintext if a monoalphabetic substitution is utilized. There should be one cipher letter with a relative frequency of occurrence of approximately 12.7%, one with a relative frequency of occurrence of about 9.06%, and so on. We wouldn't anticipate a precise match between this tiny sample and the statistical profile of the plaintext language if there was just one message available for examination. Though we may presume a monoalphabetic replacement if the relationship is near.

If, on the other hand, a Vigenère cipher is suspected, then progress depends on \sdetermining the length of the keyword, as will be seen in a moment. Let's focus on determining the keyword

length for the time being. The following is a crucial realization that results in a solution: Two identical plaintext letter sequences will produce identical ciphertext sequences if they are located at a distance that is an integer multiple of the keyword length. Two occurrences of the word sequence "red" are separated by nine character places in the example above. As a result, in both instances, r, e, and d are encrypted using the key letters p, t, and e, respectively. So the ciphertext sequence is VTW in both instances[9], [10].

In the example above, we highlight the relevant ciphertext letters and shade the pertinent ciphertext numbers to demonstrate this. The repeated sequences VTW at a displacement of 9 would be recognized by an analyst who was simply looking at the ciphertext, leading them to infer that the keyword is either three or nine letters long. It might be accidental for VTW to occur twice, and it might not represent identical plaintext characters encrypted with similar key letters.

There will, however, be a lot of these repeated ciphertext sequences if the message is lengthy enough. The analyst should be able to determine the keyword length rather accurately by searching for common elements in the displacements of the different sequences. A crucial realization is now necessary for the cipher's solution. The encryption really consists of m monoalphabetic substitution ciphers if the keyword length is m. For instance, the letters in places 1, 10, 19, and so on are all encrypted using the same monoalphabetic cipher for the term DECEPTIVE. As a result, we can attack each of the monoalphabetic ciphers independently using the plaintext language's known frequency characteristics.

Using a nonrepeating term that is as lengthy as the message itself will get rid of the keyword's periodic nature. A running key is created by concatenating a keyword with the plaintext itself in Vigenère's so-called "autokey" mechanism. Even this plan can be broken via cryptanalysis. A statistical method may be used since the plaintext's letter frequency distribution matches that of the key's. For instance, e can be anticipated to be enciphered by e with a frequency of (0.127)2 0.016, while t may be expected to be enciphered by t with a frequency that is almost half as high. Successful cryptanalysis may be accomplished by taking advantage of these regularities. Selecting a term that is as lengthy as the plaintext and has no statistical link to it is the best safeguard against such a cryptanalysis. In 1918, Gilbert Vernam, an AT&T engineer, developed such a system.

A 1917 edition of Scientific American referred to the Vigenère cipher as being "impossible of translation," despite the fact that the methods for cracking it are by no means difficult. When comparable assertions are made about contemporary algorithms, it's important to keep this in mind. His technology operates with bits of digital data rather than letters. The mechanism may be briefly explained.

ci = pi ki, where pi = ith plaintext binary digit.

ki is the key's ith binary digit.

ci = the ith binary ciphertext digit and equals the exclusive-or (XOR) operation.

This should be compared to the Vigenère cipher's equation. As a result, the plaintext and key are bitwise XORed to produce the ciphertext. Decryption only requires the same bitwise operation

because of the characteristics of the XOR: pi = ci ki, which contrasts. The method used to create the key is the crucial component of this methodology. Vernam suggested using a looping tape loop that ultimately repeated the key, resulting in a system that really functioned with a very lengthy yet repeating keyword. With a lengthy key, such a scheme poses daunting cryptanalytic challenges, but it may be cracked with enough ciphertext, by using known or likely plaintext sequences, or by combining the two.

The Vernam cipher was improved by Army Signal Corps officer Joseph Mauborgne to provide the highest level of security. In order to avoid having to repeat the key, Mauborgne advised employing a random key that is as lengthy as the message. The key must also be used to encrypt and decode a single message before being destroyed. A new key that has the same length as the new message is required for every new message. A one-time pad is a kind of such method that is impenetrable. It generates random results with no statistical connection to the plaintext. Considering the ciphertext

Imagine a cryptanalyst had discovered these two keys. There are two conceivable plaintexts generated. How does the cryptanalyst determine which decryption and therefore, which key is the right one? The cryptanalyst is unable to determine which of these two keys is more likely if the actual key were generated in a truly random manner. As a result, it is impossible to determine which plaintext is correct or which key is correct.

In fact, there is a key that can generate any plaintext that is the same length as the ciphertext given any plaintext. As a result, if you conducted a thorough search using every key, you would come up with a large number of plaintexts that are readable but you would have no way of knowing which the intended plaintext was.

As a result, the code cannot be cracked. The key's randomness is solely responsible for the one-time pad's security. The ciphertext will be made up of a stream of characters that are truly random if the stream of characters that make up the key is truly random. As a result, a cryptanalyst cannot use any patterns or regularities to attack the ciphertext. Theoretically, a cipher can be found anywhere. Although the one-time pad promises absolute security, there are two major issues that arise in use:

3. There is the issue of producing numerous random keys in a practical manner. Any system that receives a lot of traffic might frequently need millions of random characters. It is a significant task to provide truly random characters in this volume.
4. The issue of key protection and distribution is even more difficult. Both the sender and the receiver must have keys that are the same length in order for a message to be sent. Consequently, there is a massive key distribution issue.

Due to these issues, the one-time pad is only marginally useful and is best used for low-bandwidth channels requiring extremely high security

REFERENCES:

[1] S. Talwar and A. Bhat, "Unravelling the Cipher of Indian Rupee's Volatility: Testing the Forecasting Efficacy of the Rolling Symmetric and Asymmetric GARCH Models," *Theor. Econ. Lett.*, 2018, doi: 10.4236/tel.2018.86079.

[2] X. Zhang and X. Wang, "Multiple-image encryption algorithm based on the 3D permutation model and chaotic system," *Symmetry (Basel).*, 2018, doi: 10.3390/sym10110660.

[3] T. A. Patil and P. D. M. K. V. Kulhalli, "Symmetric Key Cryptography Algorithm for Data Security," *Int. J. Trend Sci. Res. Dev.*, 2018, doi: 10.31142/ijtsrd9444.

[4] A. Hosoyamada and K. Yasuda, "Building quantum-one-way functions from block ciphers: Davies-Meyer and Merkle-Damgård constructions," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018. doi: 10.1007/978-3-030-03326-2_10.

[5] A. Bogdanov, E. Tischhauser, and P. S. Vejre, "Multivariate profiling of hulls for linear cryptanalysis," *IACR Trans. Symmetric Cryptol.*, 2018, doi: 10.13154/tosc.v2018.i1.101-125.

[6] P. Zhang, S. Wang, K. Guo, and J. Wang, "A secure data collection scheme based on compressive sensing in wireless sensor networks," *Ad Hoc Networks*, 2018, doi: 10.1016/j.adhoc.2017.11.011.

[7] B. Murali Krishna, H. Khan, and G. L. Madhumati, "Reconfigurable pseudo biotic key encryption mechanism for cryptography applications," *Int. J. Eng. Technol.*, 2018, doi: 10.14419/ijet.v7i1.5.9124.

[8] Y. Xing, H. Yan, and X. Lai, "New observation on division property: Simplifying models of basic operations and modeling modular multiplication operation," in *ACM International Conference Proceeding Series*, 2018. doi: 10.1145/3207677.3278003.

[9] E. Elahi, H. Raza, and S. Ali, "A new 3D playfair based secure cipher generation model," in *Proceedings - 2017 13th International Conference on Emerging Technologies, ICET2017*, 2018. doi: 10.1109/ICET.2017.8281719.

[10] S. Coretti, Y. Dodis, and S. Guo, "Non-uniform bounds in the random-permutation, ideal-cipher, and generic-group models," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018. doi: 10.1007/978-3-319-96884-1_23.

CHAPTER 7

TRADITIONAL BLOCK CIPHER STRUCTURE

Dr. T.K. Thivakaran, Professor and Head
Department of Computer Science and Engineering, Presidency University, Bangalore, India
Email Id- thivakaran@presidencyuniversity.in

Block ciphers are an important class of symmetric-key cryptographic algorithms that are widely used to provide confidentiality and integrity of data in communication and storage systems. A block cipher is a cryptographic algorithm that takes a fixed-length block of plaintext as input and produces a fixed-length block of ciphertext as output. In this article, we will discuss the traditional block cipher structure, which is the basic building block for many modern block ciphers.

The traditional block cipher structure consists of two main components: a block cipher encryption function and a key schedule. The encryption function takes a fixed-length block of plaintext and a secret key as input and produces a fixed-length block of ciphertext as output. The key schedule is used to derive a set of round keys from the secret key, which are used by the encryption function to perform a series of rounds on the plaintext. The encryption function of a traditional block cipher is usually composed of several rounds, each of which consists of three main operations: substitution, permutation, and key mixing. In the substitution operation, the plaintext block is replaced with a new block using a substitution table or S-box. The substitution operation is designed to introduce non-linearity into the encryption process and to make it difficult to reverse-engineer the encryption function.

In the permutation operation, the positions of the bits in the plaintext block are rearranged according to a fixed permutation table or P-box. The permutation operation is designed to introduce diffusion into the encryption process and to make it difficult to discern the relationship between the plaintext and ciphertext blocks. In the key mixing operation, the round key is combined with the plaintext block using a bitwise operation such as XOR. The key mixing operation is designed to introduce confusion into the encryption process and to make it difficult to deduce the round key from the ciphertext.

The key schedule of a traditional block cipher is used to derive a set of round keys from the secret key, which are used by the encryption function to perform a series of rounds on the plaintext. The key schedule typically involves a series of key transformations that are applied to the secret key to produce a set of round keys. The key transformations are designed to introduce randomness into the round keys and to make it difficult to deduce the secret key from the round keys.

There are two main types of key schedules used in traditional block ciphers: the Feistel key schedule and the SPN key schedule. The Feistel key schedule is used by block ciphers that use a

Feistel network structure, which consists of a series of identical rounds that alternate between applying the encryption function to the left and right halves of the plaintext block. The Feistel key schedule generates round keys for each round by applying a key transformation to a portion of the secret key[1], [2].

The SPN key schedule is used by block ciphers that use a substitution-permutation network structure, which consists of a series of rounds that each apply a substitution operation, a permutation operation, and a key mixing operation to the plaintext block. The SPN key schedule generates round keys for each round by applying a key transformation to the entire secret key.

The security of traditional block ciphers depends on their ability to resist a variety of attacks, including brute-force attacks, differential cryptanalysis, linear cryptanalysis, and algebraic attacks. Brute-force attacks involve trying all possible keys until the correct key is found, and the security of a block cipher is usually measured in terms of the number of key guesses required to break the cipher. Differential and linear cryptanalysis are attacks that exploit the statistical properties of the encryption function, and algebraic attacks are attacks that exploit the algebraic structure of the encryption function.

The key space of a block cipher is the total number of possible keys that can be used with the cipher. A larger key space makes it more difficult for an attacker to guess the correct key, as the number of possible keys that must be tried increases exponentially with the size of the key space. For example, a block cipher with a 128-bit key has a key space of 2^128, which is a very large number and makes brute-force attacks impractical.

Diffusion is the process by which changes in the plaintext propagate throughout the ciphertext, making it difficult to discern the relationship between the plaintext and ciphertext blocks. Confusion is the process by which the relationship between the plaintext and ciphertext blocks is made more complex and difficult to discern. By introducing both diffusion and confusion into the encryption function, traditional block ciphers are able to provide a high level of security against a variety of attacks.

Non-linearity is an important property of the substitution operation in the encryption function. Non-linearity means that the output of the S-box cannot be predicted from its input in a simple or linear way. Non-linearity is important because linear relationships can be exploited by attackers to deduce information about the plaintext or key.

One of the earliest and most widely used traditional block ciphers is the Data Encryption Standard (DES), which was developed by IBM in the 1970s and adopted by the US government as a standard in 1977. DES uses a Feistel network structure with 16 rounds, and a key size of 56 bits. While DES was widely used for many years, it was eventually replaced by the Advanced Encryption Standard (AES), which was developed by NIST in the early 2000s.AES uses a substitution-permutation network (SPN) structure and has a variable block size (128, 192, or 256 bits) and key size (128, 192, or 256 bits). AES uses 10, 12, or 14 rounds depending on the key size, and has a high degree of diffusion, confusion, and non-linearity. AES is widely regarded as one of the most secure block ciphers currently available and is used in a wide range of

applications, including wireless communication, secure messaging, and digital rights management.

Another traditional block cipher that has been widely used is Blowfish, which was developed by Bruce Schneier in 1993. Blowfish uses a Feistel network structure with a variable number of rounds and a key size of up to 448 bits. Blowfish is known for its high speed and efficiency, and has been widely used in a variety of applications, including password storage, secure communication, and software encryption[3], [4].

The security of traditional block ciphers is not only dependent on the design of the cipher itself, but also on the implementation and usage of the cipher. A poorly implemented cipher can be vulnerable to a variety of attacks, including side-channel attacks, where an attacker can gain information about the encryption process by monitoring the physical characteristics of the device performing the encryption.

To ensure the security of traditional block ciphers, it is important to use secure implementation techniques, such as constant-time algorithms and random delays to prevent timing attacks, and to use key management practices that ensure the secrecy and integrity of the key. In addition, the use of authenticated encryption algorithms, which provide both confidentiality and integrity of the data, can provide an additional layer of security against a variety of attacks.

In conclusion, the traditional block cipher structure is the basic building block for many modern block ciphers, and is a key component in the design of cryptographic systems that provide confidentiality and integrity of data. Traditional block ciphers use a combination of substitution, permutation, and key mixing operations to introduce non-linearity, diffusion, and confusion into the encryption process, and use key schedules.

The substitution cipher maps each of the 16 potential output states, each of which is represented by 4 ciphertext bits, into a distinct one of the 16 possible input states that a 4-bit input creates. It is possible to specify the encryption and decryption mappings by tabulating the results, this is the block cipher that is the most versatile and may be used to specify any reversible plaintext-to-ciphertext mapping. Feistel calls this the perfect block cipher since it permits the greatest amount of security number of potential mappings from the plaintext block to encryption.

But the perfect block cipher has a practical issue a little block the system is comparable to a conventional substitution if a size of n = 4 is employed cipher. As we have shown, such systems are susceptible to a statistical study of the plaintext. This flaw is not a natural consequence of using a substitution cipher, but rather is the outcome of using a tiny block size. If n is a big enough number and any number, When reversible plaintext and ciphertext replacement is permitted, the statistical properties of the source plaintext are sufficiently obscured that this type is impossible for cryptanalysis.

The justification is that we are free to choose any of the 2n ciphertext blocks for the initial plaintext. We choose the second plaintext from the remaining 2n - 1 ciphertext blocks, and so on. Block ciphers and the Data Encryption Standard. The ideal block cipher is an arbitrary reversible

substitution cipher for a large however, from an implementation and performance standpoint, block size angle of view The essential for such a change is the mapping itself.

For n = 4, convert plaintext to ciphertext. The entries in the table may be used to define the mapping second column, which displays the ciphertext value for each block of plaintext. In essence, this is the key that selects the particular mapping out of all potential mappings In this situation, using this simple approach to establishing the four bits times sixteen rows, or 64 bits, make up the necessary key length. Typically, for n-bit ideal block cipher, with a length of n * 2n for the key defined in this way. To avoid statistical attacks, a 64-bit block is the ideal size. However, the Key length requirements are 64 * 264 = 270 1021 bits.

Feistel notes that in light of these challenges, what is required is a constructed from components that are simple to implement, this block cipher system approximates the ideal one for high. However, before moving on to Feistel's strategy, permit me to add one more remark. The universal block substitution might be used. Cipher but limit ourselves to a subset of in order to make its implementation manageable potential mappings that may be reversed. Let's say, for instance, that we define the mapping using a series of linear equations.

Y1 equals k11x1, k12x2, k13x3, and k14x4.

y2 = K21*, K22*, K23*, and K24*

Y3 = K31*, K32*, K33*, and K34*

Y4 = K41 x 1, K42 x 2, K43 x 3, and K44 x 4

Where kij are the binary coefficients, xi are the four binary digits of the plaintext block, yi are the four binary digits of the ciphertext block, and math is mod 2. The key is just n2 in size 16 bits in this example. This kind of formulation has the risk of being susceptible to cryptanalysis by an adversary who is knowledgeable of the algorithm's structure. In this case, the Hill is effectively what we have applied to binary data rather than characters, the encryption explained in Chapter 3

Feistel suggested that by using the notion of a product cipher, which is the execution of two or more basic block ciphers, we may approach the ideal block ciphers in a certain order such that the end result is cryptographically stronger than any of the individual component ciphers. The main component of the strategy is to create a block cipher with an n-bit block length and a k-bit key length, instead of the 2n, enabling a total of 2k possible transformations using the optimal block cipher is possible.

Feistel specifically suggested using a cipher that alternates replacements and permutations, with the following definitions for these terms. Substitution unique replacement is made for each textual element or set of components by a similar element or set of elements in the ciphertext. Permutation: A permutation is the replacement of a series of plaintext elements of that order. In other words, there are no items changed in the sequence, or more specifically, the order in which the sequence's parts occur is changed. The Triple Data Encryption Algorithm (TDEA), one of the two encryption algorithms, in particular, uses the Feistel structure. AES and other encryption algorithms that have been authorized for widespread use by the National

Standards and Technology Institute (NIST). Another use of the Feistel structure is several format-preserving encryption techniques have recently emerged prominence. Additionally, the Feistel structure used in the Camellia block encryption is one TLS and a variety of other Internet security protocols have symmetric ciphers that may be used protocols. The two fundamental components of every cryptographic system are captured by Claude Shannon. Shannon was concerned about preventing statistically-based cryptanalysis. The justification is as follows. Assume that the attacker is knowledgeable of the plaintext's statistical properties. For instance, in a legible format the frequency distribution of the different letters in a message may be known. Alternatively, there can be phrases or words that are probable to occur in the message.The specific key that was used has no bearing on any of the ciphertext statistics. Such a cipher is the arbitrary substitution cipher that we previously mentioned. Confusion and misunderstanding in statistical cryptanalysis are aggravating. Diffusion transforms the plaintext's statistical structure into long-range statistics of the

Shannon's theories on secrecy measures and the security of cryptographic algorithms are expanded upon in Appendix F. To produce the ciphertext letter yn, add k consecutive letters. It is possible to demonstrate that the plaintext's statistical structure has vanished. Consequently, the letter frequency in the ciphertext will be more nearly equal than the plaintext, as will the diagram frequencies, and so on. Diffusion may be used in binary block ciphers to be accomplished by repeatedly putting the data through a permutation, then putting a function on that permutation has the result of combining bits from various locations in the original plaintext to create a single bit of ciphertext[5], [6].

In order to use a block cipher, a block of plaintext must be converted into block of ciphertext in which the key determines the transformation. The process aims to establish a statistical correlation between the plaintext and ciphertext that is as complicated as it can be to frustrate efforts to figure out the key. However, confusion aims to muddle the connection between the figures of the encryption key's value and the ciphertext as complicated as feasible, once again to stop tries to find the key. In light of this, even if the attacker manages to on the statistics of the ciphertext, the technique in which the key was utilized to construct that ciphertext is so intricate that it is challenging to figure out the key. This is done via using a sophisticated replacement method. Contrarily, a straightforward linear replacement function would just cause little confusion.Diffusion and confusion are so effective in encapsulating the core of the desirable characteristics of a block cipher, as notes, that they have turned into the standard the foundation of contemporary block cipher design. Feistel's suggested structure a plaintext is used as one of the encryption algorithms inputs block with a key K and a length of 2w bits. There are two parts to the plaintext block. 0 LE and 0 RE. The two parts of the data are processed n times, and then combine to create the block of ciphertext. The inputs for each iteration of I are LEi-1 and REi-1 and a subkey Ki were obtained from the preceding round created using the overall K. Typically, the subkeys Ki are unique from K and one another. Although any number of rounds might be employed, 16 rounds are used the format is the same for every round. On the left, a replacement is made data in the middle. This is accomplished by rounding the right-hand side of the data, taking the exclusive-OR of that function's output, and then taking the left data in the middle. The

main structure of the round function is the same for each round nonetheless, the round subkey Ki parameterizes.

Key size: While a larger key size increases security, it may also reduce encryption rate of decryption. Higher resilience to these threats results in greater security assaults using force and further confusion. There are currently no keys larger than 64 bits often seen as being insufficient, and 128 bits has evolved into a typical amount.

The Feistel cipher's fundamental principle is that there is only one round although several rounds provide increasing security, delivers insufficient security 16 rounds is a standard size. Subkey creation method: This algorithm should be more complicated increase the complexity of cryptanalysis. Again, more complexity often equates to more cryptanalysis resistance two more factors are taken into account while creating a Feistel cipher: Fast software encryption and decryption: Encryption is often incorporated in software in utilities or programs in a manner that prevents hardware implementation. As a result, the algorithm's speed of execution becomes a

Analysis is simple, despite our desire to make our system as challenging as possible. Making the algorithm simple has significant benefits even if it is crypt analyzable to examine specifically, if the method can be succinctly and precisely described, it is simpler to examine that algorithm for cryptographic flaws and hence increase your sense of confidence in its power.

A Feistel cipher's decryption procedure is virtually identical to how encryption works. The following is the rule use ciphertext as the algorithm's input but reverse the order of the subkeys Ki. That use Kn in the first round, Kn-1 in the next, and so on until K1 is used in the last round the last contest. This is a useful feature since it eliminates the need for two distinct algorithms—one for encryption and the other for decryption and a 16-round algorithm's decryption operation moving along the right side.

For simplicity, we refer to data passing through the encryption process as LEi and REi and data passing through the decryption procedure as LDi and RDi. According to the illustration, every round's intermediate value of the matching value of the encryption process is identical to the decryption process with the value's two halves switched. In other words, let the output be LEi 'REi of the ith encryption round (LEi concatenated with REi). The result of the (16 - i)th decryption cycle is thus REi 'LEi or, alternatively,

A reversible function, F, is not necessary for the derivation. To Take a limiting case where F produces a constant output to illustrate this (e.g., all ones) irrespective of the merits of its two arguments. The equations are valid today. Let's take a closer look at a specific example to help make the previous ideas more clear. Assume that the blocks at each stage are 32 bits (two 16-bit blocks) and concentrate on the fifteenth round of encryption, which corresponds to the second round of decryption.Over time, DES emerged as the most popular symmetric encryption algorithm especially in applications involving money. NIST reiterated its support for DES for use by the federal government in 1994[7], [8].

NIST advised using DES for applications other than encryption for an additional five years than the safeguarding of sensitive information. NIST released an updated version in 1999 of its

specification (FIPS PUB 46-3), which stated that only DES should be used, triple DES (which essentially repeats the DES algorithm) is recommended for legacy systems.

Using two or three different keys, run the DES algorithm three times on the plaintext be used to generate the ciphertext. Due to the For DES and triple DES, the underlying encryption and decryption algorithms are the same. It is still crucial to comprehend the DES cipher. This paragraph presents is a little perplexing. The terms DEA and DES have previously been used synonymously. However, a description of the DEA is included in the most recent edition of the DES document along with the triple DEA (TDEA) DEA and TDEA are both a part of

DES, or the Data Encryption Standard Additionally, prior to the recent official adoption of the term TDEA, the triple DEA algorithm was typically referred to as triple DES and written as 3DES. For the sake of convenience, this is followed by a phase consisting of sixteen rounds of the same function, which involves both permutation and substitution functions. The output of the last (sixteenth) round consists of 64 bits that are a function of the input plaintext and the key. The left and right halves of the output are swapped to produce.

We now work through an example and consider some of its implications. Although are not expected to duplicate the example by hand, you will find it informative to study the hex patterns that occur from one step to the next. For this example, the plaintext is a hexadecimal palindrome. The plaintext, key, and resulting ciphertext are as follows values of the left and right halves of data after the initial permutation. The next rows show the results after each round. Also shown is the value of the 48-bit subkey generated for each round. Note that Li = Ri-1. The final row shows the left- and right-hand values after the inverse initial permutation. A desirable property of any encryption algorithm is that a small change in either the plaintext or the key should produce a significant change in the ciphertext. In particular, a change in one bit of the plaintext or one bit of the key should produceabout a year; if multiple PCs work in parallel, the time is drastically shortened. And today's supercomputers should be able to find a key in about an hour. Key sizes of 128 bits or greater are effectively unbreakable using simply a brute-force approach number of rounds is chosen so that known cryptanalytic efforts require greater[9], [10].

This criterion is attractive, because it makes it easy to judge the strength of an algorithm and to compare different algorithms. In the absence of a cryptanalytic breakthrough, the strength of any algorithm that satisfies the criterion can be judged solely on key length. The heart of a Feistel block cipher is the function F, which provides the element of confusion in a Feistel cipher. Thus, it must be difficult to "unscramble" the substitution performed by F. One obvious criterion is that F be nonlinear, as we discussed previously. The more nonlinear F, the more difficult any type of cryptanalysis will be.

There are several measures of nonlinearity, which are beyond the scope of this book. In rough terms, the more difficult it is to approximate F by a set of linear equations, the more nonlinear F. Several other criteria should be considered in designing F. We would like the algorithm to have good avalanche properties. Recall that, in general, this means that a change in one bit of the input should produce a change in many bits of the output. A more stringent version of this is the strict avalanche criterion (SAC). which states that any output bit j of an S-box (see Appendix S for a

discussion of S-boxes) should change with probability 1/2 when any single input bit I is inverted for all I j. Although SAC is expressed in terms of S-boxes, a similar criterion could another criterion proposed is the bit independence criterion.

REFERENCES:

[1] T. Cui, H. Chen, S. Mesnager, L. Sun, and M. Wang, "Statistical integral distinguisher with multi-structure and its application on AES-like ciphers," *Cryptogr. Commun.*, 2018, doi: 10.1007/s12095-018-0286-5.

[2] J. Wang and Q. Ding, "Dynamic rounds chaotic block cipher based on keyword abstract extraction," *Entropy*, 2018, doi: 10.3390/e20090693.

[3] M. A. Alahmad, "Design of a new cryptographic hash function – Titanium," *Indones. J. Electr. Eng. Comput. Sci.*, 2018, doi: 10.11591/ijeecs.v10.i2.pp827-832.

[4] L. Li, B. Liu, Y. Zhou, and Y. Zou, "SFN: A new lightweight block cipher," *Microprocess. Microsyst.*, 2018, doi: 10.1016/j.micpro.2018.04.009.

[5] T. Ichiki and A. Tsuneda, "Study on Security Enhancement of 64-Bit NFSR-based Block Cipher Systems with Ring Structure," in *9th International Conference on Information and Communication Technology Convergence: ICT Convergence Powered by Smart Intelligence, ICTC 2018*, 2018. doi: 10.1109/ICTC.2018.8539674.

[6] A. G. Buja, S. F. Abdul-Latip, and R. Ahmad, "Fault analysis of the KTANTAN Family of block ciphers: A revisited work of fault analysis of the KATAN family of block ciphers," *J. Telecommun. Electron. Comput. Eng.*, 2018.

[7] M. Sajadieh and M. Vaziri, "Using MILP in analysis of feistel structures and improving type II GFS by switching mechanism," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018. doi: 10.1007/978-3-030-05378-9_15.

[8] D. Hong, B. Koo, and C. Seo, "Differential property of PRESENT-like structure," *Discret. Appl. Math.*, 2018, doi: 10.1016/j.dam.2016.03.015.

[9] Y. Todo, T. Isobe, Y. Hao, and W. Meier, "Cube attacks on non-blackbox polynomials based on division property," *IEEE Trans. Comput.*, 2018, doi: 10.1109/TC.2018.2835480.

[10] V. Nachef, J. Patarin, and E. Volte, "Generic attacks with standard deviation analysis on a-feistel schemes," *Cryptogr. Commun.*, 2018, doi: 10.1007/s12095-017-0244-7.

CHAPTER 8

FINITE FIELD OF THE FORM AND GROUPS

Dr. S.P. Anandaraj, Professor and HOD
Department of Computer Science and Engineering, Presidency University, Bangalore, India
Email Id- anandaraj@presidencyuniversity.in

A finite field is a mathematical concept used in many areas of computer science, including cryptography and cybersecurity. It is a set of elements, along with operations that can be performed on those elements. Finite fields are often used in encryption algorithms, as they allow for fast and efficient computation of complex mathematical operations, while also providing a high degree of security. One important type of finite field is the field of the form GF(p), where p is a prime number. This field consists of integers between 0 and p-1, and arithmetic is performed modulo p. For example, if p = 7, then the elements of GF(7) are {0, 1, 2, 3, 4, 5, 6}, and the addition and multiplication operations are defined as follows:

a) Addition: (a + b) mod 7

b) Multiplication: (a * b) mod 7

These operations satisfy all of the usual properties of addition and multiplication, such as associativity, commutativity, and distributivity. One important property of finite fields is that every non-zero element has a multiplicative inverse. This means that for every a in GF(p), there exists an element b such that a * b = 1 mod p. This property is essential for many cryptographic algorithms, as it allows for the creation of one-way functions that are difficult to reverse.

Another important type of finite field is the field of the form GF(2^n), where n is a positive integer. This field consists of binary strings of length n, and arithmetic is performed using a special type of addition and multiplication called binary polynomial arithmetic. For example, in GF(2^3), the elements are {000, 001, 010, 011, 100, 101, 110, 111}, and the addition and multiplication operations are defined as follows:

a) Addition: XOR (exclusive OR) of the binary strings

b) Multiplication: multiplication of binary polynomials modulo an irreducible polynomial of degree n

In this field, every non-zero element also has a multiplicative inverse. This type of finite field is particularly important in cryptography, as it allows for the efficient implementation of many cryptographic algorithms, including the Advanced Encryption Standard (AES) and the elliptic curve cryptosystems (ECC).

Groups are also an important concept in cybersecurity, as they provide a way to model the behavior of systems and networks. A group is a set of elements, along with an operation that

combines two elements to produce a third element. The operation must satisfy four properties: closure, associativity, identity, and inverse. These properties ensure that the group is a well-behaved mathematical object, and they also have important implications for cybersecurity.

One important use of groups in cybersecurity is in the construction of cryptographic primitives, such as block ciphers and hash functions. Groups can be used to create functions that are difficult to invert, which is a key requirement for many cryptographic applications. For example, the function f(x) = x^3 mod p, where p is a large prime number, is a one-way function in the group of integers modulo p. It is easy to compute f(x) for any x, but it is very difficult to compute x given f(x).

Another important use of groups in cybersecurity is in the analysis of algorithms and protocols. Groups can be used to model the behavior of systems and networks, and to analyze the security properties of algorithms and protocols. For example, the Diffie-Hellman key exchange protocol, which is widely used in cryptographic systems, can be analyzed using the group of integers modulo a large prime number. This analysis can reveal potential weaknesses in the protocol, and can also suggest modifications to improve its security. In addition to the uses of finite fields and groups in encryption algorithms and the construction of cryptographic primitives, these mathematical concepts also have important applications in other areas of cybersecurity[1], [2].

One important application of finite fields is in error-correcting codes. An error-correcting code is a method for detecting and correcting errors that may occur when data is transmitted over a noisy channel, such as a wireless network or a satellite link. Finite fields are used to define the algebraic structure of the code, and to perform the mathematical operations needed to encode and decode the data. The Reed-Solomon code, which is widely used in applications such as digital audio and video recording, is an example of an error-correcting code that uses finite fields.Another important application of finite fields is in the design of secure multi-party computation (MPC) protocols. MPC is a cryptographic technique that allows multiple parties to compute a function on their private inputs, without revealing those inputs to each other. Finite fields are used to define the algebraic structure of the computation, and to perform the mathematical operations needed to securely combine the inputs. MPC protocols have important applications in areas such as electronic voting, financial transactions, and cloud computing.

Groups also have important applications in the analysis and design of cryptographic protocols. One important area of research is the study of group-based cryptography, which uses groups to create new cryptographic primitives and protocols that are resistant to attacks based on the discrete logarithm problem. The discrete logarithm problem is a mathematical problem that is used as the basis for many encryption algorithms, and its hardness is the foundation for the security of these algorithms. Group-based cryptography is an active area of research, and has led to the development of new cryptographic schemes such as the pairing-based cryptosystems.Another important use of groups in cybersecurity is in the analysis of network traffic. Groups can be used to model the behavior of network traffic, and to identify patterns that may indicate the presence of malicious activity, such as a distributed denial-of-service (DDoS) attack or a botnet. The group structure can also be used to model the interactions between

different elements of the network, such as routers, switches, and hosts, and to analyze the impact of various attacks and defenses.

Groups are also used in the analysis and design of access control policies, which are used to restrict the access of users to resources in a computer system or network. Groups can be used to define the set of users who have access to a particular resource, and to specify the conditions under which that access is granted. Group-based access control policies have important applications in areas such as healthcare, finance, and government, where the protection of sensitive data is critical.

In addition to the above applications, finite fields and groups have many other uses in cybersecurity, including in the design of digital signatures, in the analysis of network protocols, and in the analysis of the security of hardware and software systems. The use of these mathematical concepts has led to the development of many new techniques and protocols that are critical to the security of modern computer systems and networks[3], [4].

One of the challenges in the use of finite fields and groups in cybersecurity is the selection of appropriate parameters, such as the size of the field or the choice of the group. The security of many cryptographic algorithms is based on the difficulty of certain mathematical problems, such as factoring large integers or solving the discrete logarithm problem. As computing power continues to increase, the parameters used in these algorithms must be updated to ensure their continued security. This requires ongoing research and development in the field of finite fields and groups, as well as in other areas of cryptography and cybersecurity.

In cryptography, finite fields have grown in importance. AES and elliptic curve cryptography are two examples of cryptographic methods that strongly depend on the characteristics of finite fields. Other instances are the authorized encryption and the message authentication code CMAC.

This chapter gives the reader enough information on the ideas behind finite fields to enable them to comprehend how AES and other cryptographic algorithms that make use of finite fields were created. We approach the material in a manner that is intended to improve comprehension since students who are not acquainted with abstract algebra may find the ideas underlying finite fields to be fairly challenging to comprehend. Our strategy is as follows:

1. The wider class of algebraic structures known as rings, which in turn is a subset of the class of groups, is a subset of fields. In reality, both groups and rings may be further distinguished. Groups are simply understood and defined by a straightforward set of attributes. The qualities added by each succeeding subset—Abelian group, ring, commutative ring, and so forth increase its complexity. Following each other, sections 5.1 through 5.3 will look at groups, rings, and fields.
2. Fields having a limited number of elements are included in the category of finite fields. These kind of fields are what cryptography algorithms often use. With a solid understanding of fields, we shift to a particular category of finite fields those

with p elements, where p is prime. Such fields are used by several asymmetric cryptography techniques.

3. For cryptography, a more significant type of finite fields are those with 2n members, sometimes known as fields of the form GF(2n). Many different cryptography algorithms use them. However, we must first examine the subject of polynomial arithmetic, which is done in Section 5.5, before talking about these fields. After completing all of this preparatory work, we may finally examine finite fields of the type GF(2n) in the reader may want to go through through, which discuss pertinent number theory issues, before continuing.

You should be able to: Differentiate between groups, rings, and fields after reading this chapter Define GF-format finite fields (p). The distinctions between regular polynomial arithmetic, polynomial arithmetic with Zp coefficients, and modular polynomial arithmetic in GF(2n) should be explained.

The basic building blocks of abstract algebra, often known as contemporary algebra, are groups, rings, and fields. In abstract algebra, we are interested in sets whose components can be operated on algebraically; in other words, we may combine two elements of the set to get a third element of the set, perhaps in a number of different ways.

Specific regulations that outline the nature of the set are applicable to these activities. It is customary to use the same notation for addition and multiplication on ordinary numbers as well as the two main types of operations on set components. It's crucial to remember that in abstract algebra, we are not constrained to using standard arithmetic operations. This should all become evident as we go along[5], [6].

Define Sn as the collection of all possible combinations of n unique symbols. A permutation p of the numbers in 1, 2, n represents each element of Sn.

1. A1: The composite mapping p # r is created by permuting the elements of r in accordance with the permutation p if (p, r Sn). As an example, 3, 2, 1 # 1, 3, 2 = 2, 3, 1. Here is an explanation of the notation used for this mapping: The value of the first element of p identifies which element of r should occupy the first place in p # r, and so on. The value of the second element of p identifies which element of r should occupy the second position in p # r. obviously, p # r Sn.
2. A2: It is also simple to demonstrate that the combination of maps is associative. A3: The permutation that doesn't change the order of the n items is called an identity mapping. The identity element for Sn is 1, 2, c, and n. A4: The mapping that reverses the permutation described by p is the inverse element for p for any p Sn. Such an inverse will always exist. As an example, "2, 3, 1" # "3, 1, 2" = "1, 2, 3." This is analogous to the definition of permutation given in Chapter 2, which stated that a permutation of a finite set of items S is an ordered sequence in which each element of S appears precisely once.

An abelian group is the collection of integers (positive, negative, and 0) under addition. An abelian group is the collection of real numbers that are not zero under multiplication. For n 7 2, the set Sn from the previous example is a group but it is not an abelian group. A group is referred

to be a finite group if it has a limited number of elements, and its order is determined by the number of elements. The group is an infinite group if such is the case. If a group also meets the following requirement, it is said to be abelian:

Circular Group

Exponentiation inside a group is defined as the group operator being applied more than once, as in a3 = a # a # a. additionally, we define a-n = (a′) n, where a′ is the inverse element of an inside the group, and a0 = e as the identity element. If every element of a group G is a power of a fixed element a, then the group G is cyclic (k is an integer). It is referred to as a generator of the group G or a generator of the element a. A cyclic group may be finite or infinite and is always abelian.

The element 1 is the source of the infinite cyclic group that makes up the additive group of integers. Powers are interpreted additively in this situation, therefore n is the nth power of 1. A ring R is a collection of elements with two binary operations, addition and multiplication, 3 such that the axioms are upheld for every a, b, and c in R. A ring R is commonly symbolized by the symbols R, +, *. R is an abelian group with regard to addition (A1-A5), which means that R is consistent with axioms A1 through A5. The identity element and the opposite of an are denoted as 0 and -a, respectively, for the case of an additive group. Closure under multiplication (M1): If a and b are members of R, then ab is likewise a member of R. (M2) Multiplication associativity: For any values of a, b, and c in R, a(bc) = (ab)c. (M3) Distributive laws: For any a, b, and c in R, a(b + c) = ab + ac. For any a, b, and c in R, (a + b)c = ac + bc.

A ring is essentially a collection of elements in which addition, subtraction [a - b = a + (-b)], and multiplication may be performed without leaving the collection. In most cases, we concatenate two components to represent multiplication instead of using the multiplication sign *.

The set of all n-square matrices over the real numbers is a ring in terms of addition and multiplication. If a ring also meets the following prerequisite, it is said to be commutative: The next step is to define an integral domain, a commutative ring that adheres to the axioms listed below.

A. There is an element 1 in R such that a1 = 1a = a for each an in R (M5) Multiplicative identity. No zero divisors (M6) If R, a, and b are all zero, then either an or b must be zero.
B. Let S represent the collection of even numbers that can be added and multiplied using the standard operations (positive, negative, and 0). A commutative ring is S. A commutative ring does not exist for the collection of all n-square matrices specified in the previous example. A commutative ring is the set Zn of the numbers 0 through 1, c through n–1, and the arithmetic operations modulo n.
C. Let S represent the collection of numbers (positive, negative, and 0) that may be added and multiplied using standard operations. An integral domain is S.
D. The rational numbers, real numbers, and complex numbers are well-known examples of fields. It should be noted that only the elements 1 and -1 have multiplicative inverses in the integers, meaning that the set of all integers is not a field since every element in the set does not have a multiplicative inverse.

A field F is a set of elements with two binary operations, addition and multiplication, such that for any a, b, and c in F the following axioms are upheld. A field F is commonly represented by F, +, *. (A1–M6) F fulfills axioms A1 through A5 and M1 through M6, making it an integral domain.

(M7) Inverse multiplication: There is an element a-1 in F such that aa-1 = (a-1)a = 1 for any an in F other than 0. A field is essentially a collection of items in which addition, subtraction, multiplication, and division may be performed without leaving the collection. The following formula defines division: a/b = a(b-1).

The following alternative characterisation may be helpful in understanding fields. A field F, represented by "F, +," is a collection of items that may be combined and multiplied using two binary operations under the following circumstances:

1. In terms of addition, F forms an abelian group.
2. In terms of multiplication, the nonzero components of F constitute an abelian group.

Finite Fields Of The Form Gf(p),

3. The law of distribution is valid. This means that (a + b)c = ac + bc for any a, b, and c in F.
4. The axioms that describe groups, rings, and fields.

We described fields, along with several instances of infinite fields, as a set that complies with all of the axioms. The area of cryptography is not particularly interested in infinite fields. Finite fields are important in many cryptographic techniques, although there are two sorts of them in addition to infinite fields. The order of a finite field (number of elements in the field) may be shown to require that n be a positive integer and pn be a power of a prime. It is common to write the finite field of order pn as GF(pn); GF stands for Galois field, named after the mathematician who pioneered the study of finite fields. For our needs, two unique examples are of interest. For n = 1, we get the finite field GF(p). This finite field is investigated in this section and has a distinct structure than finite fields with n. GF(2n) fields are of special cryptographic importance for finite fields of the type GF(pn), and are Finite Fields of Order, number six.

We define the finite field of order p, GF(p), for a given prime, p, as the set Zp of numbers 0–1, c–p–1, and p–1 arithmetic operations. Therefore, take note that we are defining the operations over these fields using standard modular arithmetic. (A2) Associativity of addition: For any a, b, and c in S, a + (b + c) = (a + b) + c (A3) Additive identity: There is a 0 in R such that for any an in S, a + 0 = 0 + a = a. (A4) For any element an in S, there is an element -a in S such that a + (-a) = (-a) + a = 0 (additive inverse) (A5) Commutativity of addition: For any a, b in S, a + b = b + a (M1) Closure under multiplication: If S includes a and b, then S also includes ab (M2) Multiplication associativity: a(bc) = (ab)c for any a, b, and c in S (M3) Distributive laws: For every a, b, and c in S, a(b + c) = ab + ac, and for all a + b, c = ac + bc (M4) Multiplication commutativity: ab = ba for any a, b in S (M5) Multiple identities: For every of the a's in S, there is an element 1 in S such that a1 = 1a = a. (M6) Without zero divisors: If S and ab are both 0,

then either an or b must be zero (M7) If an is a member of S and if a 0, then a-1 is an element in S such that aa -1 = a-1a = 1 Group Ring. Recall that we demonstrated in Section 5.2 that the commutative ring Zn is the set of numbers 0–1, c–n–1, and the arithmetic operations modulo n. In addition, we discovered that each integer in Zn has a multiplicative inverse if and only if it is comparatively prime to n. There is a multiplicative inverse for every nonzero integer in Zn if n is prime because all of the nonzero integers in Zn are relatively prime to n. As a result, for Zp, the following attributes may be added.

The residues produced when we multiply all of the Zp elements by w are all of the Zp elements permuted since w is relatively prime to p. As a result, one residue only carries the value 1. Consequently, there is some integer in Zp that, when multiplied by w, gives the result 1, or the residue. That number, denoted as w-1, is w's multiplicative inverse. Consequently, Zp is a finite field. In addition, Equation (2.5) may be recast without the condition and is compatible with the presence of a multiplicative inverse: if (a * b) K (a * c)(mod p), then b K c(mod p) (5.1)

Equation is formed by multiplying both sides by the multiplicative inverse of a, giving us ((a-1) * a * b) K ((a-1) * a * c)(mod p) b K c. (mod p) Two numbers are considered to be relatively prime, as mentioned in the discussion of Equation (2.5), if their sole shared positive integer component is 1. Multiplication 0 0 - 11 1 Inverses w -w w-1 Multiplication is identical to the logical While operation in this scenario, and addition is comparable to the exclusive-OR (XOR) action.

When employing modular arithmetic modulo 7, this is a field of order 7. As can be seen, it fits every need for a field. Compare this to Table 5.1's left side, which is a copy of Table 2.2. In the latter scenario, modular arithmetic modulo 8 reveals that the set Z8 is not a field. We demonstrate how to define addition and multiplication operations on Z8 in a manner that results in a finite field.For small values of p, it is simple to determine the multiplicative inverse of an element in GF(p). The required result may be read straight from a multiplication table, such as the one in Table 5.1e. However, this strategy is impractical for high values of p. B has a multiplicative inverse modulo an if a and b are both relatively prime numbers.

The multiplicative inverse of b is modulo an if gcd(a, b) = 1. That is, there exists a b-1 6 a such that bb-1 = 1 mod a for positive integer b 6 a. If an is a prime integer and b is 6 a, then it is obvious that a and b are relatively prime and have a GCD of 1. We now demonstrate how the extended Euclidean method makes it simple to calculate b-1. Here, we, whose solution we previously demonstrated using the extended Euclidean algorithm:

We now get ax + by = 1 if gcd(a, b) = 1. We may write [(ax mod a) + (by mod a)] using the fundamental equality of modular arithmetic, described in Section 2.3. mod a 0 + (by mod a) = 1 mod a = 1. However, y = b-1 if by mod a = 1 instead. As a result, Equation (2.7) may be solved using the extended Euclidean method to determine the value of the multiplicative inverse of b if gcd(a, b) = 1. Here, an is equal to 1759, a prime number, while b is equal to 550. Y = 355 is the result of solving the equation 1759x + 550y = d. Thus, b-1 = 355. To be sure, we do the calculation 550 * 355 = 195250 mod 1759 = 1.

More broadly, the multiplicative inverse in Zn for any n may be discovered using the extended Euclidean approach. When the extended Euclidean technique is used to solve the equation nx + by = d and the result is that d = 1, y = b-1 in Zn w-1 (f) Inverses of addition and multiplication modulo 7. This section has shown how to create a finite field of order p, where p is a prime number. We defined GF(p) using the specific characteristics shown below.

1. P components make up GF(p).
2. The set is defined over the binary operations + and *.

Without leaving the set, addition, subtraction, multiplication, and division may be carried out. Other than 0, each member of the set has a multiplicative inverse, and division is achieved by multiplying the original value by the multiplicative inverse. We have shown that the components of GF(p) are the numbers 0, 1, c, and p-1 and that addition and multiplication mod p are the arithmetic operations.

We need to briefly explain polynomial arithmetic before moving on to our consideration of finite fields. We can identify three kinds of polynomial arithmetic since we are only interested in polynomials in a single variable, x. Common polynomial computation utilizing the fundamental algebraic principles. Polynomial arithmetic in which the coefficients are in GF and the computation on them is done modulo p. Polynomial arithmetic with coefficients in GF(p) and polynomials defined modulo a polynomial m(x) whose greatest power is an integer n.

The first two courses are covered in this part, while the last class is covered in the next section regular polynomial computation f(x) = anxn + an-1xn-1 + g + a1x + a0 = a n i=0 ai xi, where the ai are components of a chosen set of integers S, referred to as the coefficient set, and a 0. A polynomial of degree n (integer n 0) is an expression of the form. Such polynomials are referred to as being defined over the S coefficient set[7], [8].

A zero-degree polynomial, often known as a constant polynomial, is only one of the coefficients in the set. If a = 1, a polynomial of the nth degree is referred to as a monic polynomial. In the domain of abstract algebra, evaluating a polynomial for a specific value of x is often not of relevance. The word "indeterminate" is occasionally used to describe the variable x to underline this point.

The addition, subtraction, and multiplication operations are all included in polynomial arithmetic. The definitions of these procedures are natural, as if the variable the polynomial f(x) x is assessed for a specific value of x x and is considered as an indeterminate x made up a part of S. Similar definitions apply to division, but S must be a field. Real numbers, rational numbers, and Zp for p prime are a few examples of fields. The set of all integers is not a field and does not permit polynomial division, it should be noted.Finally, we demonstrated how to use the Euclidean technique to identify the largest common factor of two polynomials whose coefficients are parts of a field. This whole section lays the groundwork for the definition of finite fields of order pn using polynomials in the part that follows.

In a previous section of this chapter, we stated that a finite field's order must have the form pn, where p is a prime number and n is a positive integer. We examined the unique situation of finite

fields with order p in Section 5.4. We discovered that all of the axioms for a field are fulfilled when employing modular arithmetic in Zp. Operations modulo pn for polynomials over pn with n = 7 1 do not result in a field. In this part, we focus on GF(2n) and demonstrate what structure meets the axioms for a field in a collection of pn elements.

Motivation

The majority of encryption techniques, both symmetric and asymmetric, require operations on integers using arithmetic. If division is one of the operations the algorithm uses, then we must deal with arithmetic defined over a field.

Additionally, we'd want to deal with numbers that precisely fit within a certain amount of bits, with no unused bit patterns, for implementation speed. In other words, we want to deal with numbers that fit into an n-bit word and are in the range 0 through 2n – 1. Let's say we want to conduct division on a standard encryption technique that processes data 8 bits at a time. We can express numbers from 0 to 255 using 8 bits. However, as 256 is not a prime number, this collection of integers will not constitute a field if calculations are made in Z256 (arithmetic modulo 256). 251 is the nearest prime number that is smaller than 256. Therefore, using arithmetic modulo 251, the set Z251 is a field. This would result in an inefficient use of storage since the 8-bit patterns for the numbers 251 through 255 would not be utilized.

As shown by the above example, arithmetic modulo 2n will not function if all arithmetic operations are to be employed and we want to express the whole range of integers in n bits. The set of integers modulo 2n for n 7 1 is equivalently not a field. Furthermore, the usage of the set Z2n is dubious even if the encryption technique simply employs addition and multiplication and not division, as shown in the example below. Let's say we want to employ 3-bit blocks and solely addition and multiplication in our encryption technique. Thus, as seen in Table 5.1, arithmetic modulo 8 is properly defined. The nonzero integers do not, however, occur in the multiplication table equally often. For instance, there are only four instances of the number three, yet twelve of the number four. However, as was previously established, there exist finite fields of the type GF(2n), and one in particular has an order of 23 = 8. In Table 5.2, the math for this field is shown. The frequency of the nonzero integers is uniform in this situation for multiplication.

It seems sense that a method that offers an unequal mapping of the numbers onto themselves could be cryptographically weaker than one that does so uniformly. In other words, a cryptanalytic approach could be able to take advantage of the fact that certain numbers appear in the ciphertext more often than others. Thus, cryptographic techniques are drawn to finite fields of the kind GF(2n). In order to establish a field, we need to find a set of 2n items along with a definition of addition and multiplication over the set. Each component of the set may be given a distinct number in the range 0 through 2n – 1. Remember that we won't utilize modular arithmetic since we've already shown that it doesn't produce a field. Each such set S is a finite field if arithmetic operations are defined appropriately. The following components make up the definition.

1. With the following two improvements, algebraic arithmetic adheres to the standard laws of polynomial arithmetic.
2. The coefficients are used in modulo p arithmetic. In other words, we apply the arithmetic rules to the finite field Zp.
3. If a polynomial of degree more than n - 1 is produced through multiplication, the polynomial is reduced modulo an irreducible polynomial of degree n called m(x).

In other words, we maintain the residue after dividing by m(x). The remainder for a polynomial f(x) is written as r(x) = f(x) mod m. (x). The irreducible polynomial m(x) = x8 + x4 + x3 + x + 1 and arithmetic in the finite field GF(28) are used in the Advanced Encryption Standard (AES). The two polynomials f(x) = x6 + x4 + x2 + x + 1 and g(x) = x7 + x + 1 are being considered. Then f(x) + g(x) = x6 + x4 + x2 + x + 1 + x7 + x + 1 = x7 + x6 + x4 + x2 f(x) * g(x) = x13 + x11 + x9 + x8 + x6 + x5 + x3 x8 + x4 + x3 + x + 1 >x13

We have the concept of a set of residues in modular polynomial arithmetic, just as in regular modular arithmetic. There are pn items in the set of residues modulo m(x), an nth-degree polynomial. One of the pn polynomials of degree m 6 n serves as the representation for each of these components. All polynomials a(x) such that a(x) K (x + 1)(mod m(x)) belong to the residue class [x + 1], (mod m(x)). In other words, any polynomials that fulfill the equation a(x) mod m(x) = x + 1 belong to the residue class [x + 1].

It can be shown that the axioms in Figure 5.2 are satisfied by the set of all polynomials modulo an irreducible nthdegree polynomial m(x), which results in the formation of a finite field. Additionally, all finite fields of a particular order are isomorphic, meaning that any two finite-field structures of a particular order have the same structure, though the element [9], [10].

REFERENCES:

[1] Z. Sun, T. Matsuno, and H. Isobe, “Stereoisomerism and structures of rigid cylindrical cycloarylenes,” *Bull. Chem. Soc. Jpn.*, 2018, doi: 10.1246/bcsj.20180051.

[2] D. M. Hofman and N. Iqbal, “Generalized global symmetries and holography,” *SciPost Phys.*, 2018, doi: 10.21468/SciPostPhys.4.1.005.

[3] N. Bhaskhar, V. Chernousov, and A. Merkurjev, “The norm principle for type D_n groups over complete discretely valued fields,” *Trans. Am. Math. Soc.*, 2018, doi: 10.1090/tran/7558.

[4] O. A. Castro-Alvaredo, C. De Fazio, B. Doyon, and I. M. Szécsényi, “Entanglement content of quantum particle excitations. Part I. Free field theory,” *J. High Energy Phys.*, 2018, doi: 10.1007/JHEP10(2018)039.

[5] A. Argáez-García and J. Cremona, “Black Box Galois representations,” *J. Algebr.*, 2018, doi: 10.1016/j.jalgebra.2018.05.017.

[6] M. Cederwall and J. Palmkvist, “Extended geometries,” *J. High Energy Phys.*, 2018, doi: 10.1007/JHEP02(2018)071.

[7] D. Khleborodov, "Fast elliptic curve point multiplication based on binary and binary non-adjacent scalar form methods," *Adv. Comput. Math.*, 2018, doi: 10.1007/s10444-017-9581-5.

[8] A. S. Sivatski, "On common zeros of a pair of quadratic forms over a finite field," *Finite Fields their Appl.*, 2018, doi: 10.1016/j.ffa.2018.01.007.

[9] W. Lu, R. Lubbad, A. Shestov, and S. Løset, "Parallel channels' fracturing mechanism during ice management operations. Part I: Theory," *Cold Reg. Sci. Technol.*, 2018, doi: 10.1016/j.coldregions.2018.07.010.

[10] M. I. Mishchenko and M. A. Yurkin, "Impressed sources and fields in the volume-integral-equation formulation of electromagnetic scattering by a finite object: A tutorial," *J. Quant. Spectrosc. Radiat. Transf.*, 2018, doi: 10.1016/j.jqsrt.2018.04.023.

CHAPTER 9

SYMMETRIC ENCRYPTION PRINCIPLES

Dr. M. Chandra Sekhar, Professor & HOD
Department of Computer Science and Engineering, Presidency University, Bangalore, India
Email Id- mchandrasekhar@presidencyuniversity.in

Symmetric encryption is a method of encrypting messages or data that involves the use of a single secret key for both encryption and decryption. This key is used to transform plaintext, or unencrypted data, into ciphertext, or encrypted data, which can only be transformed back into plaintext with the same key. Symmetric encryption is widely used in information security to protect data from unauthorized access and interception.

History of Symmetric Encryption

Symmetric encryption has a long history, dating back to ancient civilizations that used secret codes and ciphers to protect important messages. In the early days of modern computing, symmetric encryption algorithms were used to protect data on mainframe computers and other early computing systems. One of the earliest symmetric encryption algorithms was the Data Encryption Standard (DES), which was developed by IBM in the 1970s and later adopted by the US government. DES was widely used for over two decades, but its 56-bit key length was considered insufficient for modern security requirements. Today, symmetric encryption algorithms such as Advanced Encryption Standard (AES), Blowfish, and Twofish are widely used in commercial and government applications. These algorithms use longer key lengths and more advanced encryption techniques to provide stronger security.

Basic Concepts of Symmetric Encryption

Symmetric encryption involves two basic concepts: encryption and decryption. Encryption is the process of transforming plaintext into ciphertext, while decryption is the process of transforming ciphertext back into plaintext. The secret key used for encryption and decryption must be kept confidential and shared only with authorized users. If the key falls into the hands of an unauthorized user, they may be able to decrypt the encrypted data and gain access to sensitive information. Symmetric encryption is typically used to protect data at rest, such as files stored on a hard drive or transmitted over a network. It is less well-suited for protecting data in transit, such as data being transmitted over the internet, because the key must be shared between the sender and receiver, which creates a vulnerability to interception.

Key Generation

The key used for symmetric encryption must be generated with a secure random number generator or derived from a secure key generation algorithm. The key should be long enough to resist attacks by brute force, which involves trying every possible key until the correct one is found. Key management is a critical aspect of symmetric encryption, as the security of the system depends on the confidentiality of the key. The key must be kept confidential and shared only with authorized users.

Encryption Algorithms

Symmetric encryption algorithms use mathematical functions to transform plaintext into ciphertext. The most widely used symmetric encryption algorithm is AES, which uses a block cipher to encrypt data in fixed-size blocks. Block ciphers divide plaintext into fixed-size blocks and use a key to transform each block into a corresponding block of ciphertext. The size of the block is fixed and depends on the cipher used. AES uses a block size of 128 bits, while other ciphers may use block sizes of 64 or 256 bits. The encryption algorithm used depends on the level of security required and the specific application being used. AES is considered one of the most secure symmetric encryption algorithms, with key lengths up to 256 bits[1], [2].

Security Considerations

Symmetric encryption is vulnerable to attacks that attempt to discover the secret key used for encryption and decryption. The most common attack is a brute-force attack, in which an attacker tries every possible key until the correct one is found. Other attacks include cryptanalysis, in which an attacker analyzes the encryption algorithm to discover weaknesses that can be exploited, and side-channel attacks, in which an attacker attempts to discover the secret key by analyzing the behavior of the system or the encryption algorithm.

To protect against these attacks, symmetric encryption systems must use strong encryption algorithms and Symmetric encryption is a fundamental component of information security, used to protect data from unauthorized access and interception. While symmetric encryption has many advantages, it also has some limitations and security considerations that must be taken into account.

One of the main advantages of symmetric encryption is its efficiency. Because the same key is used for both encryption and decryption, symmetric encryption is typically faster and requires less computational resources than asymmetric encryption, which uses separate keys for encryption and decryption. Symmetric encryption is also widely used in commercial and government applications, and is supported by many cryptographic libraries and tools. This makes it easier to implement and use in a wide variety of applications.

However, symmetric encryption also has some limitations that must be considered. One of the main limitations is the key distribution problem. Because the same key is used for encryption and decryption, the key must be shared between the sender and receiver in a secure manner. If an attacker intercepts the key, they can use it to decrypt the encrypted data and gain access to sensitive information.

To mitigate this problem, symmetric encryption systems typically use a key exchange protocol, such as the Diffie-Hellman key exchange, to securely exchange the key without revealing it to an attacker. Another limitation of symmetric encryption is that it does not provide message authentication or integrity protection. This means that an attacker can modify the encrypted data without being detected, or replace the encrypted data with their own data. To protect against these attacks, symmetric encryption systems must use additional cryptographic techniques, such as message authentication codes (MACs) or digital signatures.

Key Generation

The key used for symmetric encryption must be generated with a secure random number generator or derived from a secure key generation algorithm. The key should be long enough to resist attacks by brute force, which involves trying every possible key until the correct one is found.

The key generation process must be performed in a secure manner to ensure that the key is not compromised. If an attacker can guess or discover the key, they can use it to decrypt the encrypted data and gain access to sensitive information. One way to generate a secure key is to use a hardware random number generator, which generates random numbers by measuring physical phenomena, such as thermal noise or radioactive decay. These random numbers can be used to generate a secure key that is resistant to attacks.

Encryption Algorithms

Symmetric encryption algorithms use mathematical functions to transform plaintext into ciphertext. The most widely used symmetric encryption algorithm is AES, which uses a block cipher to encrypt data in fixed-size blocks. Block ciphers divide plaintext into fixed-size blocks and use a key to transform each block into a corresponding block of ciphertext. The size of the block is fixed and depends on the cipher used. AES uses a block size of 128 bits, while other ciphers may use block sizes of 64 or 256 bits. The encryption algorithm used depends on the level of security required and the specific application being used. AES is considered one of the most secure symmetric encryption algorithms, with key lengths up to 256 bits.

Security Considerations

Symmetric encryption is vulnerable to attacks that attempt to discover the secret key used for encryption and decryption. The most common attack is a brute-force attack, in which an attacker tries every possible key until the correct one is found. Other attacks include cryptanalysis, in which an attacker analyzes the encryption algorithm to discover weaknesses that can be exploited, and side-channel attacks, in which an attacker attempts to discover the secret key by analyzing the behavior of the system or the encryption algorithm. To protect against these attacks, symmetric encryption systems must use strong encryption algorithms and key lengths that are resistant to brute-force attacks. They must also use secure key generation techniques to ensure that the key cannot be easily discovered.

Five components make up a symmetric encryption method.

a. **Plaintext:** This is the algorithm's input, the original message or data.
b. **Encryption algorithm:** The encryption method alters the plaintext in a number of different ways.
c. **Secret key:** The algorithm also requires the secret key.

Ciphertext is the output message that has been encrypted. The plaintext and the secret key both play a role. Two separate keys will result in two distinct ciphertexts for the same message.

Decryption algorithm: This method effectively reverses the encryption algorithm. It generates the original plaintext using the ciphertext and the same secret key.

Two conditions must be satisfied for symmetric encryption to be used securely:

1. A reliable encryption algorithm is required. At a minimum, we would want the method to be such that an opponent who understands the algorithm and has access to one or more ciphertexts would be unable to decrypt the ciphertext or find out the key. The common version of this criterion is stronger: Even if the adversary has access to several ciphertexts as well as the plaintext that resulted in each ciphertext, they shouldn't be able to decipher the ciphertext or find the key.
2. The secret key must have been acquired by the sender and recipient in a safe manner, and both parties are required to keep the key safe. If someone can discover the key and understands the algorithm, any communication using this key is readable[3], [4].

It is crucial to remember that the secrecy of the key, not the method, determines how secure symmetric encryption is. To put it another way, it is presumptively impossible to decode a communication using just the ciphertext and a good understanding of the encryption/decryption process. In other words, the only thing we need to keep a secret is the key, not the algorithm.

Symmetric encryption is possible for broad usage because of this characteristic. Manufacturers can and have created low-cost chip implementations of data encryption methods since the algorithm need not be kept a secret. These chips are readily accessible and used in many different goods. With the usage of symmetric encryption, the fundamental security concern is maintaining the confidentiality of the key.

1. The procedures utilized to convert plaintext into ciphertext. All encryption methods are based on two main principles: substitution, in which each element in the plaintext (bit, letter, combination of bits or letters) is mapped into another element, and transposition, in which components in the plaintext are rearranged. The primary need is to ensure that no information is lost (that is, that all operations be reversible). Most systems, referred to as product systems, include numerous phases of replacements and transpositions.
2. The quantity of keys used. If both sender and receiver utilize the same key, the system \sis referred to as symmetric, single-key, secret-key, or conventional encryption. The system is known as asymmetric, two-key, or public-key encryption if the sender and recipient each utilize a distinct key.
3. The method used to process the plaintext. A block cipher creates an output block for every input block it analyzes, one block of elements at a time. A stream cipher continually processes the input elements, generating the output one element at a time.

Cryptanalysis is the process of trying to decipher the plaintext or key. The cryptanalyst's approach is determined by the characteristics of the encryption system and the data at hand. Based on how much information the cryptanalyst is aware lists the many sorts of cryptanalytic

assaults. When just the ciphertext is accessible, the most challenging issue arises. In rare instances, not even the encryption algorithm is known, but generally speaking, we may assume that the adversary is aware of the encryption method. In these conditions, a brute-force assault that tries every key is a possibility. This becomes unworkable if the key space is extremely big. As a result, the opponent is forced to depend on an examination of the ciphertext itself, usually via the use of different statistical tests. A broad understanding of the kind of plaintext that is disguised, such as English or French text, an EXE file, a Java source listing, an accounting file, etc., is required to apply this strategy.

Since the opponent has the least amount of information to deal with, the ciphertext-only attack is the simplest to counter. But often, the analyst is privy to additional details. One or more plaintext communications as well as their encryptions may be captured by the analyzer. Alternatively, the analyst could be aware that a communication would include certain plaintext patterns. For instance, a file that is encoded in the Postscript format always starts with the same pattern, and electronic funds transfer messages may include a standard header or banner, among other things. These are all examples of well-known plaintext. With this information, the analyst may be able to determine the key based on how the known plaintext is altered.

One or more plaintext-ciphertext pairings created using the secret key, known plaintext, an encryption technique, and ciphertext that has to be decoded selected plaintext, the encryption technique, the ciphertext that has to be decoded, and the plaintext message itself, as well as the ciphertext that corresponds to it and was produced using the secret key The chosen ciphertext, the encryption procedure, the ciphertext to be decoded, and the purported ciphertext selected by the cryptanalyst, together with the associated plaintext that was created using the secret key. Text chosen; encryption process; text to be decoded; plaintext message selected by cryptanalyst; and matching ciphertext created using secret key

The message, however, could be known in part if the adversary is searching for a particularly specific piece of information. For instance, if a complete accounting file is being communicated, the adversary may be aware of the location of certain key phrases in the file's header. Another example would be a copyright declaration in a predetermined point in the source code of a software created by a company. A chosen-plaintext attack is feasible if the analyst can persuade the source system to include a message of their choosing in the system. In general, if the analyst has the ability to choose the messages to encrypt, the analyst may purposely select patterns that are likely to betray the key's structure. Despite being less often used as cryptanalytic methods, there are nonetheless attack vectors that might be used. A ciphertext-only assault can only be defeated by algorithms that are somewhat poor. An encryption technique is often built to resist a known-plaintext assault.

If the ciphertext produced by the encryption method satisfies one or both of the following requirements, the encryption method is computationally secure: The cost of breaking the cipher exceeds the value of the encrypted information; the time required to break the cipher exceeds the useful lifetime of the information. Unfortunately, it is quite difficult to gauge the amount of work necessary to correctly cryptanalyze ciphertext. A brute-force technique is suggested, and in this

case we may reasonably estimate expenses and time given there are no inherent mathematical faults in the algorithm[5], [6].

A brute-force strategy entails testing every key until the ciphertext can be deciphered and converted into plaintext. To succeed, on average, 50% of all potential keys must be testeddetails the time commitment for different key sizes. When using the DES (Data Encryption Standard) algorithm, a key size of 56 bits is employed. The results are shown for each key size on the assumption that a single decryption takes 1 s to complete, which is a feasible order of magnitude for modern processors. Processing speeds that are several orders of magnitude higher could be attainable with the utilization of massively parallel groups of microprocessors. The findings for a system that can process one million keys per microsecond are taken into account in the last column. As you can see, DES is no longer deemed computationally safe at this performance level.

The inputs to the encryption process are a plaintext block of length 2w bits and a key K. Horst Feistel of IBM initially outlined the structure of several symmetric block encryption algorithms, including DES, in 1973. The plaintext block is split into LE0 and RE0, which are the two halves. The ciphertext block is created by combining the two halves of the data after n processing iterations. LEi 1 and REi 1 from the round before, as well as a subkey Ki deriving from the overall K, are inputs for each round i. A subkey generation method creates the subkeys Ki from the key; in general, they vary from K and from one another. The format is the same for every round. A replacement is made on the left half \sof the data. The right half of the data is first given a round function F, and the left half of the data is then given the exclusive-OR (XOR) of the function's output. Although the round function's overall structure is the same for every round, it is - -

All symmetric block ciphers employ a more generic structure, with the Feistel structure serving as one specific illustration. A symmetric block cipher typically consists of a series of rounds, with each round carrying out substitutions and permutations that are dependent on the value of a secret key. The selection of the following parameters and design elements determines how exactly a symmetric block cipher is realized.

A. **Block size:** Larger block sizes indicate stronger security (all other factors being \sequal) but lower encryption/decryption performance. Recent block cipher designs almost always use a block size of 128 bits since it is an acceptable trade-off.
B. **Key size:** While a larger key size increases security, it may also slow down encryption and decryption.

In contemporary algorithms, keys typically have a length of 128 bits.

a. **Number of rounds:** A symmetric block cipher's main characteristic is that several rounds provide growing security whereas a single round provides insufficient protection. 16 rounds is a standard size.
b. **Subkey generation method:** Higher complexity in this algorithm should \slead to greater difficulty of cryptanalysis.

c. **Round function:** Once again, more complexity often equates to more cryptanalysis resistance.

b) Two more factors are taken into account while creating a symmetric block cipher:

a. **Rapid software encryption and decryption:** Often, encryption is included into utilities or other software operations, making hardware implementation impossible. The algorithm's speed of execution therefore comes into question.

b. **Ease of analysis:** Although we would want to make our algorithm as difficult to cryptanalyze as possible, there are several advantages to making the method simple to understand. That is, it is simpler to assess an algorithm for cryptanalytic flaws and, as a result, generate a better degree of confidence about its strength, if the method can be succinctly and simply defined. For instance, DES lacks a feature that may be quickly assessed.

A symmetric block cipher's encryption and decryption procedures are almost identical. the following is the rule: Use the ciphertext as the algorithm's input, but apply the subkeys Ki in the other direction. To put it another way, start with Kn in the first round, Kn 1 in the second, and so on until K1 is used in the final round. This is a useful feature since it eliminates the need for two separate algorithms one for encryption and one for decryption to be implemented. Block ciphers are the most widely used symmetric encryption methods.

A block cipher creates a block of ciphertext of the same size for every fixed-sized block of plaintext it analyzes. The Data Encryption Standard (DES), triple DES (3DES), and Advanced Encryption Standard (AES) are the three most significant symmetric block ciphers discussed in this section (AES). The Data Encryption Standard (DES), published in 1977 as Federal Information Processing Standard 46 (FIPS 46) by the National Bureau of Standards, now known as the National Institute of Standards, is the foundation of the most extensively used encryption method[7], [8].

However, it is extremely cautious to assume that one encryption occurs per microsecond. When the Electronic Frontier Foundation (EFF) claimed that it had successfully cracked a DES encryption using a specialized "DES cracker" equipment that was manufactured for less than $250,000 in July 1998, DES ultimately and categorically demonstrated to be vulnerable. It just took a few of days for the onslaught. Others may construct their own cracker thanks to the EFF's publication of a comprehensive description of the device. Furthermore, hardware costs will undoubtedly continue to decline as speeds rise, rendering DES essentially useless.

It is crucial to remember that a key-search assault involves more than just sifting through all potential keys. The analyst must be able to identify plaintext as plaintext unless known plaintext is given. Even though English recognition would need to be automated, the outcome is obvious if the message is merely plain text in English. Recognition is more challenging if the text message has been compressed before encryption. Additionally, the challenge is considerably harder to automate if the message includes some more generic sort of data, such a compressed numerical file. Thus, to enhance the brute-force

The words DEA and DES have previously been used synonymously. However, the most current version of the DES document also specifies the triple DEA (3DES), which is discussed below, in addition to the DEA given above. The Data Encryption Standard includes both DEA and 3DES. Additionally, the triple DEA algorithm was often referred to as triple DES and written as 3DES until the recent introduction of the official designation 3DES. We'll use 3DES for convenience's sake.

One more thing: If brute force assaults are the only ones that can be used to break an encryption system, then using longer keys is the logical solution. Let's base our calculations on the EFF cracker to obtain a general notion of the amount of key needed. The EFF cracker was a prototype, and given current technology, we may expect that a faster machine is more affordable. A DES code would take around 10 hours to break if we assume that a cracker can carry out one million decryptions per second, which is the pace utilized. This is a speedup compared to the EFF result of around a factor. The time required to break a DES-style algorithm as a function of key size using this rate. For instance, employing the EFF cracker would need more than 1018 years to decipher a code with a 128-bit key, which is typical of modern methods. It would still take more than a million years to crack the code, even if we were able to speed up the cracker by a factor of 1 trillion (1012). Therefore, a 128-bit key will result in an algorithm that cannot be cracked by brute force.

The preferred symmetric encryption algorithm is 3DES, which is FIPS-approved. The standard only permits the use of the original DES, which has a single 56-bit key, for legacy systems. New purchases ought to support 3DES. Government agencies using outdated DES systems are urged to switch to 3DES. The Advanced Encryption Standard (AES) and 3DES are expected to coexist as FIPS-approved algorithms, enabling a progressive switch to AES.

It is clear that 3DES is a powerful algorithm. 3DES may claim the same resistance to cryptanalysis as DEA since it is the fundamental cryptographic algorithm.

(a) Cryptography E K1 D K2 E K3 P A B C
(b) Decryption D K3 E K2 D K1 C B A P

Using the method that is allegedly used for DEA, 38 CHAPTER 2 / SYMMETRIC ENCRYPTION AND MESSAGE CONFIDENTIALITY is presented. Furthermore, brute-force assaults are almost unavoidable with a 168-bit key length. AES is eventually supposed to take the role of 3DES, although it will take some time. For the foreseeable future, NIST predicts that 3DES will continue to be a recognized algorithm for use by the US government. Two benefits of the Advanced Encryption Standard 3DES ensure its widespread adoption during the next several years. It first eliminates the DEA's susceptibility to a brute-force attack because to its 168-bit key length. Second, 3DES and DEA both use the same basic encryption method. No efficient cryptanalytic attack based on this method rather than brute force has been discovered despite it having received more attention than any other encryption technique over a longer period of time. In light of this, there is a lot of certainty that 3DES is exceedingly resistant to cryptanalysis. If security were the sole factor, 3DES would make a good option for a long time to come as a common encryption technique.

The main flaw with 3DES is how slow the software implementation of the algorithm is. The original DEA does not generate effective software code since it was created for hardware implementation in the middle of the 1970s. 3DES is slower since it has three times as many rounds as DEA. The use of a 64-bit block size by both DEA and 3DES is a supplementary drawback. A greater block size is preferred for reasons of efficiency and security. Due to these issues, 3DES is not a viable option for ongoing usage. NIST requested submissions in 1997 for a new Advanced Encryption Standard (AES) that would take its place and be much more efficient and have security strength on par with or better than 3DES. AES must be a symmetric block cipher with a block length of 128 bits and support for key lengths of 128, 192, and 256 bits, in addition to these basic criteria, according to NIST.

Security, computational effectiveness, memory needs, appropriateness of hardware and software, and adaptability were among the evaluation factors. The first round of review resulted in the acceptance of 15 suggested algorithms. Five algorithms remained after a second round of elimination. In November 2001, NIST issued a final standard (FIPS PUB 197) after completing its assessment process. Rijndael was chosen by NIST as the suggested AES algorithm. Dr. Joan Daemen and Dr. Vincent Rijmen, two cryptographers from Belgium, are the authors of the Rijndael algorithm and the AES submission.

AES employs a key length that may be 128, 192, or 256 bits long and a block length of 128 bits. This section's explanation makes use of a key length of 128 bits, which is perhaps the most typical implementation. A single 128-bit block serves as the input for the encryption and decryption methods. This block is shown as a square matrix of bytes in FIPS PUB 197. The State array, which is altered at each step of encrypting or decrypting, receives a copy of this block. State is transferred to an output matrix after the last phase. The 128-bit key is represented similarly as a square matrix of bytes. The key is then extended into a set of 44 words worth of key schedule words, with each word being four bytes long for a 128-bit key. As an example, the first column of the in matrix would be occupied by the first four bytes of a 128-bit plaintext input to the encryption cipher, the second column would be occupied by the second set of four bytes, and so on. Similar to this, the first column of the w matrix is taken up by the first four bytes of the enlarged key, which make up a word. The comments that follow provide some information on AES[9], [10].

1. The fact that this structure is not a Feistel structure is one notable aspect of it. Recall that the traditional Feistel structure involves the modification of one half of the data block, followed by the swapping of the two halves. AES analyzes the full data block in parallel throughout each round using substitutions and permutations rather than a Feistel structure as described.
2. An array of forty-four 32-bit words called w[i] is created from the input key. The round key for each round is made up of four different words (128 bits).
3. One permutation stage and three substitution stages total four distinct phases are used:
    a. Substitute bytes: The block is substituted byte-by-byte using a table called an S-box4.
    b. Shift rows: This straightforward permutation is carried out row by row.

c. Mix columns: A replacement that modifies each byte in a column based on how all of the bytes in the column are changed.
d. Add round key: A straightforward bitwise XOR of the present block and some of the enlarged key.

4. The cipher starts with an Add Round Key stage for both encryption and decryption, then proceeds through nine rounds that each include all four stages, followed by a tenth round with just three stages.
    a. The key is only used in the Add Round Key step. The cipher has an Add Round Key stage at both its start and finish because of this. Any additional step, whether applied at the start or the end, is reversible without the key and hence would not increase security.
    b. The Add Round Key stage wouldn't be overwhelming by itself. The remaining three phases combine to jumble the bits, but because they don't employ the key, they wouldn't provide any protection on their own. The cipher may be thought of as a series of operations that alternate between scrambling a block the other three phases, followed by XOR encryption, and so on. This plan is very effective and secure.
    c. It is simple to reverse each step. The decryption technique employs an inverse function for the phases of Substitute Byte, Shift Row, and Mix Columns. The converse is accomplished for the Add Round Key step by XORing the block with the same round key, utilizing the result that is A B B A.
5. Similar to the majority of block ciphers, the extended key is used in reverse order by the decryption algorithm. The decryption algorithm, however, differs from the encryption algorithm. This is a result of the unique way that AES is built.
6. It is simple to confirm that decryption indeed recovers the plaintext after it has been determined that all four steps are reversible. Encryption and decryption are seen moving in opposing vertical directions. State is same for both encryption and decryption at each horizontal position such as the dashed line in the illustration.
7. There are just three steps in the final encryption and decryption processes. This is necessary to make the cipher reversible and is also a result of the unique structure of AES.

REFERENCES:

[1] E. Omid Mahdi Ebadati, F. Eshghi, and A. Zamani, “Security enhancement of wireless sensor networks: A hybrid efficient encryption algorithm approach,” *J. Inf. Syst. Telecommun.*, 2018.

[2] M. E. Manaa and R. H. Jwdha, “A proactive data security scheme of files using Minhash technique,” *J. Theor. Appl. Inf. Technol.*, 2018.

[3] L. Babenko and I. Pisarev, “Distributed E-voting system based on blind intermediaries using homomorphic encryption,” in *ACM International Conference Proceeding Series*, 2018. doi: 10.1145/3264437.3264473.

[4] H. J. Fu, B. Cai, H. Xiang, and J. Sang, "Homomorphically encrypted arithmetic operations over symmetric ternary coding," *J. Cryptologic Res.*, 2018, doi: 10.13868/j.cnki.jcr.000237.

[5] C. Albin, D. Narayan, R. Varu, and V. Thanikaiselvan, "DWT Based Audio Encryption Scheme," in *Proceedings of the 2nd International Conference on Electronics, Communication and Aerospace Technology, ICECA 2018*, 2018. doi: 10.1109/ICECA.2018.8474602.

[6] D. Schürmann, G. von Zengen, M. Priedigkeit, S. Willenborg, and L. Wolf, "µDTNSec: a security layer with lightweight certificates for Disruption-Tolerant Networks on microcontrollers," *Ann. des Telecommun. Telecommun.*, 2018, doi: 10.1007/s12243-018-0655-2.

[7] M. K. Sharma and D. Somwanshi, "Improvement in Homomorphic Encryption Algorithm with Elliptic Curve Cryptography and OTP Technique," in *3rd International Conference and Workshops on Recent Advances and Innovations in Engineering, ICRAIE 2018*, 2018. doi: 10.1109/ICRAIE.2018.8710434.

[8] Divanshu, H. Singla, V. Sharma, B. Sharma, and P. Thakral, "A Quick Tour to Encryption Techniques," in *Proceedings of the 2nd International Conference on Trends in Electronics and Informatics, ICOEI 2018*, 2018. doi: 10.1109/ICOEI.2018.8553758.

[9] T. Belkhouja, X. Du, A. Mohamed, A. K. Al-Ali, and M. Guizani, "Symmetric encryption relying on chaotic henon system for secure hardware-friendly wireless communication of implantable medical systems," *J. Sens. Actuator Networks*, 2018, doi: 10.3390/jsan7020021.

[10] C. Guo, X. Fu, Y. Mao, G. Wu, F. Li, and T. Wu, "Multi-user searchable symmetric encryption with dynamic updates for cloud computing," *Inf.*, 2018, doi: 10.3390/info9100242.

CHAPTER 10

STREAM CIPHERS AND RC4

Mr. Aishwary Awasthi, Research Scholar
Department of Mechanical Engineering, Sanskriti University, Mathura, Uttar Pradesh, India
Email Id- aishwary@sanskriti.edu.in

A stream cipher is a type of encryption algorithm that operates on a stream of data, producing a stream of encrypted data that is used to mask the original data. Stream ciphers are typically used to encrypt data that is transmitted over a network or stored on a device. One of the most widely used stream ciphers is RC4, which was developed by Ron Rivest in 1987. RC4 is a symmetric key stream cipher, which means that the same key is used for both encryption and decryption. RC4 is a popular choice for encrypting wireless network traffic, as well as for encrypting web traffic using the SSL/TLS protocol. However, due to its susceptibility to some attacks, it has been largely replaced by other encryption algorithms such as AES.

In this article, we will explore the workings of stream ciphers and the RC4 algorithm in more detail. Stream ciphers work by generating a stream of pseudo-random bits, which are combined with the plaintext data using a bitwise exclusive-OR (XOR) operation. The resulting ciphertext is then transmitted or stored, along with the key used to generate the pseudo-random bit stream. To decrypt the data, the receiver uses the same key to generate the same pseudo-random bit stream, which is then XORed with the ciphertext to recover the original plaintext. The key used for the encryption process must be kept secret, as anyone who knows the key can easily decrypt the data. The security of a stream cipher depends on the randomness of the pseudo-random bit stream, as well as the strength of the encryption key[1], [2].

The security of a stream cipher can be enhanced by using a technique known as key whitening, which involves adding an extra layer of encryption to the key. This can make it more difficult for attackers to crack the encryption, even if they are able to intercept the encrypted data and the key. RC4 is a widely used stream cipher that is used to encrypt data in many different applications, including wireless network traffic, web traffic, and other types of data transmission. The algorithm uses a variable-length key, typically between 40 and 256 bits, to generate a pseudo-random bit stream.

The key is used to initialize a permutation of the numbers 0 through 255, which is then used to generate the pseudo-random bit stream. The permutation is generated by swapping values in an array based on the key. Once the permutation is generated, the algorithm begins generating the pseudo-random bit stream by performing a series of swaps on the values in the permutation. The resulting pseudo-random bit stream is then XORed with the plaintext data to produce the ciphertext.

The RC4 algorithm is simple and efficient, and can be implemented in hardware or software with relative ease. However, it has been shown to be susceptible to several attacks that can compromise the security of the encryption. One such attack is the Fluhrer-Mantin-Shamir (FMS) attack, which can be used to recover the key used to generate the pseudo-random bit stream with

a relatively small amount of encrypted data. Another attack is the related key attack, which can be used to recover the key when the attacker has access to both the plaintext and ciphertext generated using different keys. As a result of these vulnerabilities, RC4 is no longer considered a secure encryption algorithm for use in new applications. It has largely been replaced by more modern encryption algorithms such as AES, which are more resistant to attacks and offer better security. However, it is still used in some legacy applications, and understanding how it works can be useful for understanding the basics of stream ciphers and encryption algorithms in general[3], [4].

Stream ciphers are an important type of encryption algorithm that are widely used to secure data transmissions over networks and other communication channels. The RC4 algorithm is a popular stream cipher that was widely Stream ciphers are a type of encryption algorithm that work by generating a stream of pseudo-random bits that are combined with the plaintext data using a bitwise XOR operation to produce the ciphertext. The same key is used for both encryption and decryption in a stream cipher. The key must be kept secret to ensure the security of the encryption. The strength of a stream cipher depends on the randomness of the pseudo-random bit stream and the strength of the key used to generate the bit stream. The security of a stream cipher can be enhanced by using a technique known as key whitening, which adds an extra layer of encryption to the key.

Stream ciphers are widely used in many different applications, including wireless network traffic, web traffic, and other types of data transmission. One of the most widely used stream ciphers is RC4 is a symmetric key stream cipher that was developed by Ron Rivest in 1987. The algorithm uses a variable-length key, typically between 40 and 256 bits, to generate a pseudo-random bit stream. The key is used to initialize a permutation of the numbers 0 through 255, which is then used to generate the pseudo-random bit stream.

Once the permutation is generated, the algorithm begins generating the pseudo-random bit stream by performing a series of swaps on the values in the permutation. The resulting pseudo-random bit stream is then XORed with the plaintext data to produce the ciphertext. RC4 is a simple and efficient encryption algorithm that can be implemented in hardware or software with relative ease. However, it has been shown to be susceptible to several attacks that can compromise the security of the encryption. One such attack is the Fluhrer-Mantin-Shamir (FMS) attack, which can be used to recover the key used to generate the pseudo-random bit stream with a relatively small amount of encrypted data. Another attack is the related key attack, which can be used to recover the key when the attacker has access to both the plaintext and ciphertext generated using different keys.

As a result of these vulnerabilities, RC4 is no longer considered a secure encryption algorithm for use in new applications. It has largely been replaced by more modern encryption algorithms such as AES, which are more resistant to attacks and offer better security. However, it is still used in some legacy applications, and understanding how it works can be useful for understanding the basics of stream ciphers and encryption algorithms in general.

There are several important considerations when using stream ciphers, including key management, random number generation, and the selection of appropriate encryption algorithms. Key management is a critical aspect of stream cipher security, as the strength of the encryption depends on the strength of the key used to generate the pseudo-random bit stream.

Random number generation is also an important consideration when using stream ciphers, as the quality of the random numbers used to generate the pseudo-random bit stream can impact the security of the encryption. It is important to use a high-quality random number generator to ensure the security of the encryption.

The selection of appropriate encryption algorithms is also important when using stream ciphers. While RC4 was once a popular choice for many applications, it is no longer considered secure due to its vulnerabilities to certain attacks. More modern encryption algorithms such as AES are now widely used for many applications, as they offer better security and are more resistant to attacks.

In summary, stream ciphers are an important type of encryption algorithm that are widely used to secure data transmissions over networks and other communication channels. RC4 is a popular stream cipher that was widely used in the past but is now considered insecure due to its vulnerabilities to certain attacks. While stream ciphers can be an effective way to secure data transmissions, it is important to carefully manage keys, use high-quality random number generators, and select appropriate encryption algorithms to ensure the security of the encryption.

Stream ciphers are often used in applications where data must be encrypted and transmitted quickly and efficiently, such as in wireless networks, voice-over-IP (VoIP) communications, and secure web browsing. Unlike block ciphers, which encrypt fixed-size blocks of data, stream ciphers encrypt data one bit or byte at a time, making them more efficient for encrypting data in real-time applications.

Stream ciphers can be designed to operate in a synchronous or self-synchronizing mode. In synchronous mode, the same key is used to generate the pseudo-random bit stream for each block of data, and the encryption and decryption processes are tightly synchronized. In self-synchronizing mode, the key and the state of the cipher are updated after each bit or byte of data is encrypted, allowing the cipher to recover from transmission errors or lost packets in real-time communications.

One of the primary benefits of stream ciphers is their speed and efficiency. Because stream ciphers can encrypt data one bit or byte at a time, they can be implemented in hardware or software with relatively little overhead, making them ideal for use in applications where speed and efficiency are important. Another advantage of stream ciphers is their flexibility. Stream ciphers can be designed to operate with different key sizes and pseudo-random bit stream lengths, allowing them to be tailored to the specific needs of an application. This makes stream ciphers a versatile tool for securing a wide range of applications and systems[5], [6].

However, stream ciphers also have some disadvantages and limitations. One of the primary concerns with stream ciphers is their susceptibility to certain attacks, such as the FMS attack and

related key attack, which can compromise the security of the encryption. As a result, it is important to carefully select and implement stream ciphers that have been designed to resist these types of attacks.

Another challenge with stream ciphers is the need for high-quality random number generation. Because stream ciphers rely on the generation of a pseudo-random bit stream, the quality of the random numbers used to generate the bit stream can impact the security of the encryption. It is important to use a high-quality random number generator to ensure the security of the encryption.

Despite these challenges, stream ciphers remain an important tool for securing real-time communications and other applications where speed and efficiency are important. Many modern encryption algorithms, such as AES, are designed to operate as block ciphers, but can also be used in a stream cipher mode. This allows the benefits of both block and stream ciphers to be leveraged, depending on the specific needs of an application.

In conclusion, stream ciphers are an important type of encryption algorithm that are widely used in a variety of applications. They offer speed and efficiency, as well as flexibility and versatility, making them an ideal tool for securing real-time communications and other applications where speed and efficiency are important[7]. However, stream ciphers also have some disadvantages and limitations, and it is important to carefully select and implement stream ciphers that have been designed to resist attacks and ensure the security of the encryption.

A block cipher creates an output block for every input block it analyzes, one block of elements at a time. A stream cipher continually processes the input items while generating the output one element at a time. Although Despite the fact that block ciphers are far more prevalent, there are certain situations where a stream cipher is preferable. Examples are provided later in this book. In this part, we examine RC4, which is perhaps the most well-known symmetric stream cipher[8], [9].

Although a stream cipher may be configured to work on one bit at a time or on units bigger than a byte at a time, they typically encrypt plaintext one byte at a time. A typical illustration of the construction of stream ciphers is shown in Figure 2.8. In this structure, a key is used as input to a pseudorandom bit generator, which outputs a stream of seemingly random 8-bit values.A pseudorandom stream has a seemingly random character but is unpredictable without knowing the input key. The bitwise exclusive-OR (XOR) operation is used to combine the keystream, which is the generator's output, with the plaintext stream one byte at a time.

1. A long time should be present in the encryption sequence. A function that generates a predictable stream of bits that ultimately repeats is used in pseudorandom number generators. The longer the repetition time, the more challenging cryptanalysis will be.
2. The keystream should closely resemble the characteristics of a real random number stream. For instance, there should be about equal amounts of 1s and 0s. All 256 potential byte values should occur almost equally regularly if the keystream is viewed as a stream of bytes. The

ciphertext is more randomized and more difficult to decrypt the more random the keystream seems to be.

3. Take note of the fact that the output of the pseudorandom number generator is dependent on the key value entered as input. The key must be long enough to prevent brute-force assaults. Here, the same factors that affect block ciphers also apply. Therefore, a key length of at least 128 bits is preferred in light of current technology.

A stream cipher of equivalent key length may be as safe as a block cipher with a well-designed pseudorandom number generator. The main benefit of a stream cipher is that it nearly always operates quicker and with far less code than block ciphers.

Just a few lines of code are required to implement the RC4 example from this section compares the execution timings of RC4 with those of three well-known symmetric block ciphers using information. You may reuse keys using a block cipher, which is a benefit. However, cryptanalysis is often relatively easy when two plaintexts are encrypted with the same key using a stream cipher the outcome of XORing the two ciphertext streams together is the XOR of the original plaintexts. Cryptanalysis may be successful if the plaintexts are text strings, credit card numbers, or other byte streams with well-known features.

A stream cipher could be a preferable option for applications that need to encrypt or decode a stream of data such as through a data-communications channel or a browser/Web connection). Block ciphers could be better suitable for applications that deal with blocks of data (such as file transfer, email, and databases.

Ron Rivest created the stream cipher RC4 for RSA Security in 1987. It is a stream cipher with byte-oriented operations and changeable key sizes. A random permutation is used as the foundation of the method. According to analysis, the period of the cipher is very certainly more than 10100. Each output byte requires eight to sixteen computer operations, and the cipher should operate relatively rapidly in software.

The authors show that a specific attack strategy may compromise the WEP protocol, which is designed to offer anonymity on 802.11 wireless LAN networks. In essence, the method through which keys are created for use as input to RC4 is the issue rather than RC4 itself. This specific issue can be fixed in WEP by altering the method keys are created, and it does not seem to be applicable to other programs that use RC4. This issue highlights the challenge of creating a safe system that incorporates both cryptographic functions and the protocols that utilize them.

One block of data is processed at a time by a symmetric block cipher. The block length for DES and 3DES is 64 bits, whereas the block length for AES is 128 bits. Breaking the plaintext into b-bit blocks is important for larger plaintext quantities (padding the last block if necessary). NIST (Special Publication 800-38A) has outlined five modes of operation to use a block cipher in a range of applications. The five modes are intended to cover almost all of the encryption-related uses that a block cipher may be put to. These modes are designed to be used with triple DES and AES, as well as other symmetric block ciphers. In the remaining portion of this section, the most significant modes are briefly outlined.

The easiest method is to use the electronic codebook (ECB) mode, in which each block of plaintext is encrypted with the same key and is treated b bits at a time. The word "codebook" refers to the fact that every b-bit block of plaintext has a specific ciphertext for a certain key. As a result, one may picture a massive codebook with entries for each and every conceivable b-bit plaintext pattern that display the appropriate ciphertext.

With ECB, the same ciphertext is always produced if the same b-bit block of plaintext occurs more than once in the message. The ECB mode may not be secure for extended messages as a result. It could be feasible for a cryptanalyst to take advantage of these regularities if the message is substantially organized. The cryptanalyst may have a number of known plaintext-ciphertext pairings to work with, for instance, if it is known that the message always begins with a certain set of predetermined fields. The analyst may recognize these components if the message contains repeating elements with a duration of repetition that is a multiple of b bits. This could facilitate analysis or provide a chance to swap out or rearrange blocks.

We would want a method where the same plaintext block, when repeated, generates distinct ciphertext blocks in order to get around ECB's security flaws. The input to the encryption method in the cipher block chaining (CBC) mode is the XOR of the current plaintext block and the previous ciphertext block; the same key is used for each block. In essence, we have chained together the plaintext block sequence's processing [10], [11].

Software productivity: Similar to this, processors that provide parallel features (such aggressive pipelining, multiple instruction dispatch per clock cycle, a high number of registers, and SIMD instructions) may be used successfully due to the potential for parallel execution in CTR mode.

Preprocessing: The input of the plaintext or ciphertext is not necessary for the execution of the underlying encryption technique. Therefore, the output of the encryption boxes that feed into the XOR functions may be prepared using preprocessing provided enough memory is available and security is maintained. The sole calculation is a sequence of XORs when either plaintext or ciphertext input is given. Throughput is considerably increased by such a method.

CTR can be demonstrated to be at least as secure as the other modes covered in this section. Ease of use: Unlike ECB and CBC modes, CTR mode just needs the encryption algorithm to be implemented, not the decryption method. This is particularly important when the encryption and decryption algorithms diverge significantly, as they do with AES. Additionally, it's not necessary to handle the scheduling of the decryption key. The issues in this chapter are explored in further depth is a crucial reference work for cryptographic algorithm coverage; it includes descriptions of almost every cryptographic algorithm and protocol released up to the time of the book's authoring is a further insightful survey provides a more thorough analysis with substantial mathematical argumentation.

REFERENCES:

[1] F. Akbar, H. Mawengkang, and S. Efendi, “Comparative analysis of RC4+ algorithm, RC4 NGG algorithm and RC4 GGHN algorithm on image file security,” in *IOP Conference Series: Materials Science and Engineering*, 2018. doi: 10.1088/1757-899X/420/1/012131.

[2] A. Jana and G. Paul, "Revisiting RC4 key collision: Faster search algorithm and new 22-byte colliding key pairs," *Cryptogr. Commun.*, 2018, doi: 10.1007/s12095-017-0231-z.

[3] M. Kumari and S. Gupta, "A Novel Image Encryption Scheme Based on Intertwining Chaotic Maps and RC4 Stream Cipher," *3D Res.*, 2018, doi: 10.1007/s13319-018-0162-2.

[4] R. Rifki, A. Septiarini, and H. R. Hatta, "Cryptography using random RC4 stream cipher on SMS for android-based smartphones," *Int. J. Adv. Comput. Sci. Appl.*, 2018, doi: 10.14569/IJACSA.2018.091214.

[5] R. B. Gandara, "Analisa Enkripsi pada Protokol IEEE 802.15.4 dengan Algoritma Rabbit untuk Aplikasi Industri Sektor Migas," *J. Telekomun. dan Komput.*, 2018, doi: 10.22441/incomtech.v8i3.5674.

[6] M. A. Budiman, Amalia, and N. I. Chayanie, "An Implementation of RC4+ Algorithm and Zig-zag Algorithm in a Super Encryption Scheme for Text Security," in *Journal of Physics: Conference Series*, 2018. doi: 10.1088/1742-6596/978/1/012086.

[7] Y. He, M. Zhang, X. Yang, J. Luo, and Y. Chen, "A survey of privacy protection and network security in user on-demand anonymous communication," *IEEE Access*, 2020, doi: 10.1109/ACCESS.2020.2981517.

[8] K. G. Paterson and J. C. N. Schuldt, "Statistical attacks on cookie masking for RC4," *Cryptogr. Commun.*, 2018, doi: 10.1007/s12095-018-0280-y.

[9] X. Hu and Y. Zhao, "One to one identification of cryptosystem using fisher's discriminant analysis," *Int. J. Networked Distrib. Comput.*, 2018, doi: 10.2991/ijndc.2018.6.3.4.

[10] X. Hu and Y. Zhao, "One to one identification of cryptosystem using fisher's discriminant analysis," in *ACM International Conference Proceeding Series*, 2018. doi: 10.2991/ijndc.2018.3.6.4.

[11] R. B. Gandara, "dengan Algoritma Rabbit untuk Aplikasi Industri Sektor Migas," *IncomTech, J. Telekomun. dan Komput.*, 2018.

CHAPTER 11

APPROACHES TO MESSAGE AUTHENTICATION

Dr. Devendra Singh, Assistant Professor
Department of Computer Science and Engineering, Sanskriti University, Mathura, Uttar Pradesh, India
Email Id- devendras.soeit@sanskri.edu.in

Message authentication is the process of verifying the integrity of a message to ensure that it has not been tampered with or altered in any way. This is an essential aspect of cybersecurity, as it helps to protect sensitive information from being intercepted, modified, or stolen by unauthorized parties. There are several approaches to message authentication, including the following:

1. **Message Authentication Codes (MACs):** A MAC is a cryptographic function that generates a fixed-length value based on a message and a secret key. The MAC can be used to verify the authenticity and integrity of the message, as any tampering with the message will result in a different MAC value. MACs are widely used in network protocols and data encryption systems.
2. **Digital Signatures:** A digital signature is a mathematical scheme used to validate the authenticity and integrity of a digital message or document. It involves the use of a private key to encrypt a message or document, and a public key to decrypt and verify the signature. Digital signatures are commonly used in electronic transactions, such as online banking and e-commerce.
3. **Hash Functions:** A hash function is a mathematical algorithm that takes an input message and generates a fixed-length output called a hash value. The hash value can be used to verify the integrity of the message, as any changes to the message will result in a different hash value. Hash functions are commonly used in digital forensics and data verification.
4. **Public Key Infrastructure (PKI):** PKI is a system of digital certificates, public key encryption, and other cryptographic technologies that enable secure communication over the internet. It involves the use of a trusted third party, known as a Certificate Authority (CA), to issue digital certificates that validate the identity of individuals, organizations, or devices.
5. **Time Stamping:** Time stamping involves the use of a trusted third party to add a digital timestamp to a message, indicating the time of its creation or transmission. This can be used to provide evidence of the authenticity and integrity of the message, as any tampering with the message will result in a different timestamp. Time stamping is commonly used in legal and financial transactions.

These are some of the approaches to message authentication in cybersecurity. It is important to select the appropriate approach based on the specific requirements and constraints of the application[1], [2].

Message authentication is a critical aspect of cybersecurity, as it helps to ensure that digital messages are not tampered with or altered in any way. There are many different approaches to message authentication, each with its own strengths and weaknesses. In this essay, we will explore these approaches in more detail, including the algorithms used, their advantages, and their limitations.

Message Authentication Codes (MACs) are one of the most commonly used methods of message authentication. They are essentially a cryptographic function that takes a message and a secret key, and generates a fixed-length value that can be used to verify the integrity and authenticity of the message. This value is known as the MAC. MACs can be used in a variety of contexts, including network protocols, data encryption systems, and digital signatures. They are widely used because they are relatively simple and efficient, and can provide strong authentication and integrity protection.

There are many different types of MACs, including Hash-based MACs (HMACs) and CBC-MACs. HMACs are based on a hash function, such as SHA-256 or SHA-3, and use a secret key to generate the MAC value. CBC-MACs use a block cipher, such as AES or DES, in Cipher Block Chaining (CBC) mode to generate the MAC value. The main advantage of MACs is that they provide strong message authentication and integrity protection, and are relatively efficient. However, they require a shared secret key between the sender and the recipient, which can be difficult to manage in certain contexts. Additionally, they do not provide any confidentiality protection, which means that the message contents are still vulnerable to eavesdropping.

Digital signatures are another commonly used approach to message authentication. They are essentially a mathematical scheme that is used to validate the authenticity and integrity of a digital message or document. A digital signature is created by taking a message or document, and using a private key to encrypt it. The resulting encrypted message is the digital signature.

To verify the signature, the recipient uses the public key of the sender to decrypt the signature. If the decrypted signature matches the original message, then the signature is considered valid, and the authenticity and integrity of the message are confirmed. Digital signatures are widely used in electronic transactions, such as online banking and e-commerce, as they provide strong authentication and integrity protection. They are also relatively efficient, and do not require any shared secret keys between the sender and the recipient.

One of the main limitations of digital signatures is that they are not anonymous, as they are tied to the identity of the sender. Additionally, they do not provide any confidentiality protection, which means that the message contents are still vulnerable to eavesdropping. Hash functions are another approach to message authentication. They are essentially a mathematical algorithm that takes an input message and generates a fixed-length output called a hash value. The hash value can be used to verify the integrity of the message, as any changes to the message will result in a different hash value.

Hash functions are commonly used in digital forensics and data verification, as they provide strong integrity protection. They are also relatively efficient, and can be used in a variety of contexts. There are many different types of hash functions, including MD5, SHA-1, SHA-2, and SHA-3. The choice of hash function depends on the specific requirements and constraints of the application[3], [4].

The main advantage of hash functions is that they provide strong message integrity protection, and are relatively efficient. However, they do not provide any authentication or confidentiality protection, which means that the message contents are still vulnerable to eavesdropping and tampering and digital signatures that is used to provide secure communication over a network. PKI is based on a hierarchical system of trust, where digital certificates are used to verify the identity of users and devices.

In a PKI system, each user or device has a public key and a private key. The public key is used to encrypt messages and verify digital signatures, while the private key is used to decrypt messages and create digital signatures. Digital certificates are used to link a user or device's public key to their identity, and are issued by a trusted third party called a Certificate Authority (CA). PKI systems are widely used in network security, including secure email, e-commerce, and online banking. They provide strong authentication, confidentiality, and integrity protection, and are relatively efficient.

One of the main advantages of PKI is that it provides strong authentication and confidentiality protection, and can be used in a variety of contexts. However, it requires the use of digital certificates, which can be difficult to manage and distribute in certain contexts. Additionally, PKI systems are vulnerable to attacks, such as man-in-the-middle attacks, where an attacker intercepts and modifies communication between two parties. Zero-Knowledge Proofs (ZKPs) are a relatively new approach to message authentication that is based on complex mathematical algorithms. ZKPs are used to prove the authenticity and integrity of a message without revealing any of its contents.

ZKPs work by creating a challenge and response protocol, where the sender proves that they know a secret value without revealing it to the recipient. This is done by using a series of mathematical operations that are computationally difficult to reverse. ZKPs are currently being used in blockchain technology, where they are used to verify the authenticity and integrity of transactions without revealing any of the transaction details. They are also being used in privacy-preserving authentication systems, where they can be used to authenticate users without revealing any personal information.

One of the main advantages of ZKPs is that they provide strong authentication and confidentiality protection, while also preserving privacy. However, they are currently limited in their use, as they are relatively new and require complex mathematical algorithms to implement. There are many different approaches to message authentication, each with its own strengths and weaknesses. MACs, digital signatures, hash functions, PKI, and ZKPs are all widely used in network security, and provide strong authentication and integrity protection.

The choice of message authentication approach depends on the specific requirements and constraints of the application. For example, if confidentiality is a concern, then digital signatures or PKI may be a better choice. If privacy is a concern, then ZKPs may be a better choice. Regardless of the approach used, it is important to implement message authentication measures in order to ensure the security and integrity of digital messages. Encryption guards against unarmed assault (eavesdropping). Protecting against active assault is a distinct necessity (falsification of data and transactions). Message authentication is a defense against such assaults[4], [5].

When a communication, file, document, or other collection of data is real and originates from its claimed source, it is said to be authentic. The process of message authentication enables parties involved in communication to confirm the validity of communications they have received. Verifying that the message's substance hasn't been changed and that the source is reliable are the two most crucial factors. We could also want to confirm a message's order in relation to other messages flowing between two parties and timeliness to make sure it hasn't been maliciously delayed and replayed.

Conventional Encryption for Authentication

It would seem that symmetric encryption alone may be used to carry out authentication. Only the legitimate sender would be able to encrypt a message if we presume that only the sender and recipient share a key which is how it should be.

For the sake of convenience, we'll refer to message authentication for the rest of this chapter. This refers to both message transmission and data storage authentication (data authentication) if the recipient is able to identify a legitimate message. Additionally, the receiver is certain that no modifications have been performed and that sequencing is correct if the message contains an error-detection code and a sequence number. The receiver is certain that the message has not been delayed beyond what is typically anticipated for network transit if the message additionally contains a timestamp.

Symmetric encryption by itself is really an inadequate instrument for data authentication. For instance, if an attacker rearranges the ciphertext blocks in the ECB form of encryption, each block will still correctly decode. The overall meaning of the data sequence might change as a result of the rearranging. Even while sequence numbers could be utilized at some level (like with each IP packet), it is uncommon for each b-bit block of plaintext to have its own unique sequence number. Block reordering is a danger as a result.

In this part, we look at a number of non-encrypted methods for message authentication. An authentication tag is created and added to each message for transmission in each of these methods. The message itself is not encrypted and is readable at the destination regardless of the destination's authentication process. The message secrecy is not ensured by the methods outlined in this section since the message is not encrypted. Message encryption by itself does not provide a secure method of authentication, as was already explained. However, by encrypting a communication together with its authentication tag, it is feasible to combine authentication and secrecy in a single technique. Message authentication is often offered separately from message

encryption, however. Three scenarios are provided by [DAVI89] where message authentication without secrecy is preferable:

8. The same message may be sent to several locations in a variety of applications. Users being informed that the network is now inaccessible and a control center receiving an alert are two instances. It is less expensive and more dependable to have only one location in charge of ensuring authenticity. As a result, the message must be sent in plaintext along with a message authentication tag. The relevant system carries out authentication. A global alarm is sounded in the event of a violation, alerting the other destination systems.
9. Another situation that could occur in an exchange is when one party is overburdened and lacks the time to decode every message that comes in. Selective authentication is used, with messages picked at random for verification.
10. Plaintext computer program authentication is a useful service[6], [7].

Without having to repeatedly decrypt it, which would be a waste of CPU resources, the computer program may be run. However, if the program had a message authentication tag, it could be examined anytime confirmation of the program's integrity was needed.

AUTHENTICATION CODE FOR MESSAGE

A message authentication code (MAC), a brief piece of data that is attached to the message as part of one authentication method, is created using a secret key. This method implies that A and B, two communication parties, share a secret key called KAB. The message authentication code is calculated by A when it has a message to deliver to B based on the message and the key: MACM F. (KAB, M). The targeted recipient receives both the message and the code. To create a new message authentication code, the receiver applies the same computation to the received message using the same secret key.

The following assertions are true if the received code matches the computed code and we assume that only the sender and receiver are aware of the identity of the secret key:

1. The message's integrity is guaranteed to the recipient. The receiver's estimate of the code will vary from the received code if the attacker modifies the message but leaves the code alone. The attacker cannot change the code to match the changes in the message since it is presumed that they do not possess the secret key.
2. The message's authenticity is confirmed to the recipient.No one else could create a message with an appropriate code since no one else is aware of the secret key.
3. The receiver may be guaranteed that the message is in the right order if it contains a sequence number (like those used with HDLC and TCP), since an attacker cannot effectively change them.

The FIPS PUB 113 standard from the NIST advises using DES. The last few bits of the ciphertext are used as the code to create an encrypted version of the message using DES. Commonly, a 16- or 32-bit code is used. The just-discussed procedure is comparable to encryption. One distinction is that, unlike the decryption method, the authentication algorithm

does not have to be reversible. The mathematical characteristics of the authentication function make it less breakable than encryption.

The one-way hash function is an alternative to the message authentication code. A hash function receives a variable-size message M and outputs a fixed-size message digest H(M), much as the message authentication code does. A hash function does not need a secret key as input, unlike the MAC. The message digest is provided together with the message in a manner that ensures its authenticity in order to authenticate the communication.

If it is assumed that only the sender and recipient have the encryption key, the message digest may be encrypted using traditional encryption (part a), ensuring authenticity. Public-key encryption (part b) may be used to encrypt the message digest; this is covered in Section 3.5. The public-key strategy has two benefits: (1) It offers both message authentication and a digital signature. (2) It does not demand that communication parties be given access to keys.

The fact that these two methods need less processing than methods that encrypt the whole message is another benefit over those methods. However, there has been interest in creating a method that completely eliminates encryption. In [TSUD92], it is noted that there are many causes for this interest:

a. **Software for encryption operates slowly:** Even though each transmission only requires a minimal amount of data to be encrypted, a system may constantly be receiving and sending messages.
b. **Hardware expenses for encryption are quite low:** There are affordable chip implementations of DES, but the price rises if every node in a network has to have this functionality.
c. **Hardware for encryption is designed for huge data sizes:** An encryption technique may be patented; for tiny blocks of data, a large percentage of the time is consumed by initialization/invocation overhead.

A method for message authentication that employs a hash function but no encryption. This method presupposes that A and B, two communication parties, share the secret value SAB. A computes the hash function over the concatenation of the secret value and the message when it has a message to transmit to B: MDM H (SAB7M). After that, [M7MDM] is sent to B. B can recompute H(SAB7M) and validate MDM since it has SAB. Since the secret value itself suggests concatenation, it is = 2 7 in value.

A communication that has been intercepted cannot be altered by an attacker using a One-Way Hash Function that has not been transmitted. Additionally, an attacker cannot produce a fake message as long as the secret value is kept a secret. The IP security protocol (covered in Chapter 8) uses a derivative of the third approach called HMAC; it has also been standardized for SNMPv3 (Chapter 12).

SECURE HASH FUNCTIONS

The one-way hash function, often known as the secure hash function, is crucial for both digital signatures and message authentication. We start off by talking about the specifications for a

secure hash function in this section. The most significant hash function, SHA, is then examined. To create a "fingerprint" of a file, communication, or other block of data, a hash function is used.

A hash function H has to possess the following characteristics in order to be helpful for message authentication:

1. H may be used on a data block of any size.
2. H generates an output of a set length.
3. For every given x, H(x) is comparatively simple to calculate, making hardware and software implementations feasible.
4. It is computationally impossible to obtain x such that H(x) h for any given code. One-way or preimage resistant is the term used to describe a hash function having this characteristic.
5. Finding y x for each given block x with H(y) H is computationally impossible (x). Second preimage resistant is a characteristic that describes a hash function. This is also known as poor collision resistance.
6. Finding any pair (x, y) such that H(x) H is computationally impossible (y).

Collision-resistant hash functions have this characteristic. This is also known as being very collision resistant. For a hash function to be used effectively for message authentication, the first three qualities must be present. Preimage resistant, the fourth property, is a "one-way" quality: Giving a message makes it simple to produce a code, while giving a code makes it very hard to generate a message. If the authentication method uses a secret value, this feature is crucial. The secret value is not provided in its whole, but if the hash function is not one-way, an attacker may quickly learn what it is: The message M and the hash code C H are obtained by the attacker if they are able to watch or intercept a transmission (SAB7M). After that, the attacker flips the hash algorithm to get SAB 7 M H 1. Now that the attacker has both M and SAB7M, recovering SAB is simple. The second preimage resistance attribute ensures that no message with the identical hash value as a particular message may be found elsewhere.

When a hash code is encrypted, this prevents forgery. The following series of actions would be possible for an attacker if this attribute weren't true: Observe or intercept a message together with its encrypted hash code first. Then, using the message, create an unencrypted hash code. Finally, create a different message with the same hash code. X is referred to as a preimage of y for f(x) y. There could be more than one preimage value for a given y unless f is one-to-one. Weak hash function refers to a hash function that meets the first five criteria in the list above. If the sixth criteria is also met, then it is \sreferred to as a strong hash function. The sixth characteristic, collision resistance, guards against the sophisticated birthday assault type of attack. The scope of this text does not extend to the specifics of this assault. The attack decreases an m-bit hash function's strength from 2m to 2m/2.

A message digest not only offers authentication but also data integrity. Similar to a frame check sequence, it does the following: The message digest will be incorrect if any bits in the message are unintentionally changed while in transit. The two methods of attacking a secure hash function are cryptanalysis and brute-force assault, much like symmetric encryption.

Similar to symmetric encryption schemes, a hash function's cryptanalysis entails finding logical loopholes in the algorithm. The length of the hash code generated by the algorithm alone determines how strong a hash function is against brute-force assaults. The amount of work needed is proportional to the following for a hash code of length n:

If collision resistance is necessary (and this is desired for a general-purpose secure hash code), the strength of the hash code against brute-force assaults is determined by the number 2n/2. For MD5, which has a 128-bit hash length, Van Oorschot and Wiener [VANO94] offered a concept for a $10 million collision search engine that could identify a collision in 24 days. Therefore, a 128-bit coding may be considered insufficient. If a hash code is thought of as a series of 32 bits, then a 160-bit hash length is the next level above. The same search engine would take more than 4,000 years to locate a collision with a 160-bit hash length. The time would be substantially shorter with current technology, making the 160 bits suspicious.

The following fundamental ideas underlie the operation of all hash functions. The input is seen as a series of n-bit blocks, including messages, files, etc. An n-bit hash function is created by iteratively processing each block of the input. The bit-by-bit exclusive-OR (XOR) of each block is one of the most basic hash functions. Ci bi1 bi2... bim may be used to represent this, where Ci ith bit of the hash code, 1 I n m number of n-bit input blocks, and bij ith bit in jth block = XOR operation. resistive to preimage 2n. This technique, which creates a straightforward parity for each bit location and is known as a longitudinal redundancy check. It works well as a data integrity check for random data. The likelihood that each n-bit hash value being equal. Therefore, the likelihood that a data mistake will leave the hash value unaltered is 2 n.

The function performs worse with data that is more consistently formatted. For instance, the high-order bit of each octet in most regular text files is consistently zero. Therefore, if a 128-bit hash value is employed, the hash function on this kind of data has an effectiveness of 2 112, rather than 2 128. One easy method to make things better is to rotate or shift the hash value by one bit after each block is processed. This has the effect of more thoroughly "randomizing" the input and eradicating any regularities that may be there.

Even while the second method offers a reasonable level of data integrity, it is essentially worthless for data security when a plaintext message and an encrypted hash code are combined, as shown in Figures 3.2a and b. It is simple to create a new message that generates that hash code given a message: Prepare the alternative message you want, and then add an n-bit block to it. This causes the combination of the new message and block to produce the hash code you want.

Despite the fact that a basic XOR or rotated XOR (RXOR) is inadequate if just the hash code is encrypted, you could still believe that such a simple function might be helpful when the message and hash code are both encrypted. But caution is advised. The National Bureau of Standards first suggested a method that included applying a basic XOR to 64-bit blocks of the message and then encrypting the whole message using the cipher block chaining (CBC) mode[8], [9].

Secure Hash Function (SHA)

The Secure Hash Algorithm has become the most used hash function in recent years (SHA). By 2005, SHA was essentially the only standardized hash algorithm still in use since almost every other commonly used hash function had been shown to have significant cryptanalytic flaws. The National Institute of Standards and Technology (NIST) created SHA, which was then released as FIPS 180, a federal information processing standard, in 1993. After SHA (now referred to as SHA-0flaws)'s were found, a corrected version, known as SHA-1, was released as FIPS 180-1 in 1995. "Secure Hash Standard" is the name of the real standards paper. Based on the hash algorithm MD4, SHA's structure is very similar to that of MD4. RFC 3174, which largely repeats the content of FIPS 180-1 but adds a C code implementation, also specifies SHA-1.

Using addition modulo 264, each of the eight words in the buffer is individually added to each of its corresponding words in Hi 1. Every bit of the hash code generated by the SHA-512 algorithm is a function of every bit of the input. It is improbable that two messages picked at random, even if they display comparable regularities, would have the same hash code due to the complicated repeating of the fundamental function F.

The difficulty of discovering two messages with the same message digest is on the order of 2256 operations, whereas the difficulty of finding a message with a given digest is on the order of 2512 operations, unless there is some hidden flaw in SHA-512 that has not yet been disclosed. An surge in interest in creating a MAC generated from a cryptographic hash algorithm, such SHA-1, has been seen in recent years.

The reasons for this interest include the availability of library code for cryptographic hash functions and the fact that cryptographic hash functions often run more quickly in software than traditional encryption methods like DES. A hash function such as SHA-1 was not intended for use as a MAC and \scannot be used directly for that purpose since it does not depend on a secret key. There have been many suggestions for adding a secret key to an existing hash algorithm. The strategy with the greatest backing is HMAC [BELL96a, BELL96b]. HMAC is used in IP Security, Transport Layer Security (TLS), Secure Electronic Transactions, and other Internet protocols. It was published as RFC 2104 and was selected as the MAC for IP Security that must be implemented (SET).

DESIGN OBJECTIVES FOR HMAC

The following design goals are listed in RFC 2104 for HMAC.

To utilize the available hash functions unaltered. In particular, hash functions that function well in software and whose source code is freely and widely available. They should also:

11. Allow for easy replacement of the embedded hash function in case faster or more secure hash functions are found or required.
12. Preserve the original performance of the hash function without significantly degrading it.
13. Use and handle keys simply.

For HMAC to be accepted, the first two goals must be met. The hash function is treated by HMAC as a "black box." This offers two advantages. First, HMAC may be implemented by using a module from an existing implementation of a hash function. As a result, the majority of the HMAC code is already packed and prepared for usage. Second, all that is necessary to swap out a specific hash function in an HMAC implementation is to uninstall the old module and replace it with the new one. If a quicker hash function was needed, this might be accomplished. More importantly, if the embedded hash function's security were to be compromised, the security of HMAC could still be maintained by simply substituting a more secure hash function.

The fundamental benefit of HMAC over other suggested hash-based systems is, in fact, the final design purpose on the list above. If the integrated hash function has some respectable cryptographic capabilities, HMAC may be shown to be safe. Later on in this section, we will return to this idea, but first, let's look at HMAC's structure. Keep in mind that the XOR on the iPad flips half of the bits in K. Similar to the XOR, the XOR with opad flips one-half of K's bits, but with a new set of bits. In essence, we have produced two keys from K by running Si and So through the hash algorithm. For lengthy communications, HMAC should run roughly at the same speed as the inbuilt hash function. HMAC increases the number of basic hash function runs by three (for Si, So, and the block produced from the inner hash). In this part, we examine a number of block cipher-based MACs.

A message authentication code based on ciphers (CMAC) AES and triple DES are compatible with the Cipher-based Message Authentication Code (CMAC) mode of operation. NIST Special Publication 800-38B has details on it. Let's start by thinking about CMAC's functionality when the message is an integer multiple of n of the encryption block length b. b 128 for AES and b 64 for triple DES.

There are n blocks in the message (M1, M2, . . ., Mn). The technique uses an n-bit key, K1, and a k-bit encryption key, K. The key size for triple DES is 112 or 168 bits, whereas the key size for AES is 128, 192, or 256 bits.

C1 E(K, M1) (K, M1)

C2 E(K, [M2 C1]) (K, [M2 C1])

C3 E(K, [M3 C2]) (K, [M3 C2])

Cn E(K, [MN Cn 1 K1]) (K, [MN Cn 1 K1])

T message authentication code, commonly known as the tag Tlen bit length of T MSBs(X), is composed of the s leftmost bits of the bit string X. The last block is padded to the right (least significant bits) with a 1 and as many 0s as required so that it is also of length b if the message is not an integer multiple of the encryption block length. The block cipher is used on the block made up completely of 0 bits to get the two n-bit keys.

The first subkey is obtained from the ciphertext as a consequence of a one-bit left shift and, optionally, by XORing a constant that is block-size dependent. The first subkey is used to generate the second subkey in the same way.

AUTHENTICATION CODE FOR CHAINING-MESSAGE CIPHER BLOCK COUNTER

An authenticated encryption mode, as described in NIST SP 800-38C, is the CCM mode of operation. Systems of encryption that safeguard both the secrecy and the authenticity (integrity) of communications are referred to as authenticated encryption. Both types of security are necessary for many applications and protocols, but up until recently, the two services were created independently.

The AES encryption method the CTR mode of operation, and the CMAC authentication algorithm are the three main algorithmic components of CCM. Both the MAC algorithm and encryption utilize the same key, K. Three components make up the input to the CCM encryption process.

1. Information that will be encrypted and authenticated. The data block's plaintext message P is shown here.
2. Associated data A that won't be encrypted but will be authenticated
3. A nonce N that is linked to the data and payload.

PRINCIPLES OF PUBLIC-KEY CRYPTOGRAPHY

The 79 by zero or more A-containing blocks, then zero or more P-containing blocks. The output block sequence is sent into the CMAC algorithm, which generates a MAC value with length Tlen that is less than or equal to the block length.

A series of counters that are created for encryption must be separate from the nonce. Using the lone counter Ctr0 and CTR mode, the authentication tag is encrypted. To create an encrypted tag, the output's Tlen most significant bits are XORed with the tag. The plaintext is encrypted in CTR mode using the remaining counters. To create the ciphertext output, the encrypted tag and plaintext are combined.

PRINCIPLES OF PUBLIC-KEY CRYPTOGRAPHY

Public-key encryption, used for message authentication and key sharing, is equally important to traditional encryption. This section first examines the fundamental idea of public-key encryption and provides a quick overview of key distribution problems.

Structure for Public-Key Encryption

The concept of public-key encryption was originally put forward in print by Diffie and Hellman in 1976 [DIFF76], and it represents the field's first really significant development in literally thousands of years. As opposed to symmetric encryption methods, which employ straightforward operations on bit patterns, public-key algorithms are based on mathematical functions. Asymmetric public-key cryptography employs two different keys, which is more significant than symmetric conventional encryption, which uses only one key. In terms of secrecy, key distribution, and authentication, using two keys has significant ramifications.

We should first address a few widespread misunderstandings about public-key encryption before moving on. One is that compared to traditional encryption, public-key encryption is more

resistant to cryptanalysis. In actuality, two factors determine how secure an encryption system is: (1) the length of the key, and (2) the computing effort required to decipher a cipher. There is nothing fundamentally different between conventional and public-key encryption that would make one more effective at thwarting cryptanalysis. Another fallacy is the idea that public-key encryption is a universal method that has supplanted traditional encryption. Contrarily, there does not seem to be a foreseeable possibility that traditional encryption would be abandoned due to the computational expense of present public-key encryption systems. Using public-key encryption seems to make key distribution straightforward in comparison to the rather laborious handshaking required with key distribution centers for conventional encryption, which is the case with conventional encryption.

In reality, some kind of protocol is necessary, often including a central agent, and the associated steps are neither more straightforward nor more effective than those needed for traditional encryption. Public and private keys are a set of keys that have been chosen such that if one is used for encryption, the other may be used for decryption. The input public or private key determines the precise modifications that the encryption algorithm does. Ciphertext is the output message that has been encrypted. It depends on the plaintext and the key. Two separate keys will result in two distinct ciphertexts for the same message.

Decryption algorithm:

This algorithm accepts the ciphertext and the matching key and produces the original plaintext.

The pair's public key, as its names imply, is made available for usage by the general public, whilst the private key is exclusively known to its owner. One key is used for encryption and another, related key is used for decryption in a general-purpose public-key cryptographic technique.

The following are the crucial steps:

1. In order to encrypt and decode communications, each user produces a pair of keys.
2. Each user puts one of the two keys in an accessible file or public register.

a) The public key is this.
b) The companion key remains a secret.
c) Each user keeps a collection of public keys they have acquired from other users.

3. Bob encrypts a message using Alice's public key if he wants to send her a private message.
4. Alice uses her private key to decode the message when she gets it. Because only Alice has access to Alice's private key, no other receiver will be able to decode the message.

In this method, private keys are produced locally by each participant and never need to be shared since everyone has access to the public keys. Incoming communication is secure as long as a user safeguards his or her private key. A user may always update their private key and publish a

new associated public key to replace their previous one. The term "secret key" is often used to describe the key used in traditional encryption.

The public key and the private key are the two keys that make up public-key encryption. The private key is always kept hidden, but it's called a private key rather than a secret key to distinguish it from regular encryption.

Public-key cryptosystem applications

We need to make one point about public-key cryptosystems that is otherwise likely to be unclear before moving on. The employment of a cryptographic technique with two keys one kept secret and one made accessible to the public defines public-key systems. The sender performs some kind of cryptographic operation using either the sender's private key, the receiver's public key, or both, depending on the application. We may broadly divide the applications of public-key cryptosystems into three groups:

1. **Encryption and decryption:** A communication is encrypted using the recipient's public key by the sender.
2. **Digital signature:** The sender "signs" a message with its private key. A cryptographic technique is used to sign the message or a short block of data that functions as part of the message.
3. **Key exchange:** To exchange a session key, two parties work together. The use of one or both parties' private keys is involved in a number of different strategies.

While some algorithms can be used for one or two of these applications, others are only appropriate for one or two of these three. The applications that RSA and Diffie Hellman. Additionally listed in this table are the later-discussed elliptic-curve cryptography and the Digital Signature Standard (DSS).

Public-Key Cryptography Requirements

The cryptographic algorithm used by the system depicted in Figure 3.9 is based on two related keys. This system was proposed by Diffie and Hellman without any proof that such algorithms exist. They did, however, specify the requirements that such algorithms must meet [DIFF76]:

1. The generation of a pair (public key PUb, private key PRb) by party B is computationally simple.
2. Generating the corresponding ciphertext is computationally simple for sender A given the public key and the encrypted message, M:
3. C E(PUb, M) (PUb, M)Recovering the original message from the generated ciphertext using the private key is computationally simple for receiver B:
4. M D[PRb, E(PUb, M)] D(PRb, C)
5. It is computationally impossible for an adversary to ascertain the private key, PRb, even if they are aware of the public key, PUb.
6. Recovering the original message, M, is computationally impossible for an adversary who has access to the public key, PUb, and a ciphertext, C.

7. We can include a sixth condition that, while valuable, is not required for all public-key applications.Either related key can be used for encryption, and the other key can be used for decryption.
8. PUb, E(PRb, M), M = D PRb, E(PUb, M), D

REFERENCES:

[1] E. Dubrova, M. Näslund, G. Selander, and F. Lindqvist, "Message Authentication Based on Cryptographically Secure CRC without Polynomial Irreducibility Test," *Cryptogr. Commun.*, 2018, doi: 10.1007/s12095-017-0227-8.

[2] G. K. Sodhi *et al.*, "Preserving authenticity and integrity of distributed networks through novel message authentication code," *Indones. J. Electr. Eng. Comput. Sci.*, 2018, doi: 10.11591/ijeecs.v12.i3.pp1297-1304.

[3] S. V. Agievich, "EHE: Nonce misuse-resistant message authentication," *Prikl. Diskretn. Mat.*, 2018, doi: 10.17223/20710410/39/3.

[4] Q. Huang, Y. Yang, and Y. Shi, "SmartVeh: Secure and efficient message access control and authentication for vehicular cloud computing," *Sensors (Switzerland)*, 2018, doi: 10.3390/s18020666.

[5] D. Chen *et al.*, "An LDPC Code Based Physical Layer Message Authentication Scheme with Prefect Security," *IEEE J. Sel. Areas Commun.*, 2018, doi: 10.1109/JSAC.2018.2825079.

[6] E. Dubrova, G. Selander, M. Näslund, and F. Lindqvist, "Lightweight message authentication for constrained devices," in *WiSec 2018 - Proceedings of the 11th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2018. doi: 10.1145/3212480.3212482.

[7] Y. Xie, F. Xu, D. Li, and Y. Nie, "Efficient message authentication scheme with conditional privacy-preserving and signature aggregation for vehicular cloud network," *Wirel. Commun. Mob. Comput.*, 2018, doi: 10.1155/2018/1875489.

[8] X. Lu, W. Yin, Q. Wen, K. Liang, L. Chen, and J. Chen, "Message integration authentication in the internet-of-things via lattice-based batch signatures," *Sensors (Switzerland)*, 2018, doi: 10.3390/s18114056.

[9] M. R. Asaar, M. Salmasizadeh, W. Susilo, and A. Majidi, "A secure and efficient authentication technique for vehicular Ad-Hoc networks," *IEEE Trans. Veh. Technol.*, 2018, doi: 10.1109/TVT.2018.2822768.

CHAPTER 12

SYMMETRIC KEY DISTRIBUTION USING SYMMETRIC ENCRYPTION

Dr. Sovit Kumar, Assistant Professor
Department of Computer Science and Engineering, Sanskriti University, Mathura, Uttar Pradesh, India
Email Id- sovit.soeit@sanskriti.edu.in

Symmetric key distribution is a process used to securely distribute secret keys between two parties, allowing them to communicate securely using symmetric encryption. Symmetric encryption involves using a single secret key to both encrypt and decrypt data, unlike asymmetric encryption, which uses a pair of keys, one public and one private. In this, we will explore the basics of symmetric key distribution using symmetric encryption. We will cover the process of key generation, key exchange, and the various symmetric encryption algorithms that can be used.

Symmetric Key Generation

The first step in symmetric key distribution is the generation of the secret key. This key is generated by one of the two parties involved in the communication, and it must be kept secret from anyone else who might intercept the communication. One common method of generating symmetric keys is to use a pseudorandom number generator (PRNG) to produce a sequence of random bits. The length of the key depends on the specific algorithm being used, but it is typically between 128 and 256 bits. Once the key has been generated, it must be securely transmitted to the other party so that they can use it for encryption and decryption.

Symmetric Key Exchange

The process of securely transmitting the symmetric key from one party to the other is known as key exchange. There are several methods that can be used to exchange the key, including:

1. **In-person exchange:** The simplest and most secure method of key exchange is for the two parties to meet in person and physically exchange the key. This method is often used in high-security environments, such as military or government communications.
2. **Trusted third party:** A trusted third party, such as a certificate authority (CA), can be used to securely transmit the key between the two parties. The CA generates a public key and a private key for each party, and the public keys are exchanged. The parties then use their own private keys to decrypt the symmetric key, which has been encrypted using the other party's public key.
3. **Diffie-Hellman key exchange:** The Diffie-Hellman key exchange is a cryptographic protocol that allows two parties to establish a shared secret over an insecure communication channel. This method is widely used in internet communications, such as secure web browsing and virtual private networks (VPNs).

Symmetric Encryption Algorithms

There are several symmetric encryption algorithms that can be used to encrypt and decrypt data using the shared secret key. Some of the most commonly used algorithms include:

1. **Advanced Encryption Standard (AES):** AES is a widely used symmetric encryption algorithm that was chosen by the National Institute of Standards and Technology (NIST) as the standard encryption algorithm for US government organizations. It uses a block cipher with a key length of 128, 192, or 256 bits[1], [2].
2. **Blowfish:** Blowfish is a symmetric encryption algorithm that was designed by Bruce Schneier in 1993. It uses a variable-length key between 32 and 448 bits and is often used in virtual private networks (VPNs) and other network security applications.
3. **Triple DES (3DES):** 3DES is a symmetric encryption algorithm that uses three rounds of the Data Encryption Standard (DES) algorithm to provide greater security than the original DES algorithm. It uses a key length of 168 bits.
4. **Twofish:** Twofish is a symmetric encryption algorithm that was designed by Bruce Schneier in 1998. It uses a block cipher with a key length of 128, 192, or 256 bits and is often used in applications that require high levels of security.
5. **RC4:** RC4 is a symmetric encryption algorithm that was designed by Ron Rivest in 1987. It uses a variable-length key and is often used in wireless networking, as well as in some web browsers.

The two parties to an exchange must share the same key and keep it secure in order for symmetric encryption to function. Additionally, frequent key changes are often preferred to reduce the amount of data that might be compromised if an attacker discovers the key. Consequently, every cryptographic system's strength depends on using the key distribution approach, which is the process of sending a key to two people that want to share data without letting anybody else view it. Different methods may be used to distribute keys. The alternatives available for parties A and B are as follows:

9. A might decide on a key and hand-deliver it to B.
10. The key might be chosen and physically delivered to A and B by a third party.
11. If A and B have recently and previously used the same key, one of them may communicate the new key to the other while encrypting it using the old key.
12. If A and B each have an encrypted connection to a third party named C, then C may be able to send A and B a key across those encrypted connections.

For options 1 and 2, a key must be manually delivered. This is a realistic requirement for link encryption as each link encryption device will only be communicating with its partner at the other end of the connection. Manual distribution is problematic for end-to-end encryption across a network, however. Any particular host or terminal in a distributed system can eventually need to communicate with a large number of other hosts and terminals. Therefore, each device

requires a set of keys that are dynamically given. In a wide-area dispersed system, the issue is more challenging.

Link encryption or end-to-end encryption are both possible with option 3, but if a hacker ever manages to get even one key, all future 4.2/KERBEROS 99 keys are made public. Even if the connection encryption keys need to be changed often, this should be done manually. Option 4 should be used if end-to-end encryption keys are needed[3], [4].

Option 4 uses two different types of keys:

A. **Session key:** In order to communicate, two end systems (hosts, terminals, etc.) must first create a logical connection (e.g., virtual circuit). All user information is encrypted using a one-time session key for the length of that logical connection, known as a session. The session key is deleted at the end of the session.
B. **Long-term key:** For the purpose of sharing session keys, entities employ permanent keys.Option 4 needs a key distribution center as a component (KDC). Which systems are permitted to communicate with one another is decided by the KDC. A one-time session key is made available by the key distribution center when consent is given for two systems to connect.

Generally speaking, a KDC's operation goes as follows:

1. Host A sends a connectionrequest packet to the KDC when it wants to establish a connection with host B. Using a master key that is only known to A and the KDC, communication between the two parties is encrypted.
2. The KDC creates a special one-time session key if it accepts the connection request. It uses the permanent key it shares with A to encrypt the session key before sending the encrypted session key to A. Similar to that, it uses the permanent key it shares with B to encrypt the session key before giving it to B.
3. With the temporary session key as encryption key, A and B may now establish a logical connection and exchange messages and data.

The automated key distribution method offers the adaptability and dynamic properties required to let many users to access multiple servers and for the servers to communicate with one another. The most popular application that employs this strategy is Kerberos, which is covered in more detail in the next section.

KERBEROS

A key distribution and user authentication service called Kerberos was created at MIT.

This is the issue that Kerberos attempts to solve: Assume that users at workstations want to access services from servers that are dispersed around the network in an open, distributed environment. We want servers to be able to authenticate service request requests and limit access to authorized users. A workstation cannot be trusted under this setting to accurately identify its users to network services. There are in particular the following three dangers:

1. A user may acquire access to a specific workstation and impersonate another user while using it.
2. A user may change a workstation's network address to make requests issued from that workstation seem to be coming from a different workstation[5], [6].

DISTRIBUTION OF KEYS AND USER AUTHENTICATION

A user has the option to listen in on conversations and launch a replay attack to access a server or interfere with work. An unauthorized user could be able to access services and data in any of these scenarios even if they are not supposed to. Kerberos offers a centralized authentication server whose purpose is to authenticate users to servers and servers to users in place of developing complex authentication protocols at each server.

Kerberos doesn't employ public-key encryption and only uses symmetric encryption. In use are two different Kerberos versions. Although this version is being phased out, there are still implementations of version 4 [MILL88, STEI88]. Version 5 [KOHL94], which addresses some of version 4's security flaws, has been released as a proposed Internet Standard (RFC 4120). It is preferable to begin with an explanation of version 4 of Kerberos because to its complexity. This makes it possible for us to understand the core of the Kerberos approach without taking some of the specifics needed to address nuanced security concerns into account.

Kerberos Iteration

In a somewhat complex protocol, Version 4 of Kerberos uses DES to provide the authentication function. It is difficult to understand the need of the several components of the protocol when looking at it as a whole. Therefore, we follow Bill Bryant's [BRYA88] method of starting with a few fictitious exchanges and working our way up to the whole protocol. The complexity of each subsequent discussion increases in order to address the security flaws that the prior dialogue disclosed. After looking at the protocol, we move on to version 4's additional features.

AN EASY IDENTIFICATION DIALOGUE

Any client may request a service from any server in a network environment that isn't secured. The most significant security danger is impersonation. An adversary may impersonate another client to gain unapproved access to server computers. Servers must be able to verify the identity of service requesters in order to mitigate this hazard. For each client/server contact, each server may be forced to do this activity, however in an open environment, this creates a significant strain on each server.

Utilizing an authentication server (AS) that has access to all users' credentials and keeps them all in one place is an option. Each server also receives a separate secret key from the AS. These keys were handed over physically or in another safe way. The sender and recipient are indicated to the left of the colon, the message's contents are indicated to the right, and the concatenation sign (#7) is indicated to the left of the colon.

101 KERBEROS

IDC network address of user on C Kv secret encryption key shared by AS and V PC password of user on C IDV identifier. In this case, the user signs on to a workstation and asks to connect to server V. The user's workstation's client module C queries the user for their password before sending a message to the AS with the user's ID, the server's ID, and the user's password. In order to determine if the user is authorized to access server V and whether the user has provided the correct password for this user ID, the AS consults its database. If the user passes both checks, the AS acknowledges them as valid and must now persuade the server that the user is legitimate. The AS produces a ticket including the user's ID, network address, and server's ID in order to do this. The secret key that the AS and this server have allows for the encryption of this ticket. Then C receives this ticket back. The ticket cannot be changed by C or an opponent since it is encrypted. Now that C has this ticket, he or she may ask V for assistance. C sends V a message that includes the ticket and his ID. The user ID in the ticket is decrypted, and V then confirms that it matches the message's unencrypted user ID. The server accepts the user as authenticated and provides the requested service if these two matches[7], [8].

Each component of message (3) has importance. To guard against forging or tampering, the ticket is encrypted. The ticket contains the server's ID (IDV) so that the server can confirm that the ticket has been correctly encrypted. The IDC symbol on the ticket serves as a reminder that C authorized its issuance. Finally, ADC works to neutralize the ensuing danger. An adversary may intercept the ticket sent in message (2), then, assuming the name IDC, send a message of type (3) from a different workstation. The user on that other workstation would be given access by the server after receiving a legitimate ticket that matches the user ID. The network address of the first request is given in the ticket by the AS to thwart this attack. The ticket is no longer valid unless it is sent from the original workstation from which it was requested.

AN AUTHENTICATION DIALOGUE THAT IS MORE SECURE

Even if the aforementioned scenario addresses some of the issues with authentication in an open network setting, issues still exist. Particularly, two stick out. We first want to reduce the amount of times a user needs to input a password. Assume that each ticket may only be used once. User C must provide a password to get a ticket for the mail server if C wants to check his or her mail at a mail server after logging on to a workstation in the morning. Every time C tries to check his or her mail throughout the day, the password must be entered again. By stating that tickets are reusable, the situation may be made better.

The workstation may save the mail-server ticket when it is obtained and utilize it on behalf of the user for several visits to the mail server during a single login session. A user would still need a separate ticket for each distinct service under this plan, however. The first time a user wanted to access a print server, mail server, file server, etc., a new ticket would be needed, therefore the user would need to enter the password. The second issue is that the password was sent in plaintext in the preceding instance. Any service that the victim has access to may be used by an eavesdropper who had the password.

We present a method for avoiding plaintext passwords and a new server called the ticket-granting server to address these additional issues (TGS). The updated scenario, which is still speculative, is as follows.

Users who have been authenticated to AS are issued tickets via the new service, TGS. As a result, the user starts by asking the AS for a ticket-granting ticket (Tickettgs). This ticket is saved by the client module at the user workstation. The client applies to the TGS and uses the ticket to authenticate itself each time the user needs access to a new service. The TGS then issues a ticket for the specific service. Each time a certain service is requested, the client utilizes the saved service-granting tickets to authenticate the user to the server. Let's examine the specifics of this plan:

1. The client sends the TGS ID and its user's ID to the AS, together with a request to utilize the TGS service, to request a ticket-granting ticket on behalf of the user.
2. In response, the AS sends a ticket that has been encrypted using a key that was created using the user's previously saved password (KC). When the client receives this answer, it asks the user for a password, creates the key, and makes an attempt to decode the incoming message. The ticket is successfully retrieved if the right password is provided.

Only the appropriate person should have access to the password, and only that user should be able to get the ticket. We avoided having to provide the password in plaintext by using the password to gain credentials via Kerberos. The user's ID, network address, and TGS ID are all included on the ticket itself.

KERBEROS 103

This relates to the first situation. The client may utilize this ticket to make several service-granting requests, according to the concept. Therefore, the ticket that grants entry must be redeemed. We do not, however, want a rival to be able to seize the ticket and utilize it. Think about the following example: Once the user has logged off of their workstation, the adversary takes control of the login ticket. The adversary then either obtains access to that computer or sets up his computer using the victim's network address. The rival might use the ticket again to spoof the TGS. To combat this, the ticket has a lifespan and a timestamp that show the time and date that it was issued and the duration of its validity, respectively (e.g., eight hours). The client no longer has to ask the user for a password for each new service request since they have a reusable ticket. The ticket-granting ticket is encrypted with a secret key that is only known to the AS and the TGS, which is the last point to make. This stops the ticket from being altered. With a key derived from the user's password, the ticket is re-encrypted. By providing the necessary authentication, this ensures that only the right user will be able to retrieve the ticket.

A Kerberos realm is the term used to describe such a setting. The following is an explanation of the realm notion. A collection of controlled nodes using the same Kerberos database is known as a realm. The Kerberos master computer system, which must be housed in a physically secure room, houses the Kerberos database. Other Kerberos computer systems may host a read-only copy of the Kerberos database. However, the master computer system is where all database

modifications must be done. The Kerberos master password is necessary in order to modify or access a Kerberos database. A Kerberos principal is a service or user that is recognized by the Kerberos system, and it is a related idea. The principal name is how Kerberos principals are recognized. A service or user name, an instance name, and a realm name make up principal names[9], [10]. Typically, distinct realms are networks of clients and servers operated by several administrative bodies. That is to say, having users and servers in one administrative domain registered with a Kerberos server elsewhere is either not possible or is against administrative policy. However, users from one realm could need access to servers from another, and certain servers might be open to serving users from other realms as long as they can be verified as legitimate.

A means for providing such interrealm authentication is provided by Kerberos. A third prerequisite is introduced in order for two realms to allow interrealm authentication: A secret key is shared between each Kerberos server in an interoperating realm and the server in the other realm. The two Kerberos servers have each other's registrations.

The Kerberos server in one realm must be trusted by the Kerberos server in the other realm in order for the scheme to work. The second realm's participating servers must likewise be willing to put their faith in the first realm's Kerberos server. A ticket for that server is required if a user wants service on a server in a different realm. After completing the customary steps to access the local TGS, the user's client requests a ticket-granting ticket for a remote TGS (a TGS in a different realm). The client may then submit an application to the remote TGS to request a service-granting ticket for the chosen server inside the remote TGS's domain. The previous method had the drawback of not scaling effectively over several kingdoms. For each Kerberos realm to communicate with every other Kerberos realm, there must be N(N 1)/2 secure key exchanges if there are N realms.

REFERENCES:

[1] J. A. Grieve, R. Bedington, Z. Tang, R. C. M. R. B. Chandrasekara, and A. Ling, “SpooQySats: CubeSats to demonstrate quantum key distribution technologies,” *Acta Astronaut.*, 2018, doi: 10.1016/j.actaastro.2018.06.005.

[2] R. Wardoyo, E. Setyaningsih, and A. K. Sari, “Symmetric key distribution model using RSA-CRT method,” in *Proceedings of the 3rd International Conference on Informatics and Computing, ICIC 2018*, 2018. doi: 10.1109/IAC.2018.8780446.

[3] R. Kuchipudi, A. A. M. Qyser, V. V. S. S. S. Balaram, and A. Manusha Reddy, “Energy efficient key distribution for wireless sensor networks,” *Int. J. Eng. Technol.*, 2018, doi: 10.14419/ijet.v7i4.6.20234.

[4] S. A. Jassim and W. K. Awad, “Searching over encrypted shared data via cloud data storage,” *J. Theor. Appl. Inf. Technol.*, 2018.

[5] S. Varghese and S. M. C. Vigila, “A varied approach to attribute based access model for secure storage in cloud,” in *Proceedings of 2017 International Conference on Innovations in Information, Embedded and Communication Systems, ICIIECS 2017*, 2018. doi: 10.1109/ICIIECS.2017.8276130.

[6] R. Bhavani, K. S. Suganya, and D. Yazhini Priyanka, "Autonomous PHR Sharing: A Patient Centric Scalable and Flexible e-Healthcare Framework," *Int. J. Sci. Res. Netw. Secur. Commun.*, 2018, doi: 10.26438/ijsrnsc/v6i2.1114.

[7] L. Lv, W. Sun, X. Yang, and X. Wang, "Key encapsulation mechanism from multilinear maps," in *Lecture Notes on Data Engineering and Communications Technologies*, 2018. doi: 10.1007/978-3-319-59463-7_35.

[8] D.-I. Curiac, F. Dragan, O. Banias, and D. Iercan, "A knowledge based system approach in securing distributed wireless sensor networks," *arXiv*. 2018.

[9] O. Datcu, R. Hobincu, M. Stanciu, and R. A. Badea, "Encrypting multimedia data using modified baptista's chaos-based algorithm," in *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, 2018. doi: 10.1007/978-3-319-92213-3_27.

[10] S. K. Desai, A. Dua, N. Kumar, A. K. Das, and J. J. P. C. Rodrigues, "Demand Response Management Using Lattice-Based Cryptography in Smart Grids," in *2018 IEEE Global Communications Conference, GLOBECOM 2018 - Proceedings*, 2018. doi: 10.1109/GLOCOM.2018.8647560.

CHAPTER 13

PUBLIC-KEY INFRASTRUCTURE

Dr. Ravindra Kumar, Assistant Professor
Department of Computer Science and Engineering, Sanskriti University, Mathura, Uttar Pradesh, India
Email Id- ravindrak.soeit@sanskriti.edu.in

Public-Key Infrastructure (PKI) is a system used to create, manage, and distribute digital certificates that authenticate the identities of users, devices, and other entities in a networked environment. PKI is a critical component of many modern security systems, providing a means of secure communication, digital signatures, and encryption.

The need for PKI arises from the problem of trust in a networked environment. In a traditional face-to-face transaction, trust is established by personal interactions, but in a networked environment, it is necessary to rely on digital certificates to authenticate the identities of the parties involved in a transaction. PKI provides a mechanism for creating and managing these digital certificates, and for ensuring that they are trusted by all parties in the network.

In a PKI system, there are two types of keys: public keys and private keys. The public key is used to encrypt data and is available to everyone, while the private key is used to decrypt data and must be kept secret. Each entity in the network has a public and private key pair, which is used to encrypt and decrypt messages sent between them. The private key is kept secret and is only accessible by the owner of the key, while the public key is widely distributed and available to anyone who needs to send encrypted messages to that entity[1], [2].

To ensure that the public key is associated with the correct identity, a digital certificate is created that contains the public key and other identifying information, such as the name of the entity and the name of the issuing organization. The digital certificate is signed by a trusted third party, known as a Certificate Authority (CA), which vouches for the identity of the entity. This creates a chain of trust, in which the digital certificate is trusted because it is signed by a trusted CA.

The process of creating a digital certificate and signing it with a CA is known as certificate enrollment. When a new entity is added to the network, it must first generate a public and private key pair, and then request a digital certificate from a CA. The CA will verify the identity of the entity and issue a digital certificate that can be used to authenticate the identity of the entity in future transactions.

PKI is used in a wide range of applications, including secure web browsing, email encryption, and digital signatures. In secure web browsing, PKI is used to encrypt data sent between the user's browser and the web server, ensuring that sensitive information, such as passwords and credit card numbers, cannot be intercepted by unauthorized parties. In email encryption, PKI is used to encrypt email messages, ensuring that they can only be read by the intended recipient. In digital signatures, PKI is used to provide a means of verifying the authenticity of a document, by

allowing the owner of the private key to sign the document and provide a digital signature that can be verified by anyone with access to the public key.

PKI relies on a number of different technologies and standards, including the X.509 standard for digital certificates, the Transport Layer Security (TLS) protocol for secure web browsing, and the S/MIME standard for email encryption. PKI also requires the use of hardware security modules (HSMs) to securely store and manage private keys[3], [4].

Despite its many benefits, PKI can be complex and difficult to manage, particularly in large-scale environments. Key management is a critical aspect of PKI, and it is important to ensure that private keys are properly protected and managed throughout their lifecycle. Additionally, the process of certificate revocation, which is necessary when a digital certificate is compromised or when an entity's identity changes, can be difficult to manage in large-scale environments.

In recent years, there has been increased interest in alternative technologies to PKI, such as blockchain-based solutions. These solutions seek to provide a decentralized means of establishing trust, without relying on while these solutions have some potential benefits, they also come with their own set of challenges, and it remains to be seen whether they will be able to fully replace PKI in the future. Overall, PKI is a critical component of modern security systems, providing a means of establishing trust and securing communications in networked environments. While it can be complex and difficult to manage, PKI remains a powerful tool for securing data and transactions, and is likely to continue to play an important role in the future of cybersecurity.

Public-key infrastructure (PKI) is defined by RFC 2822 (Internet Security Glossary) as the necessary gear, software, personnel, guidelines, and practices Asymmetric cryptography-based digital certificates may be managed, stored, distributed, and revoked. The main goal of creating a PKI is to make it possible to acquire public keys in a safe, simple, and effective manner. A task force on Internet engineering. A formal (and general) model based on X.509 has been established thanks to the efforts of the (IETF) Public Key Infrastructure X.509 (PKIX) working group[5], [6].

REFERENCES:

[1] V. Lozupone, “Analyze encryption and public key infrastructure (PKI),” *Int. J. Inf. Manage.*, 2018, doi: 10.1016/j.ijinfomgt.2017.08.004.

[2] K. Balan *et al.*, “RSSI and public key infrastructure based secure communication in Autonomous Vehicular Networks,” *Int. J. Adv. Comput. Sci. Appl.*, 2018, doi: 10.14569/IJACSA.2018.091243.

[3] A. S. Konoplev, A. G. Busygin, and D. P. Zegzhda, “A Blockchain Decentralized Public Key Infrastructure Model,” *Autom. Control Comput. Sci.*, 2018, doi: 10.3103/S0146411618080175.

[4] V. S. Janani and M. S. K. Manikandan, “Efficient trust management with Bayesian-Evidence theorem to secure public key infrastructure-based mobile ad hoc networks,” *Eurasip J. Wirel. Commun. Netw.*, 2018, doi: 10.1186/s13638-017-1001-5.

[5] A. Alrawais, A. Alhothaily, X. Cheng, C. Hu, and J. Yu, "SecureGuard: A Certificate Validation System in Public Key Infrastructure," *IEEE Trans. Veh. Technol.*, 2018, doi: 10.1109/TVT.2018.2805700.

[6] K. Prakasha, B. Muniyal, V. Acharya, S. Krishna, and S. Prakash, "Efficient digital certificate verification in wireless public key infrastructure using enhanced certificate revocation list," *Inf. Secur. J.*, 2018, doi: 10.1080/19393555.2018.1516836.

CHAPTER 14

WEB SECURITY CONSIDERATIONS

Mr. Aishwary Awasthi, Research Scholar
Department of Mechanical Engineering, Sanskriti University, Mathura, Uttar Pradesh, India
Email Id- aishwary@sanskriti.edu.in

Web security is the practice of protecting websites, web applications, and web services from attacks and unauthorized access. In recent years, web security has become increasingly important due to the rise in cyberattacks and the amount of sensitive information stored online. In this article, we will discuss the most important web security considerations that every developer, administrator, and user should know.

1. Authentication and Authorization

Authentication is the process of verifying the identity of a user, while authorization is the process of determining what a user can access once they are authenticated. Proper authentication and authorization are crucial to web security as they prevent unauthorized access to sensitive information. Authentication can be achieved using various methods, such as passwords, biometric authentication, or two-factor authentication. Authorization can be achieved by implementing role-based access control (RBAC), which grants access to resources based on a user's role.

2. Secure Communication

Communication between a user and a web server should be encrypted to prevent eavesdropping and man-in-the-middle attacks. Hypertext Transfer Protocol Secure (HTTPS) is a secure version of HTTP that encrypts data in transit using Transport Layer Security (TLS) or Secure Sockets Layer (SSL). SSL/TLS certificates can be obtained from trusted certificate authorities and should be configured correctly to prevent errors or vulnerabilities.

3. Input Validation

Input validation is the process of verifying that user input is valid and safe to use. Malicious input, such as SQL injections or cross-site scripting (XSS) attacks, can cause serious damage to a website or web application. Input validation can be achieved by using input filters and limiting the input types and length.

4. Cross-Site Scripting (XSS)

XSS attacks occur when a website or web application includes untrusted data in a web page without proper validation or encoding. This allows an attacker to inject malicious scripts, which can steal user data or redirect users to malicious websites. Preventing XSS attacks can be achieved by encoding user input, sanitizing user input, and using content security policy (CSP) headers.

5. Cross-Site Request Forgery (CSRF)

CSRF attacks occur when an attacker tricks a user into performing an action they did not intend to perform, such as clicking a malicious link or submitting a form on a malicious website. CSRF attacks can be prevented by using a CSRF token, which is a unique identifier that is attached to every form submission or request[1], [2].

6. SQL Injection

SQL injections occur when an attacker injects malicious SQL statements into a website or web application's database query, which allows them to access sensitive information or modify the database. Preventing SQL injections can be achieved by using parameterized queries, input validation, and avoiding dynamic SQL queries.

7. Secure File Uploads

File uploads are a common feature of web applications, but they can also be used to upload malicious files that can compromise the security of the website or web application. Preventing file upload vulnerabilities can be achieved by verifying the file type, limiting the file size, and storing uploaded files outside the web root directory.

8. Password Security

Passwords are the most common form of authentication, and they should be stored securely to prevent unauthorized access to user accounts. Password security can be improved by enforcing strong password policies, using password hashing algorithms, and using multi-factor authentication.

9. Server-Side Security

Server-side security is crucial to web security, as it protects the server and the web application from attacks. Server-side security can be achieved by updating software regularly, using firewalls, and implementing intrusion detection and prevention systems.

10. User Education

Users play a crucial role in web security, as they are often the first line of defense against attacks. User education can be achieved by providing security awareness training, implementing strong password policies, and encouraging users to report suspicious activity. Both Internet and TCP/IP intranets are used to execute the client/server application that makes up the majority of the World Wide Web. As a result, the safety techniques and technologies covered so far in this book are applicable to the problem of web security. The Web poses fresh difficulties that are underappreciated in terms of computer and network security, according to [GARF02]. The Internet is bidirectional. The Web is susceptible to assaults on the Web servers through the Internet, unlike conventional publishing environments—even electronic publishing systems employing teletext, voice response, or fax-back.

Considerations for web security

1. The Web is becoming a more prominent channel for company and product information as well as a venue for commercial transactions. If the Web servers are compromised, reputations might be ruined and money could be lost.
2. The underlying software is very sophisticated, despite the fact that web browsers are quite simple to use, web servers are generally simple to deploy and administer, and developing web content is becoming easier. Many possible security weaknesses may be concealed by this complicated program. The brief history of the Web is full with instances of newly installed, updated systems that are open to various security threats.
3. A Web server may be used as a launching pad into the whole computer network of a company or government organization. Once the Web server has been compromised, an attacker could be able to access information and systems that aren't really linked to the Web but are instead connected to the server at the local site.
4. Web-based services often cater to casual and inexperienced (in security considerations) consumers. These users lack the resources and information necessary to implement efficient defenses, and they may not be aware of the security dangers that are there[3], [4].

Internet security risks

These dangers may be categorized in terms of passive and aggressive assaults. Eavesdropping on network traffic between a browser and a server and getting access to data on a website that is meant to be limited are examples of passive attacks. Active attacks include spoofing other users, tampering with communications while they are being sent between a client and a server, and changing data on a website. The location of the danger, such as a web server, browser, or network activity between a browser and a server, may also be used to categorize web security concerns. Computer system security concerns include those relating to server and browser security; Part Four of this book discusses system security generally but also applies to web system security. This chapter addresses topics related to network security, which includes concerns with traffic security.

Approaches for Securing Web Traffic

There are many methods for offering Web security. The numerous strategies that have been taken into consideration are comparable in terms of the services they provide and, to some degree, in terms of the mechanisms they use, but they vary in terms of their applicability and relative positioning within the TCP/IP protocol stack. Additionally, IPsec has a filtering feature so that only certain traffic needs to be subject to the overhead of IPsec processing. Implementing security right above TCP is a different, more universal approach. The Secure is the best illustration of this strategy.

There are two implementation options at this level. For complete generality, SSL (or TLS) might be made available as a component of the underlying protocol suite and be invisible to

applications as a result. As an alternative, SSL might be included with certain products. For instance, SSL is built into the Netscape and Microsoft Explorer browsers, and it is used by the majority of Web servers.

The particular application has embedded security services that are application-specific. The examples of this design are shown in Figure 5.1c. The benefit of this strategy is that the service may be customized to meet the unique requirements of a particular application. Integrity, User Data Modification, Trojan Horse Browser, Memory Modification, Modification of Message Traffic in Transit, Information Loss, Machine Compromise Availability to all other dangers

Cryptographic checksums

Internet eavesdropping, information loss, data theft from clients and servers, knowledge of network setup, knowledge of which clients communicate with servers, loss of privacy, and information theft are all concerns.

Web proxies, and encryption

Denial of Service, user thread termination, flooding the system with erroneous requests, memory or disk overflow, and DNS attack isolation disruptive, irritating, prevent the user from finishing their task Authentication is difficult to avoid, and unauthorized users are often impersonated

1. Data fraud
2. User misrepresentation
3. Acceptance of misleading information as true
4. Cryptographic techniques

SECURE TRANSPORT LAYER AND SECURE SOCKET LAYER

1. IP \sTCP
2. Record Protocol for SSL
3. SSL \sHandshake \sProtocol
4. Change Cipher Spec Protocol for SSL
5. Alert Protocol for SSL in HTTP Protocol Stack for SSL

TRANSPORT LAYER AND SECURE SOCKET LAYER SECURITY

Initiated by Netscape, SSL. The protocol's third version was developed with assistance from industry and was released as an Internet draft document. When it was decided to submit the protocol for Internet standardization, the IETF's TLS working group was established to create a shared standard. This first release of TLS is effectively an SSLv3.1 and is backwards compatible with SSLv3 as well as being quite similar to it. A discussion of SSLv3 is the focus of this section. The key distinctions between TLS and SSLv3 are discussed in the next section.

SSL Technology

TCP is intended to be used by SSL in order to provide a dependable end-to-end secure service. The SSL Record Protocol offers fundamental security services to different higher-layer protocols, although SSL is really two levels of protocols. SSL may be used on top of the

Hypertext Transfer Protocol (HTTP), which offers the transfer service for Web client/server interaction. The Handshake Protocol, Change Cipher Spec Protocol, and Alert Protocol are three higher-layer protocols that are included in SSL. These SSL-specific protocols, which are used to handle SSL exchanges, are covered in more detail in the section below.

1. The SSL session and SSL connection are two crucial SSL concepts that are specified in the standard as follows.
2. Connection: According to the OSI layering model, a connection is a transport that offers the appropriate kind of service. These partnerships are peer-to-peer ones for SSL. The relationships are fleeting. One session is connected to each connection.
3. SSL sessions are connections between a client and a server. The Handshake Protocol establishes sessions. Sessions establish a collection of cryptographic security settings that may be shared by several connections. The costly negotiation of new security settings for each connection is avoided by using sessions.

Any two parties (applications like HTTP on the client and server) may have several secure connections to one another. Although it is not common in reality, many simultaneous sessions between parties are theoretically possible. Each session has a variety of states connected to it. Once a session has been formed, read and write operations (also known as receive and transmit) are in a current working state. Additionally, pending read and write states are established during the Handshake Protocol. The pending states change to the current states when the Handshake Protocol is successfully completed[5], [6].

The following factors determine a session state.

1. Session identifier: The server may identify an active or resumeable session state by a random byte sequence.
2. Peer certification: The peer's X509.v3 certificate. This state component's value might be null.
3. Compression algorithm: The technique used to reduce the size of data before it is encrypted.
4. Cipher specification: Describes the hash method (such as MD5 or SHA-1) used to calculate the MAC as well as the bulk data encryption scheme (such as null, AES, etc.). Additionally, it specifies cryptographic characteristics like hash size. 48-byte shared secret between the client and server is known as the "master secret."
5. Is resumable: A flag letting you know if you may start new connections with the current session.

The following factors determine a connection status.

A. Server and client random: For each connection, the server and client choose random byte sequences. The secret key used in MAC operations on data transmitted by the server is known as the "Server Write MAC Secret."
B. Client write MAC secret: The private key used to perform MAC operations on client-sent data.

C. Server write key: The private encryption key used to encrypt and decode client-decrypted data.
D. Client write key: The symmetric encryption key used for client-side data encryption and server-side decryption.
E. Initialization vectors: Each key in a block cipher operating in CBC mode has its own initialization vector (IV). The SSL Handshake Protocol initially initializes this variable. The last block of ciphertext from each record is then saved to be used as the IV with the subsequent record.
F. Sequence numbers: For each connection, each party keeps a unique sequence number for messages that are sent and received.

Record Protocol for SSL

For SSL connections, the SSL Record Protocol offers the following two services:

A. Discretion: The Handshake Protocol specifies a shared secret key that is used for standard SSL payload encryption.
B. Message Integrity: The Handshake Protocol also specifies how to create a message authentication code using a shared secret key (MAC).

The general behavior of the SSL Record Protocol is seen in Figure 5.3. The Record Protocol takes an application message that has to be delivered, breaks the data up into smaller, more manageable blocks, may optionally apply a MAC, encrypt, and add a header before sending the resultant unit across a TCP segment. Prior to being sent to higher-level users, received data are encrypted, validated, decompressed, and reassembled.

Fragmentation is the initial action. Each message in the higher layer is broken up into blocks that are 214 bytes (16384 bytes) or smaller. Compression is then optionally used. Compression has to be lossless and may only lengthen information by a maximum of 1024 bytes. 1 The default compression method in SSLv3 (and the current version of TLS) is null since no compression technique is provided.

Computing a message authentication code over the compressed data is the next stage of processing. An agreed-upon secret key is utilized for this. Naturally, one hopes that compression causes data to shrink rather than grow. However, for extremely small blocks, it is conceivable that the compression method will actually produce output that is longer than the input due to formatting rules.

Connection Termination

By adding the next line to an HTTP record, either an HTTP client or HTTP server may signal the termination of a connection: Close connection. This means that when this record is provided, the connection will be cut off. Closing the underlying TCP connection is necessary for TLS to terminate the connection with the peer TLS entity on the remote side in order to terminate the HTTPS connection. Each side should utilize the TLS alert protocol to deliver a close notify alert in order to properly terminate a connection at the TLS level. Before severing a connection, TLS implementations must start an exchange of closure alerts. An "incomplete close" is produced

when a TLS implementation closes a connection before waiting for the peer to send its closure alert[7], [8].

The implementation that does this might decide to reuse the session, so keep that in mind. This should only be carried out when the application is certain that it has received all the message data that it requires (typically through the detection of HTTP message boundaries).

Additionally, HTTP clients must be able to handle a scenario in which the underlying TCP connection is closed without a Connection: close indicator or a prior close notify alert. Such a circumstance might result from a 162 programming issue. A communication error or a TRANSPORT-LEVEL SECURITY error on the server can both result in the termination of the TCP connection. The unexpected TCP closure, however, might be proof of an attack. Therefore, when this happens, the HTTPS client should display some sort of security alert. A secure network communication protocol called Secure Shell (SSH) was created with ease of use and low cost in mind. To replace TELNET and other insecure remote logon schemes, the initial version of SSH1 was designed to offer a secure remote logon facility. SSH can be used for network operations like file transfers and email because it also offers a more general client/server capability. SSH2, a new version, corrects several security issues with the initial plan.

For the majority of operating systems, there are numerous SSH client and server applications available. It has quickly evolved into one of the most widely used applications for encryption technology outside of embedded systems and has replaced other methods for remote login and X tunneling. Three protocols that typically sit on top of TCP make up SSH's organizational structure:

ransport Layer Protocol: This protocol offers server authentication, data confidentiality, and data integrity with forward secrecy, which ensures that the security of earlier sessions is unaffected if a key is compromised during one session.

1. Compression may optionally be offered by the transport layer.
2. Protocol for SSH User Authentication
3. Transport Layer Protocol for SSH
4. The TCP/IP Internet protocol allows for the delivery of datagrams across various networks.
5. End-to-end delivery using the transmission control protocol is dependable and connection-oriented.
6. guarantees server confidentiality, integrity, and authentication.
7. Compression is an optional extra that it might offer.
8. enables server authentication for the client-side user.
9. The encrypted tunnel is multiplexed into different logical channels using the SSH Connection Protocol.
    A. **User Authentication Protocol:** Authenticates the user to the server.
    B. **Connection Protocol:** Multiplexes multiple logical communications channels over a single, underlying SSH connection.

C. **Transport Layer Protocol:** HOST KEYS Server authentication occurs at the transport layer, based on the server possessing a public/private key pair. A server may have multiple host keys using multiple different asymmetric encryption algorithms. Multiple hosts may share the same host key. In any case, the server host key is used during key exchange to authenticate the identity of the host. For this to be possible, the client must have a \priori knowledge of the server's public host key. RFC 4251 dictates two alternative \trust models that can be used:
   1. The client has a local database that associates each host name (as typed by \the user) with the corresponding public host key. This method requires no \centrally administered infrastructure and no third-party coordination. The \downside is that the database of name-to-key associations may become \burdensome to maintain.
   2. The host name-to-key association is certified by a trusted certification authority (CA) (CA). The client only knows the CA root key and can verify the validity of all host keys certified by accepted CAs. This alternative eases the maintenance problem, since ideally, only a single CA key needs to be securely stored on theclient. On the other hand, each host key must be appropriately certified by a central authority before authorization is possible.

PACKET EXCHANGE

The sequence of events in the SSH Transport Layer Protocol. First, the client establishes a TCP connection to the server[9], [10]. This is done via the TCP protocol and is not part of the Transport Layer Protocol. Once the connection is established, the client and server exchange data, referred to as packets, in the data field of a TCP segment. Each packet is in the following format.

A. Packet length: Length of the packet in bytes, not including the packet length and MAC fields.
B. Padding length: Length of the random padding field.
C. Payload: Useful contents of the packet. Prior to algorithm negotiation, this field is uncompressed. If compression is negotiated, then in subsequent packets, this field is compressed.
D. Random padding: Once an encryption algorithm has been negotiated, this field is added. It contains random bytes of padding so that that total length of the packet (excluding the MAC field) is a multiple of the cipher block size, or \s8 bytes for a stream cipher.
E. Message authentication code (MAC): If message authentication has been negotiated, this field contains the MAC value. The MAC value is computed over the entire packet plus a sequence number, excluding the MAC field. The sequence number is an implicit 32-bit packet sequence that is initialized to 164 transport-level security 0 for the initial packet and increased for every packet.

The sequence number is not included in the packet sent over the TCP connection. Once an encryption algorithm has been negotiated, the entire packet excluding the MAC field is encrypted after the MAC value is calculated. The SSH Transport Layer packet exchange consists of a sequence of steps. The first step, the identification string exchange, begins with the client \sending a packet with an identification string of the form: SSH-proto version-software version SP comments CR LF where SP, CR, and LF are space character, carriage return, and line feed, respectively.

An example of a valid string is SSH-2.0-billsSSH 3.6.3q3<CR><LF>. The server responds with its own identification string. These strings are used in the DiffieHellman key exchange.

Next comes algorithm negotiation. Each side sends an SSH MSG KEXINIT containing lists of supported algorithms in the order of preference to the sender. There is one list for each type of cryptographic algorithm. The algorithms include key exchange, encryption, MAC algorithm, and compression algorithm. For each category, the algorithm chosen is the first algorithm on the client's list that is also supported by the server.

The next step is key exchange. The specification allows for alternative methods of key exchange, but at present, only two versions of Diffie-Hellman key exchange are specified. Both versions are defined in RFC 2409 and require only one packet in each direction. The following steps are involved in the exchange. In this, C is the client; S is the server; is a large safe prime; is a generator for a subgroup of GF( ); is the order of the subgroup; V S is S's identification string; V C is C's identification string; K S is S's public host key; I C is C's SSH MSG KEXINIT message and I S is S's SSH MSG KEXINIT message that have been exchanged before this part begins. The values of and are known to both client and server as a result of the algorithm selection negotiation. The hash function hash() is also decided during algorithm negotiation.

REFERENCES:

[1] B. K. Ayeni, J. B. Sahalu, and K. R. Adeyanju, "Detecting Cross-Site Scripting in Web Applications Using Fuzzy Inference System," *J. Comput. Networks Commun.*, 2018, doi: 10.1155/2018/8159548.

[2] M. B. Shuaibu and R. A. Ibrahim, "Web application development model with security concern in the entire life-cycle," in *4th IEEE International Conference on Engineering Technologies and Applied Sciences, ICETAS 2017*, 2018. doi: 10.1109/ICETAS.2017.8277849.

[3] L. Marrero and O. M. Brüggemann, "Institutional violence during the parturition process in Brazil: integrative review," *Revista brasileira de enfermagem*. 2018. doi: 10.1590/0034-7167-2017-0238.

[4] A. K. Kassem, B. Daya, and P. Chauvet, "A proposed methodology on predicting visitor's behavior based on web mining technique," *Int. J. Adv. Comput. Sci. Appl.*, 2018, doi: 10.14569/IJACSA.2018.091236.

[5] B. McFadden, T. Lukasiewicz, J. Dileo, and J. Engler, "Security Chasms of WASM," *NCC Gr. Whitepaper*, 2018.

[6] C. Vélez Álvarez, C. P. Jaramillo Ángel, and A. Giraldo Osorio, "Teaching-service: Social responsibility in the training of human talent in health in Colombia," *Educacion Medica*. 2018. doi: 10.1016/j.edumed.2017.08.002.

[7] L. Gaur and K. Anshu, "Consumer preference analysis for websites using e-TailQ and AHP," *Int. J. Eng. Technol.*, 2018, doi: 10.14419/ijet.v7i2.11.10999.

[8] C. Vélez Álvarez, C. P. Jaramillo Ángel, and A. Giraldo Osorio, "Docencia-servicio: responsabilidad social en la formación del talento humano en salud en Colombia," *Educ. Médica*, 2018, doi: 10.1016/j.edumed.2017.08.002.

[9] M. H. Shirvani, "Web Service Composition in multi-cloud environment: A bi-objective genetic optimization algorithm," in *2018 IEEE (SMC) International Conference on Innovations in Intelligent Systems and Applications, INISTA 2018*, 2018. doi: 10.1109/INISTA.2018.8466267.

[10] M. Devare, "Analysis and design of IoT based physical location monitoring system," *Adv. Parallel Comput.*, 2018, doi: 10.3233/978-1-61499-882-2-120.

CHAPTER 15

WIRELESS LAN

Dr. Pooja Sagar, Assistant Professor
Department of Computer Science and Engineering, Sanskriti University, Mathura, Uttar Pradesh, India
Email Id- pooja@sanskriti.edu.in

Web security is the practice of protecting websites, web applications, and web services from attacks and unauthorized access. In recent years, web security has become increasingly important due to the rise in cyber-attacks and the amount of sensitive information stored online. In this article, we will discuss the most important web security considerations that every developer, administrator, and user should know.

1. Authentication and Authorization

Authentication is the process of verifying the identity of a user, while authorization is the process of determining what a user can access once they are authenticated. Proper authentication and authorization are crucial to web security as they prevent unauthorized access to sensitive information. Authentication can be achieved using various methods, such as passwords, biometric authentication, or two-factor authentication. Authorization can be achieved by implementing role-based access control (RBAC), which grants access to resources based on a user's role.

2. Secure Communication

Communication between a user and a web server should be encrypted to prevent eavesdropping and man-in-the-middle attacks. Hypertext Transfer Protocol Secure (HTTPS) is a secure version of HTTP that encrypts data in transit using Transport Layer Security (TLS) or Secure Sockets Layer (SSL). SSL/TLS certificates can be obtained from trusted certificate authorities and should be configured correctly to prevent errors or vulnerabilities.

3. Input Validation

Input validation is the process of verifying that user input is valid and safe to use. Malicious input, such as SQL injections or cross-site scripting (XSS) attacks, can cause serious damage to a website or web application. Input validation can be achieved by using input filters and limiting the input types and length.

4. Cross-Site Scripting (XSS)

XSS attacks occur when a website or web application includes untrusted data in a web page without proper validation or encoding. This allows an attacker to inject malicious scripts, which can steal user data or redirect users to malicious websites. Preventing XSS attacks can be achieved by encoding user input, sanitizing user input, and using content security policy (CSP) headers.

5. **Cross-Site Request Forgery (CSRF)**

CSRF attacks occur when an attacker tricks a user into performing an action they did not intend to perform, such as clicking a malicious link or submitting a form on a malicious website. CSRF attacks can be prevented by using a CSRF token, which is a unique identifier that is attached to every form submission or request.

6. **SQL Injection**

SQL injections occur when an attacker injects malicious SQL statements into a website or web application's database query, which allows them to access sensitive information or modify the database. Preventing SQL injections can be achieved by using parameterized queries, input validation, and avoiding dynamic SQL queries[1], [2].

7. **Secure File Uploads**

File uploads are a common feature of web applications, but they can also be used to upload malicious files that can compromise the security of the website or web application. Preventing file upload vulnerabilities can be achieved by verifying the file type, limiting the file size, and storing uploaded files outside the web root directory.

8. **Password Security**

Passwords are the most common form of authentication, and they should be stored securely to prevent unauthorized access to user accounts. Password security can be improved by enforcing strong password policies, using password hashing algorithms, and using multi-factor authentication.

9. **Server-Side Security**

Server-side security is crucial to web security, as it protects the server and the web application from attacks. Server-side security can be achieved by updating software regularly, using firewalls, and implementing intrusion detection and prevention systems.

10. **User Education**

Users play a crucial role in web security, as they are often the first line of defense against attacks. User education can be achieved by providing security awareness training, implementing strong password policies, and encouraging users to report suspicious activity.

Wireless Local Area Network (WLAN) is a type of computer network that allows devices to connect wirelessly to a local network or the internet. WLAN is a popular alternative to wired networks as it provides flexibility and mobility to users. In this article, we will discuss the technical aspects of WLAN, its benefits, and its security concerns.

1. **WLAN Architecture**

A WLAN typically consists of a wireless access point (AP) and wireless clients. The AP serves as a central point of communication and provides wireless connectivity to the clients. WLAN clients can be any device that supports Wi-Fi, such as laptops, smartphones, or tablets. WLAN

can also be extended through the use of range extenders or mesh networks, which provide additional coverage and capacity.

2. WLAN Standards

There are several WLAN standards, including 802.11a, 802.11b, 802.11g, 802.11n, 802.11ac, and 802.11ax. These standards differ in their frequency bands, data rates, and features. The most commonly used WLAN standard is 802.11n, which supports data rates of up to 600 Mbps and is compatible with most devices.

3. WLAN Frequency Bands

WLAN operates in two frequency bands: 2.4 GHz and 5 GHz. The 2.4 GHz band is a crowded band that is shared with other devices, such as Bluetooth and microwaves, which can cause interference and reduce performance. The 5 GHz band is less crowded and provides faster data rates, but has a shorter range than the 2.4 GHz band.

4. WLAN Security

WLAN security is a critical aspect of WLAN as it protects the network from unauthorized access and data breaches. The following are the most common WLAN security measures:

a. Wired Equivalent Privacy (WEP)

WEP is an outdated security protocol that is no longer recommended. WEP uses a shared key encryption method, which is easy to crack and can be compromised within minutes.

b. Wi-Fi Protected Access (WPA)

WPA is a security protocol that uses the Advanced Encryption Standard (AES) encryption method and a Pre-Shared Key (PSK) for authentication. WPA is more secure than WEP, but it can still be vulnerable to attacks.

c. Wi-Fi Protected Access II (WPA2)

WPA2 is the most widely used security protocol for WLAN. WPA2 uses AES encryption and a more secure authentication method, such as 802.1X or RADIUS. WPA2 is more secure than WPA, but it can still be vulnerable to attacks.

d. Wi-Fi Protected Access III (WPA3)

WPA3 is the latest security protocol for WLAN. WPA3 provides stronger encryption and authentication methods, such as Simultaneous Authentication of Equals (SAE) and 192-bit encryption keys. WPA3 is more secure than WPA2, but it may not be compatible with all devices.

5. WLAN Management

WLAN management involves the configuration, monitoring, and maintenance of WLAN networks. WLAN management can be performed using a WLAN controller, which is a central

device that manages and controls the access points. WLAN management can also be performed using cloud-based WLAN management software, which provides a web-based interface for managing and monitoring WLAN networks.

6. WLAN Benefits

WLAN provides several benefits, including:

a. Mobility and Flexibility

WLAN allows users to connect to the network wirelessly from anywhere within the coverage area. This provides users with mobility and flexibility to work from anywhere.

b. Scalability

WLAN can be easily expanded by adding more access points or range extenders, providing additional coverage and capacity.

c. Cost-effective

WLAN is often more cost-effective than wired networks

7. WLAN Deployment

WLAN deployment involves the planning, installation, and configuration of the WLAN network. The following are the key steps in WLAN deployment:

a. Site Survey

A site survey is the first step in WLAN deployment. A site survey involves a survey of the site to determine the coverage area, the number of access points required, and the optimal location for access points.

b. Access Point Installation

Access points should be installed in optimal locations to ensure proper coverage and performance. Access points should be mounted on ceilings or walls and should be placed in areas where there is minimal interference.

c. Access Point Configuration

Access points should be configured to optimize performance and security. Access point configuration involves setting up the SSID, security settings, and network parameters.

d. Client Configuration

WLAN clients should be configured to connect to the WLAN network. Client configuration involves setting up the SSID, security settings, and network parameters.

8. WLAN Interference

WLAN interference is a common issue that can affect WLAN performance. The following are the most common types of WLAN interference:

a. Co-Channel Interference

Co-channel interference occurs when multiple access points are configured to use the same channel. This can cause interference and reduce performance.

b. Adjacent-Channel Interference

Adjacent-channel interference occurs when access points are configured to use adjacent channels. This can cause interference and reduce performance.

c. Microwave Interference

Microwave interference can cause interference and reduce performance. Microwave ovens operate in the 2.4 GHz frequency band, which is the same frequency band used by WLAN.

d. Bluetooth Interference

Bluetooth devices can cause interference and reduce performance. Bluetooth devices operate in the 2.4 GHz frequency band, which is the same frequency band used by WLAN.

9. WLAN Performance

WLAN performance can be affected by several factors, including signal strength, interference, and network congestion. The following are the most common factors that affect WLAN performance:

a. Signal Strength

Signal strength is a critical factor in WLAN performance. Access points should be installed in optimal locations to ensure proper coverage and signal strength.

b. Interference

Interference can cause WLAN performance to degrade. Interference can be caused by other wireless networks, Bluetooth devices, or other wireless devices.

c. Network Congestion

Network congestion can cause WLAN performance to degrade. Network congestion can be caused by too many users connecting to the network or by too much data traffic.

d. Bandwidth Limitations

WLAN performance can be limited by bandwidth limitations. WLAN bandwidth limitations can be caused by the number of users on the network, the amount of data traffic, or the type of data traffic.

10. WLAN Future

The future of WLAN is expected to see continued growth and development. The following are the key trends in WLAN technology:

a. 802.11ax

802.11ax is the latest WLAN standard, which is designed to provide faster data rates and better performance. 802.11ax is expected to be widely adopted in the coming years.

b. Internet of Things (IoT)

The growth of IoT is expected to drive the adoption of WLAN. WLAN will be a critical component in the deployment of IoT devices and applications.

c. Cloud-based WLAN Management

Cloud-based WLAN management is expected to grow in popularity. Cloud-based WLAN management provides a web-based interface for managing and monitoring WLAN networks.

d. 5G

The growth of 5G is expected to drive the adoption of WLAN. WLAN will be a critical component in the deployment of 5G networks and applications.

WLAN is a popular alternative to wired networks as it provides flexibility and mobility to users. WLAN provides several benefits, including mobility, scalability, and cost-effectiveness. WLAN deployment, and WLAN interference can affect WLAN performance. WLAN performance can be affected by several factors, including signal strength, interference, network congestion, and bandwidth limitations. The future of WLAN is expected to see continued growth and development, with the adoption of new technologies such as 802.11ax, IoT, cloud-based WLAN management, and 5G[3], [4].

In summary, WLAN is a popular technology that provides mobility, flexibility, and cost-effectiveness to users. However, WLAN security, deployment, and interference can affect WLAN performance. To ensure optimal WLAN performance, WLAN networks should be properly planned, installed, and configured. WLAN security measures should also be implemented to protect WLAN networks from security threats. With the continued development of new WLAN technologies, the future of WLAN looks bright, and WLAN is expected to play a critical role in the deployment of emerging technologies such as IoT and 5G.Some additional points to consider in regards to WLAN:

WLAN Best Practices

To ensure optimal WLAN performance and security, the following best practices should be considered:

a. Limit Network Access

Limiting network access is an important aspect of WLAN security. Access to the WLAN network should be limited to authorized users only.

b. Implement Network Segmentation

Network segmentation is another important aspect of WLAN security. Segmented networks provide an added layer of security by limiting access to certain areas of the network.

c. Use Encryption

Encryption is a critical aspect of WLAN security. Encryption should be used to protect WLAN traffic from interception and eavesdropping.

d. Use Strong Authentication

Strong authentication is another important aspect of WLAN security. Strong authentication mechanisms should be used to prevent unauthorized access to the WLAN network.

e. Regularly Update WLAN Firmware and Software

Regularly updating WLAN firmware and software is important to ensure that the WLAN network is secure and functioning optimally.

WLAN Security Risks

WLAN networks are susceptible to several security risks, including:

a. Rogue Access Points

Rogue access points are unauthorized access points that can be used to gain access to the WLAN network. Rogue access points can be used to steal sensitive information or launch attacks on the network.

b. Man-in-the-Middle Attacks

Man-in-the-middle attacks occur when an attacker intercepts and alters data being transmitted over the WLAN network. Man-in-the-middle attacks can be used to steal sensitive information or launch attacks on the network[5], [6].

c. Denial of Service (DoS) Attacks

DoS attacks occur when an attacker floods the WLAN network with traffic, causing the network to crash or become unusable. DoS attacks can be used to disrupt network operations or launch attacks on the network.

d. Eavesdropping

Eavesdropping occurs when an attacker intercepts WLAN traffic and listens in on communications between WLAN clients and access points. Eavesdropping can be used to steal sensitive information or launch attacks on the network.

WLAN Security Measures

To protect WLAN networks from security threats, the following security measures should be implemented:

a. Authentication and Access Control

Authentication and access control mechanisms should be implemented to prevent unauthorized access to the WLAN network.

b. Encryption

Encryption should be used to protect WLAN traffic from interception and eavesdropping.

c. Intrusion Detection and Prevention

Intrusion detection and prevention mechanisms should be implemented to detect and prevent attacks on the WLAN network.

d. Security Audits and Testing

Regular security audits and testing should be conducted to identify security vulnerabilities and address them before they can be exploited.

WLAN Management

WLAN management involves the monitoring and control of the WLAN network. The following are the key aspects of WLAN management:

a. Performance Monitoring

Performance monitoring involves monitoring WLAN performance to identify and resolve performance issues.

b. Configuration Management

Configuration management involves managing WLAN network configuration to ensure optimal performance and security[7].

c. Fault Management

Fault management involves detecting and resolving faults in the WLAN network.

d. Security Management

Security management involves managing WLAN security to prevent security breaches and attacks on the network.

WLAN and the Internet of Things (IoT)

WLAN is expected to play a critical role in the deployment of IoT devices and applications. WLAN will be used to connect IoT devices to the network and provide the necessary connectivity for IoT applications.

WLAN and 5G

WLAN is expected to play a critical role in the deployment of 5G networks and applications. WLAN will be used to provide connectivity to 5G devices and applications and to provide the necessary backhaul connectivity for 5G networks.

WLAN is a popular technology that provides mobility, flexibility, and cost-effectiveness to users. WLAN networks should be properly planned, installed, configured, and secured to ensure optimal performance and security[8], [9]. WLAN security is a critical aspect of WLAN deployment, and WLAN interference can affect WLAN performance. WLAN performance can be affected by several factors, including signal strength, interference, network congestion, and bandwidth limitations. The future of WLAN is expected to see continued growth and development, with the adoption of new technologies such as 802.11ax, IoT, cloud-based WLAN management, and 5G.

WLAN Standards

WLAN standards are a set of guidelines that define how WLAN networks should be designed, implemented, and secured. The following are some of the key WLAN standards:

a. IEEE 802.11

IEEE 802.11 is the most widely used WLAN standard. It defines the physical and data link layer specifications for WLANs.

b. IEEE 802.11a

IEEE 802.11a is a WLAN standard that operates in the 5 GHz frequency band. It provides higher data rates than IEEE 802.11b, but has a shorter range.

c. IEEE 802.11b

IEEE 802.11b is a WLAN standard that operates in the 2.4 GHz frequency band. It provides lower data rates than IEEE 802.11a, but has a longer range.

d. IEEE 802.11g

IEEE 802.11g is a WLAN standard that operates in the 2.4 GHz frequency band. It provides higher data rates than IEEE 802.11b, but has a shorter range.

e. IEEE 802.11n

IEEE 802.11n is a WLAN standard that provides higher data rates and better coverage than IEEE 802.11a, IEEE 802.11b, and IEEE 802.11g.

f. IEEE 802.11ac

IEEE 802.11ac is a WLAN standard that provides higher data rates and better coverage than IEEE 802.11n. It operates in the 5 GHz frequency band.

g. IEEE 802.11ax

IEEE 802.11ax is the latest WLAN standard. It provides higher data rates and better coverage than IEEE 802.11ac. It operates in the 2.4 GHz and 5 GHz frequency bands.

WLAN Deployment Models

WLAN can be deployed in several ways, depending on the organization's needs and requirements. The following are some of the key WLAN deployment models:

a. Centralized WLAN

Centralized WLAN is a deployment model in which WLAN access points are managed by a central controller. This model provides a high level of control and management of the WLAN network[10], [11].

b. Distributed WLAN

Distributed WLAN is a deployment model in which WLAN access points are managed by individual access points. This model provides a high level of scalability and flexibility.

c. Cloud-based WLAN

Cloud-based WLAN is a deployment model in which WLAN access points are managed through a cloud-based management platform. This model provides a high level of flexibility and scalability, and is ideal for organizations that need to manage WLAN networks across multiple locations.

WLAN Interference

WLAN interference can affect WLAN performance and cause connectivity issues. The following are some of the key sources of WLAN interference:

a. Other WLAN Networks

Other WLAN networks operating in the same frequency band can cause interference and affect WLAN performance.

b. Bluetooth Devices

Bluetooth devices can cause interference in the 2.4 GHz frequency band, which can affect WLAN performance.

c. Microwave Ovens

Microwave ovens can cause interference in the 2.4 GHz frequency band, which can affect WLAN performance.

d. Cordless Phones

Cordless phones can cause interference in the 2.4 GHz frequency band, which can affect WLAN pe rformance.

REFERENCES:

[1] A. G. A. Rasyid, "Perancangan Jaringan Rt / Rw Net Menggunakan Teknologi Wireless Lan (Studi Kasus : Rz Reload Connection)," *Fak. Tek. Unpas*, 2018.

[2] R. T. Jurnal, "PERANCANGAN KENDALI GARASI RUMAH BERBASIS WEB VIA WIRELESS LAN," *Sutet*, 2018, doi: 10.33322/sutet.v7i2.80.

[3] B. Karthik, S. P. Vijayaragavan, and M. Sriram, "Microstrip patch antenna for wireless LAN," *Int. J. Pure Appl. Math.*, 2018.

[4] H. Goto, "Cityroam, Providing Secure Public Wireless LAN Services with International Roaming," in *Proceedings - 2018 Advances in Wireless and Optical Communications, RTUWO 2018*, 2018. doi: 10.1109/RTUWO.2018.8587899.

[5] S. Prasad and N. Lakshmi, "Architecture of Optimized Wireless LAN," *Int. J. Mod. Trends Sci. Technol.*, 2018.

[6] T. Jamal *et al.*, "Denial of Service Attack in Wireless LAN," *12th Int. Conf. Digit. Soc. eGovernments*, 2018.

[7] L. Wang, S. Jajodia, A. Singhal, P. Cheng, and S. Noel, "K-zero day safety: A network security metric for measuring the risk of unknown vulnerabilities," *IEEE Trans. Dependable Secur. Comput.*, 2014, doi: 10.1109/TDSC.2013.24.

[8] X. He, T. Chomsiri, P. Nanda, and Z. Tan, "Improving cloud network security using the Tree-Rule firewall," *Futur. Gener. Comput. Syst.*, 2014, doi: 10.1016/j.future.2013.06.024.

[9] L. R. Bays, R. R. Oliveira, M. P. Barcellos, L. P. Gaspary, and E. R. Mauro Madeira, "Virtual network security: threats, countermeasures, and challenges," *J. Internet Serv. Appl.*, 2015, doi: 10.1186/s13174-014-0015-z.

[10] J. Ahn, Y. Y. Kim, and R. Y. Kim, "Virtual reality-wireless local area network: Wireless connection-oriented virtual reality architecture for next-generation virtual reality devices," *Appl. Sci.*, 2018, doi: 10.3390/app8010043.

[11] S. Roy, K. L. Baishnab, and U. Chakraborty, "Beam focusing compact wideband antenna loaded with mu-negative metamaterial for wireless LAN application," *Prog. Electromagn. Res. C*, 2018, doi: 10.2528/pierc18012908.

CHAPTER 16

WIRELESS APPLICATION PROTOCOL

Dr. Lokesh Kumar, Assistant Professor
Department of Computer Science and Engineering, Sanskriti University, Mathura, Uttar Pradesh, India
Email Id- lokesh@sanskriti.edu.in

The Wireless Application Protocol (WAP) is a universal, open standard developed by the WAP Forum to provide mobile users of wireless phones and other wireless terminals such as pagers and personal digital assistants (PDAs) access to telephony and information services, including the Internet and the Web. WAP is designed to work with all wireless network technologies (e.g., GSM, CDMA, and TDMA). WAP is based on existing Internet standards, such as IP, XML, HTML, and HTTP, as much as possible. It also includes security facilities. At the time of this writing, the current release of the WAP specification is version 2.0. Strongly affecting the use of mobile phones and terminals for data services are the significant limitations of the devices and the networks that connect them. The devices have limited processors, memory, and battery life. The user interface is also limited, and the displays small. The wireless networks are characterized by relatively low bandwidth, high latency, and unpredictable availability and stability compared to wired connections.

Moreover, all of these features vary widely from terminal device to terminal device and from network to network. Finally, mobile, wireless users have different expectations and needs from other information systems users. For instance, mobile terminals must be extremely easy to use — much easier than workstations and personal computers. WAP is designed to deal with these challenges. The WAP specification includes:

1. A programming model based on the WWW Programming Model
2. A markup language, the Wireless Markup Language, adhering to XML
3. A specification of a small browser suitable for a mobile, wireless terminal
4. A lightweight communications protocol stack
5. A framework for wireless telephony applications (WTAs) HMAC-SHA-1 | | K A 0 B i R = HMAC-SHA-1(K, A || 0 || B || i) + 1

The WAP Programming Model is based on three elements: the client, the gateway, and the original server. HTTP is used between the gateway and the original server to transfer content. The gateway acts as a proxy server for the wireless domain. Its processor(s) provide services that offload the limited capabilities of the hand-held, mobile, wireless terminals. For example, the gateway provides DNS services, converts between WAP protocol stack and the WWW stack (HTTP and TCP/IP), encodes information from the Web into a more compact form that minimizes wireless communication, and in the other direction, decodes the compacted form into standard Web communication conventions[1].

The gateway also caches frequently requested information. Figure 6.12 illustrates key components in a WAP environment. Using WAP, a mobile user can browse Web content on an

ordinary Web server. The Web server provides content in the form of HTML-coded pages that are transmitted using the standard Web protocol stack (HTTP/TCP/IP). The HTML content must go through an HTML filter, which either may be colocated with the WAP proxy or in a separate physical module.The filter translates the HTML content into WML content. If the filter is separate from the proxy, HTTP/TCP/IP is used to deliver the WML to the proxy. The proxy converts the WML to a more compact form known as binary WML and delivers it to the mobile user over a wireless network using the WAP protocol stack. If the Web server is capable of directly generating WML content, then the WML is delivered using HTTP/TCP/IP to the proxy, which converts the WML to binary WML and then delivers it to the mobile node using WAP protocols.

The WAP architecture is designed to cope with the two principal limitations of wireless Web access: the limitations of the mobile node (small screen size, limited input capability) and the low data rates of wireless digital networks. Even with the introduction of 3G wireless networks, which provide broadband data rates, the small hand-held mobile nodes continue to have limited input and display capabilities. Thus, WAP or a similar capability will be needed for the indefinite future. Wireless Markup Language WML was designed to describe content and format for presenting data on devices with limited bandwidth, limited screen size, and limited user input capability. It is designed to work with telephone keypads, styluses, and other input devices common to mobile, wireless communication.

WML permits the scaling of displays for use on two-line screens found in some small devices,as well as the larger screens found on smart phones. WAE User Agent Encoders and Decoders CGI Scripts etc. Encoded requests Client Gateway Original Server Requests Encoded response Response (content).

WAP Infrastructure For an ordinary PC, a Web browser provides content in the form of Web pages coded with the Hypertext Markup Language (HTML). To translate an HTMLcoded Web page into WML with content and format suitable for wireless devices, much of the information, especially graphics and animation, must be stripped away. WML presents mainly text-based information that attempts to capture the essence of the Web page and that is organized for easy access for users of mobile devices. Important features of WML include:

A. **Text and image support:** Formatting and layout commands are provided for text and limited image capability.
B. **Deck/card organizational metaphor:** WML documents are subdivided into small, well-defined units of user interaction called cards. Users navigate by moving back and forth between cards.

A card specifies one or more units of interaction (a menu, a screen of text, or a text-entry field). A WML deck is similar to an HTML page in that it is identified by a Web address (URL) and is the unit of content transmission. Support for navigation among cards and decks: WML includes provisions for event handling, which is used for navigation or executing scripts. In an HTML-based Web browser, a user navigates by clicking on links. At a WML-capable mobile device, a user interacts with cards, moving forward and back through the deck. WAP Architecture Figure

6.13, from the WAP architecture document, illustrates the overall stack architecture implemented in a WAP client. In essence, this is a five-layer model. Each layer provides a set of functions and/or services to other services and applications through a set of well-defined interfaces. Each of the layers of the architecture is accessible by the layers above, as well as by other services and applications. Many of the services in the stack may be provided by more than one protocol. For example, either HTTP or WSP may provide the Hypermedia Transfer service. Common to all five layers are sets of services that are accessible by multiple layers.

These common services fall into two categories: security services and service discovery. Service Discovery Security Crypto libraries Auth. Identity PKI Secure transport Secure bearer EFI Provisioning Navigation Discovery Service lookup Push Content Formats WEA/WTA user agent(s) Multimedia messaging Push-OTA Synchronisation Cookies Capability negotiation Session Services Application Framework Protocol Framework IPv6 IPv4 MPAK SOS ReFLEX FLEX GUTS GHOST USSD SMS Bearer Networks Datagrams Transport Services Hypermedia transfer Streaming Message transfer Transfer Services Connections Figure 6.13 WAP Architecture 6.3 / WIRELESS APPLICATION PROTOCOL OVERVIEW 201 SECURITY SERVICES The WAP specification includes mechanisms to provide confidentiality, integrity, authentication, and nonrepudiation.

The security services include the following.

1. **Cryptographic libraries:** This application framework level library provides services for signing of data for integrity and non-repudiation purposes.
2. **Authentication:** WAP provides various mechanisms for client and server authentication. At the Session Services layer, HTTP Client Authentication (RFC2617) may be used to authenticate clients to proxies and application servers. At the Transport Services layer, WTLS and TLS handshakes may be used to authenticate clients and servers.
3. **Identity:** The WAP Identity Module (WIM) provides the functions that store and process information needed for user identification and authentication.
4. **PKI:** The set of security services that enable the use and management of public-key cryptography and certificates.
5. **Secure transport:** The Transport Services layer protocols are defined for secure transport over datagrams and connections.

WTLS is defined for secure transport over datagrams and TLS is defined for secure transport over connections (i.e., TCP).

A. **Secure bearer:** Some bearer networks provide bearer-level security. For example, IP networks (especially in the context of IPv6) provide bearer-level security with IPsec. SERVICE DISCOVERY There is a collection of service discovery services that enable the WAP client and the Web server to determine capabilities and services. Examples of service discovery services include the following.

B. **EFI:** The External Functionality Interface (EFI) allows applications to discover what external functions/services are available on the device.
C. **Provisioning:** This service allows a device to be provisioned with the parameters necessary to access network services.
D. **Navigation discovery:** This service allows a device to discover new network services (e.g., secure pull proxies) during the course of navigation such as when downloading resources from a hypermedia server. The WAP Transport Level End-to-End Security specification, described in Section 6.5, defines one navigation discovery protocol.
E. **Service lookup:** This service provides for the discovery of a service's parameters through a directory lookup by name. One example of this is the Domain Name System (DNS). Wireless Application Environment The WAE specifies an application framework for wireless devices such as mobile telephones, pagers, and PDAs[2], [3].

The major elements of the WAE model are:

WAE user agents: Software that executes in the user's wireless device and that provides specific functionality (e.g., display content) to the end user.

1. **Wireless telephony applications (WTA):** A collection of telephony-specific extensions for call and feature control mechanisms that provide authors advanced mobile network services. Using WTA, applications developers can use the microbrowser to originate telephone calls and to respond to events from the telephone network.
2. **Standard content encoding:** Defined to allow a WAE user agent (e.g., a browser) to conveniently navigate Web content. On the server side are content generators. These are applications (or services) on origin servers (e.g., CGI scripts) that produce standard content formats in response to requests from user agents in the mobile terminal. WAE does not specify any standard content generators but expects that there will be a variety available running on typical HTTP origin servers commonly used in WWW today.
3. **Push:** A service to receive push transmissions from the server, i.e., transmissions that are not in response to a Web client request but are sent on the initiative of the server. This service is supported by the Push-OTA (Push over the Air) session service.
4. **Multimedia messaging:** Provides for the transfer and processing of multimedia messages, such as e-mail and instant messages, to WAP devices. WAP Protocol Architecture

A collection of services at each level and provides interface specifications at the boundary between each pair of layers. Because several of the services in the WAP stack can be provided using different protocols based on the circumstances, there are more than one possible stack configurations. A common protocol stack configuration in which a WAP client device connects to a Web server via a WAP gateway.

This configuration is common with devices that implement version 1 of the WAP specification but is also used in version 2 devices (WAP2) if the bearer network does not support TCP/IP. Bearer In the remainder of this subsection, we provide an overview of the WAP protocols, with the exception of WTLS. WIRELESS SESSION PROTOCOL WSP provides applications with an interface for two session services.The connection-oriented session service operates above WTP, and the connectionless session service operates above the unreliable transport protocol WDP. In essence,WSP is based on HTTP with some additions and modifications to optimize its use over wireless channels. The principal limitations addressed are low data rate and susceptibility to loss of connection due to poor coverage or cell overloading. WSP is a transaction-oriented protocol based on the concept of a request and a reply. Each WSP protocol data unit (PDU) consists of a body, which may contain WML, WMLScript, or images; and a header, which contains information about the data in the body and about the transaction. WSP also defines a server push operation, in which the server sends unrequested content to a client device. This may be used for broadcast messages or for services, such as news headlines or stock quotes, that may be tailored to each client device.

WIRELESS TRANSACTION PROTOCOL

WTP manages transactions by conveying requests and responses between a user agent (such as a WAP browser) and an application server for such activities as browsing and e-commerce transactions. WTP provides a reliable transport service but dispenses with much of the overhead of TCP, resulting in a lightweight protocol that is suitable for implementation in “thin” clients (e.g., mobile nodes) and suitable for use over low-bandwidth wireless links. WTP includes the following features. Three classes of transaction service. Optional user-to-user reliability: WTP user triggers the confirmation of each received message. Optional out-of-band data on acknowledgments. PDU concatenation and delayed acknowledgment to reduce the number of messages sent. Asynchronous transactions. WTP is transaction oriented rather than connection oriented.With WTP, there is no explicit connection setup or teardown but rather a reliable connectionless service.

WTP provides three transaction classes that may be invoked by WSP or another higher layer protocol:

Class 0: Unreliable invoke message with no result message
Class 1: Reliable invoke message with no result message
Class 2: Unreliable invoke message with one reliable result message

Class 0 provides an unreliable datagram service, which can be used for an unreliable push operation. Data from a WTP user are encapsulated by WTP (the initiator, or client) in an invoke PDU and transmitted to the target WTP (the responder, or server) with no acknowledgment. The responder WTP delivers the data to the target WTP user. Class 1 provides a reliable datagram service, which can be used for a reliable push operation. Data from an initiator are encapsulated in an invoke PDU and transmitted to the responder.

The responder delivers the data to the target WTP user and acknowledges receipt of the data by sending back an ACK PDU to the WTP entity on the initiator side, which confirms the

transaction to the source WTP user. The responder WTP maintains state information for some time after the ACK has been sent to handle possible retransmission of the ACK if it gets lost and/or the initiator retransmits the invoke PDU. Class 2 provides a request/response transaction service and supports the execution of multiple transactions during one WSP session. Data from an initiator are encapsulated in an invoke PDU and transmitted to the responder, which delivers the data to the target WTP user. The target WTP user prepares response data, which are handed down to the local WTP entity.The responder WTP entity sends these data back in a result PDU. If there is a delay in generating the response data beyond a timer threshold, the responder may send an ACK PDU before sending the result PDU.This prevents the initiator from unnecessarily retransmitting the invoke message. WIRELESS DATAGRAM PROTOCOL WDP is used to adapt a higher-layer WAP protocol to the communication mechanism called the bearer used between the mobile node and the WAP gateway. Adaptation may include partitioning data into segments of appropriate size for the bearer and interfacing with the bearer network. WDP hides details of the various bearer networks from the other layers of WAP. In some instances, WAP is implemented on top of IP. 6.4 WIRELESS TRANSPORT LAYER SECURITY WTLS provides security services between the mobile device (client) and the WAP gateway. WTLS is based on the industry-standard Transport Layer Security (TLS) Protocol,which is a refinement of the Secure Sockets Layer (SSL) protocol. TLS is the standard security protocol used between Web browsers and Web servers. WTLS is more efficient that TLS, requiring fewer message exchanges[4], [5]. To provide end-toend security, WTLS is used between the client and the gateway, and TLS is used between the gateway and the target server (Figure 6.14). WAP systems translate between WTLS and TLS within the WAP gateway. Thus, the gateway is a point of vulnerability and must be given a high level of security from external attacks. WTLS provides the following features.

A. Data integrity: Uses message authentication to ensure that data sent between the client and the gateway are not modified.
B. Privacy: Uses encryption to ensure that the data cannot be read by a third party.
C. Authentication: Uses digital certificates to authenticate the two parties.
D. Denial-of-service protection: Detects and rejects messages that are replayed or not successfully verified.

However, the discussion in this section is self-contained; you do not need to read a description of TLS first. Sessions and Connections Two important WTLS concepts are the secure session and the secure connection, which are defined in the specification as:

1. **Secure connection:** A connection is a transport (in the OSI layering model definition) that provides a suitable type of service. For SSL, such connections are peer-to-peer relationships. The connections are transient. Every connection is associated with one session.
2. **Secure session:** An SSL session is an association between a client and a server. Sessions are created by the Handshake Protocol. Sessions define a

set of cryptographic security parameters, which can be shared among multiple connections. Sessions are used to avoid the expensive negotiation of new security parameters for each connection.

Between any pair of parties applications such as HTTP on client and server, there may be multiple secure connections. In theory, there may also be multiple simultaneous sessions between parties, but this feature is not used in practice. There are a number of states associated with each session.

Once a session is established, there is a current operating state for both read and write (i.e., receive and send). In addition, during the Handshake Protocol, pending read and write states are created. Upon successful conclusion of the Handshake Protocol, the pending states become the current states. A session state is defined by the following parameters:

A. Session identifier: An arbitrary byte sequence chosen by the server to identify an active or resumable session state.
B. Protocol version: WTLS protocol version number.
C. Peer certificate: Certificate of the peer. This element of the state may be null.
D. Compression method: The algorithm used to compress data prior to encryption.
E. Cipher spec: Specifies the bulk data encryption algorithm (such as null, RC5, DES, etc.) and a hash algorithm (such as MD5 or SHA-1) used for MAC calculation. It also defines cryptographic attributes such as the hash size.
F. Master secret: A 20-byte secret shared between the client and server.
G. Sequence number: Which sequence numbering scheme (off, implicit, or explicit) is used in this secure connection.
H. Key refresh: Defines how often some connection state values (encryption key, MAC secret, and IV) calculations are performed.
I. Is resumable: A flag indicating whether the session can be used to initiate new connections.

The connection state is the operating environment of the record protocol. It includes all parameters that are needed for the cryptographic operations (encryption/decryption and MAC calculation/verification)[6], [7].Each secure connection has a connection state, which is defined by the following parameters

- Connection end: Whether this entity is considered a client or a server in this secure session.
- Bulk cipher algorithm: Includes the key size of this algorithm, how much of that key is secret, whether it is a block or stream cipher, and the block size of the cipher (if appropriate).
- MAC algorithm: Includes the size of the key used for MAC calculation and the size of the hash which is returned by the MAC algorithm.
- Compression algorithm: Includes all information the algorithm requires to do compression.
- Master secret: A 20-byte secret shared between the client and server.

- Client random: A 16-byte value provided by the client.
- Server random: A 16-byte value provided by the server.
- Sequence number mode: Which scheme is used to communicate sequence numbers in this secure connection?
- Key refresh: Defines how often some connection state parameters (encryption key, MAC secret, and IV) are updated.

New keys are calculated at every messages, that is, when the sequence number is 0, , , etc. WTLS Protocol Architecture WTLS is not a single protocol but rather two layers of protocols. The WTLS Record Protocol provides basic security services to various higher-layer protocols. In particular, the Hypertext Transfer Protocol (HTTP), which provides the transfer service for Web client/server interaction, can operate on top of WTLS. Three higher-layer protocols are defined as part of WTLS: the Handshake Protocol, The Change Cipher Spec Protocol, and the Alert Protocol.

These WTLS-specific protocols are used in the management of WTLS exchanges and are examined subsequently in this section. WTLS RECORD PROTOCOL The WTLS Record Protocol takes user data from the next higher layer (WTP, WTLS Handshake Protocol, WTLS Alert Protocol, and WTLS Change Cipher Spec Protocol) and encapsulates these data in a PDU.

The following steps occur (Figure 6.16). 3n 2n n = 2key_refresh WDP or UDP/IP WTLS Record Protocol WTLS Handshake Protocol WTLS Change Cipher Spec Protocol WTLS Alert Protocol WTP Figure 6.15 WTLS Protocol Stack 6.4 / WIRELESS TRANSPORT LAYER SECURITY 207 Step 1. The payload is compressed using a lossless compression algorithm. Step 2. A message authentication code (MAC) is computed over the compressed data, using HMAC. One of several hash algorithms can be used with HMAC, including MD-5 and SHA-1.

The length of the hash code is 0, 5, or 10 bytes. The MAC is added after the compressed data. Step 3. The compressed message plus the MAC code are encrypted using a symmetric encryption algorithm.

The allowable encryption algorithms are DES, triple DES, RC5, and IDEA. Step 4. The Record Protocol prepends a header to the encrypted payload. The Record Protocol header consists of the following fields (Figure 6.17).

A. **Record type (8 bits):** Consisting of the subfields: –Record length field indicator (1 bit): Indicates whether a record length field is present. –Sequence number field indicator (1 bit): Indicates whether a sequence number field is present. –Cipher spec indicator (1 bit): If this bit is zero, it indicates that no compression, MAC protection, or encryption is used. –Content type (4 bits): The higher-layer protocol above the WTLS Record Protocol.
B. **Sequence number (16 bits):** A sequence number associated with this record. This provides reliability over an unreliable transport service.
C. **Record length (16 bits):** The length in bytes of the plaintext data (or compressed data if compression is used).

CHANGE CIPHER SPEC PROTOCOL

Associated with the current transaction is a cipher spec, which specifies the encryption algorithm, the hash algorithm used as User Data Compress Add MAC Encrypt Append WTLS Record Header WTLS Record Protocol Operation.There are two states associated with each session. Once a session is established, there is a current operating state for both read and write (i.e., receive and send). In addition, during the Handshake Protocol, pending read and write states are created. The Change Cipher Spec Protocol is one of the three WTLS-specific protocols that use the WTLS Record Protocol, and it is the simplest. This protocol consists of a single message, which consists of a single byte with the value 1.

The sole purpose of this message is to cause the pending state to be copied into the current state, which updates the cipher suite to be used on this connection. Thus, when the Change Cipher Spec message arrives, the sender of the message sets the current write state to the pending state and the receiver sets the current read state to the pending state. ALERT PROTOCOL The Alert Protocol is used to convey WTLS-related alerts to the peer entity. As with other applications that use WTLS, alert messages are compressed and encrypted, as specified by the current state. r = reserved C = cipher spec indicator S = sequence number field indicator L = record length field indicator MAC = message authentication code Content type Sequence number Record length rCSL Plaintext (optionally compressed) MAC (0, 16, or 20 bytes) Encrypted Scope of MAC.

If the level is fatal, WTLS immediately terminates the connection. Other connections on the same session may continue, but no new connections on this session may be established. A critical alert message results in termination of the current secure connection. Other connections using the secure session may continue and the secure identifier may also be used for establishing new secure connections. The connection is closed using the alert messages. Either party may initiate the exchange of the closing messages. If a closing message is received, then any data after this message is ignored. It is also required that the notified party verifies termination of the session by responding to the closing message[8]. Error handling in the WTLS is based on the alert messages. When an error is detected, the detecting party sends an alert message containing the occurred error. Further procedures depend on the level of the error that occurred[9], [10].

This protocol allows the server and client to authenticate each other and to negotiate an encryption and MAC algorithms and cryptographic keys to beWireless Application Protocol (WAP) is a standard protocol for accessing the internet on mobile devices, such as smartphones and tablets, over wireless networks. It was introduced in the late 1990s as a way to bring internet services to mobile devices, which had limited computing power and bandwidth at the time. WAP works by using a thin client architecture, where the mobile device sends requests to a WAP gateway, which in turn retrieves the requested information from the internet and sends it back to the device in a format optimized for mobile viewing. WAP uses a markup language called Wireless Markup Language (WML), which is similar to HTML but is designed to work on smaller screens and slower connections. WAP was widely used in the early days of mobile internet, but has since been largely superseded by other protocols, such as HTML5 and native mobile apps. However, some legacy systems may still use WAP, and it remains an important part of mobile internet history.

REFERENCES:

[1] S. A. Alabady, F. Al-Turjman, and S. Din, "A Novel Security Model for Cooperative Virtual Networks in the IoT Era," *Int. J. Parallel Program.*, 2020, doi: 10.1007/s10766-018-0580-z.

[2] K. Sivaraman and G. Kavitha, "Mobile multi media opportunities and challengers," *Eurasian J. Anal. Chem.*, 2018.

[3] M. A. N. Islamy, "PELAYANAN INFORMASI PERPUSTAKAAN BERBASIS MOBILE," *Tibanndaru J. Ilmu Perpust. dan Inf.*, 2018, doi: 10.30742/tb.v2i2.551.

[4] N. Kaur, S. Verma, and Kavita, "A survey of routing protocols in wireless sensor networks," *Int. J. Eng. Technol.*, 2018, doi: 10.14419/ijet.v7i4.12.21094.

[5] M. Ahmed, M. Salleh, and M. I. Channa, "Routing protocols based on protocol operations for underwater wireless sensor network: A survey," *Egyptian Informatics Journal*. 2018. doi: 10.1016/j.eij.2017.07.002.

[6] Y. Zhang, Y. Xiang, W. Wu, and A. Alelaiwi, "A variant of password authenticated key exchange protocol," *Futur. Gener. Comput. Syst.*, 2018, doi: 10.1016/j.future.2017.02.016.

[7] K. Sivaraman and G. Kavitha, "Services without network traffic," *Eurasian J. Anal. Chem.*, 2018.

[8] K. Cao, "Research and analysis on network security modeling," *Int. J. Secur. its Appl.*, 2016, doi: 10.14257/ijsia.2016.10.4.14.

[9] L. Novelli, L. Jorge, P. Melo, and A. Koscianski, "Application Protocols and Wireless Communication for IoT: A Simulation Case Study Proposal," in *2018 11th International Symposium on Communication Systems, Networks and Digital Signal Processing, CSNDSP 2018*, 2018. doi: 10.1109/CSNDSP.2018.8471765.

[10] B. N. Silva *et al.*, "Contiki: The Open Source Operating System for the Internet of Things," *Wirel. Networks*, 2018.

CHAPTER 17

WIRELESS TRANSPORT LAYER SECURITY

Dr. Himanshu Singh, Assistant Professor
Department of Computer Science and Engineering, Sanskriti University, Mathura, Uttar Pradesh, India
Email Id- himanshu.singh@sanskriti.edu.in

Wireless Transport Layer Security (WTLS) is a protocol designed to provide secure communication over wireless networks. WTLS is a part of the Wireless Application Protocol (WAP), which is a suite of protocols designed for accessing the internet from mobile devices. In this article, we will discuss WTLS in detail, including its features, architecture, and security mechanisms.

WTLS Features: WTLS provides the following features:

1. **Authentication:** WTLS provides mutual authentication between the client and server. This means that both parties can verify the identity of the other party.
2. **Encryption:** WTLS provides encryption of data transmitted between the client and server. This ensures that the data cannot be intercepted and read by unauthorized parties.
3. **Integrity:** WTLS ensures that the data has not been tampered with during transmission.
4. **Replay protection:** WTLS provides protection against replay attacks, where an attacker intercepts and replays a previously transmitted message.

i. **WTLS Architecture:** WTLS is designed to operate between the application layer and the network layer. It provides security services to the WAP application layer by interacting with the underlying network layer protocols. The WTLS architecture consists of two layers: the Record Layer and the Handshake Layer.

ii. **Record Layer:** The Record Layer is responsible for providing confidentiality, integrity, and replay protection for data transmitted between the client and server. It encrypts the data using a symmetric encryption algorithm and provides integrity protection using a message authentication code (MAC). The Record Layer also provides support for fragmenting and reassembling large messages.

iii. **Handshake Layer:** The Handshake Layer is responsible for the negotiation of security parameters between the client and server. It establishes a shared secret key for use by the Record Layer and performs mutual authentication between the client and server. The Handshake Layer uses a series of messages to negotiate the security parameters and exchange authentication information.

iv. **WTLS Security Mechanisms:** WTLS provides several security mechanisms to ensure the security of the communication. These mechanisms include:

1. **Cryptographic algorithms:** WTLS supports several cryptographic algorithms for encryption, key exchange, and message authentication. These include the RSA, Diffie-Hellman, and SHA-1 algorithms.
2. **Public key infrastructure (PKI):** WTLS uses a PKI to provide mutual authentication between the client and server. The server presents its digital certificate to the client, which verifies the authenticity of the server's identity. The client may also present its digital certificate to the server for authentication.
3. **Session Keys:** WTLS uses session keys for encryption and integrity protection of data transmitted between the client and server. Session keys are generated during the handshake process and are used only for the duration of the session.
4. **Certificate Revocation:** WTLS supports certificate revocation, which allows a certificate authority to revoke a digital certificate if it is found to be compromised or no longer valid.
5. **Message Sequencing:** WTLS provides protection against replay attacks by using a message sequence number. The message sequence number is included in each message and is used to detect and reject duplicate messages[1], [2].

Conclusion: WTLS is a protocol designed to provide secure communication over wireless networks. It provides authentication, encryption, integrity, and replay protection for data transmitted between the client and server. WTLS uses a PKI for mutual authentication, session keys for encryption and integrity protection, and message sequencing for protection against replay attacks. WTLS is an important protocol for securing wireless communication and is widely used in mobile devices such as smartphones and tablets.

Wireless communication has become a ubiquitous part of modern life, with smartphones, tablets, and other mobile devices becoming increasingly important for communication, commerce, and entertainment. However, wireless networks are inherently less secure than wired networks due to the ease with which radio waves can be intercepted and the vulnerability of wireless devices to attacks. To address this problem, several security protocols have been developed to provide secure communication over wireless networks, including Wireless Transport Layer Security (WTLS).

WTLS was developed as part of the Wireless Application Protocol (WAP), a suite of protocols designed to enable mobile devices to access the internet. WTLS was designed to provide security services to the WAP application layer, which is responsible for providing web content and other services to mobile devices. WTLS was based on the Transport Layer Security (TLS) protocol, which is widely used for secure communication over the internet.

WTLS provides several security features, including authentication, encryption, integrity, and replay protection. Authentication is provided through the use of digital certificates, which are used to verify the identity of the client and server. The client and server exchange digital certificates during the handshake process, which allows each party to authenticate the other. Encryption is provided using symmetric encryption algorithms such as Advanced Encryption

Standard (AES), which ensures that the data transmitted between the client and server is protected from interception by unauthorized parties. Integrity is provided using message authentication codes (MACs), which are used to detect if the data has been tampered with during transmission. Replay protection is provided by using a message sequence number, which ensures that duplicate messages are detected and rejected.

WTLS uses a two-layer architecture consisting of the Handshake Layer and the Record Layer. The Handshake Layer is responsible for negotiating the security parameters between the client and server, including the encryption algorithm and key exchange algorithm. The Handshake Layer also performs mutual authentication between the client and server, which ensures that both parties can verify the identity of the other. The Record Layer is responsible for providing the encryption, integrity, and replay protection services for data transmitted between the client and server. The Record Layer uses the session keys generated during the handshake process to encrypt and authenticate the data.

WTLS also provides support for certificate revocation, which allows a certificate authority (CA) to revoke a digital certificate if it is found to be compromised or no longer valid. This is important for maintaining the security of the system, as compromised or invalid certificates can be used to launch attacks against the system.

In addition to the security features provided by WTLS, there are several best practices that can be used to improve the security of wireless communication. One of the most important best practices is to use strong passwords or passphrases for authentication. Weak passwords are vulnerable to brute-force attacks, where an attacker tries a large number of possible passwords in order to guess the correct one. Strong passwords or passphrases are much more difficult to guess, making them a more effective means of securing the system.

Another best practice is to ensure that the wireless network is secured with a strong encryption algorithm, such as WPA2 (Wi-Fi Protected Access II). WPA2 provides a much stronger level of encryption than its predecessor, WEP (Wired Equivalent Privacy), which is vulnerable to several well-known attacks. It is also important to ensure that mobile devices are kept up-to-date with the latest security patches and updates. Mobile devices are frequently targeted by attackers, and vulnerabilities can be exploited to gain access to the device or to intercept data transmitted by the device.

Finally, it is important to be aware of the risks associated with public Wi-Fi networks. Public Wi-Fi networks are often unsecured or poorly secured, making them vulnerable to attacks. It is recommended to avoid using public Wi-Fi networks for sensitive tasks, such as online banking or accessing confidential information[3], [4].In order to give mobile users of wireless phones and other wireless terminals, such as pagers and personal digital assistants (PDAs), access to telephony, the Wireless Application Protocol (WAP) was created by the WAP Forumand information services, such as the Web and the Internet. All wireless network technologies are compatible with WAP (e.g., GSM, CDMA, and TDMA). As far as feasible, WAP is built on current Internet standards including IP, XML, HTML, and HTTP. Facilities for security are also included. Version 2.0 of the WAP standard is the most recent version as of this writing.

The severe restrictions of the devices and the networks that link them have a major impact on how mobile phones and terminals are used for data services. The gadgets' CPUs, memory, and battery life are constrained. The screens are modest, and the user interface is equally restricted. Compared to conventional connections, wireless networks exhibit uncertain availability and stability, significant latency, and comparatively limited capacity. All of these features also differ significantly from one network to the next and from one terminal device to another. Finally, compared to other information system users, mobile and wireless users have different needs and expectations. Mobile terminals, for instance, must be much simpler to use than workstations and personal computers. WAP is made to handle these difficulties. The Wireless Markup Language, which adheres to XML, and a programming model based on the WWW Programming Model are both included in the WAP specification.

The services offered by its processor(s) offload the constrained functionality of portable, mobile, wireless terminals. For instance, the gateway offers DNS services, converts data between the WWW stack (HTTP and TCP/IP) and the WAP protocol stack, compresses data from the Web to reduce wireless communication, and reverse-encodes the compressed data back into standard Web communication conventions. The gateway additionally stores frequently requested data in its cache.

The essential elements of the WAP environment are shown in Figure 6.12. A mobile user can browse Web content on a standard Web server using WAP. The web server offers content in the form of HTML-coded pages that are sent over the HTTP/TCP/IP web protocol stack. The HTML content must pass through an HTML filter, which may be housed in a separate physical module or collocated with the WAP proxy. The filter converts the HTML information into WML information. The WML is delivered to the proxy using HTTP/TCP/IP if the filter and the proxy are separate.

The proxy uses the WAP protocol stack to deliver the binary WML—a more condensed version of WML—to the mobile user over a wireless network. If the Web server has the ability to generate WML content directly, the WML is sent over HTTP/TCP/IP to the proxy, which converts it to binary WML before sending it over WAP protocols to the mobile node.

The WAP architecture is made to deal with the two main drawbacks of wireless Web access: the mobile node's limitations (small screen size, limited input capability), and the wireless digital networks' slow data rates. The small hand-held mobile nodes still only have a few input and display options, despite the advent of 3G wireless networks that offer broadband data rates.

Portable Document Format

WML was created to describe the format and content for presenting data on devices with constrained user input, bandwidth, and screen size. It is made to function with telephone keypads, styluses, and other common mobile, wireless input devices. WML enables scaling of displays for use on larger screens found on smartphones as well as two-line screens found in some small devices. For an ordinary PC, a Web browser provides content in the form of Web pages coded with the Hypertext Markup Language (HTML) (HTML). To translate an HTMLcoded Web page into WML with content and format suitable for wireless devices, much

of the information, especially graphics and animation, must be stripped away. WML presents mainly text-based information that attempts to capture the essence of the Web page and that is organized for easy access for users of mobile devices.

Important features of WML include:

D. Text and image support: Formatting and layout commands are provided for text and limited image capability.

E. Deck/card organizational metaphor: WML documents are subdivided into small, well-defined units of user interaction called cards. Users navigate by moving back and forth between cards. A card specifies one or more units of interaction (a menu, a screen of text, or a text-entry field) (a menu, a screen of text, or a text-entry field). A WML deck is similar to an HTML page in that it is identified by a Web address (URL) and is the unit of content transmission[5], [6].

WIRELESS NETWORK SECURITY

Support for navigation among cards and decks: WML includes provisions for event handling, which is used for navigation or executing scripts. In an HTML-based Web browser, a user navigates by clicking on links. At a WML-capable mobile device, a user interacts with cards, moving forward and back through the deck.

WAP Architecture

The WAP architecture document, illustrates the overall stack architecture implemented in a WAP client. In essence, this is a five-layer model. Each \slayer provides a set of functions and/or services to other services and applications through a set of well-defined interfaces. Each of the layers of the architecture is accessible by the layers above, as well as by other services and applications. Many of the services in the stack may be provided by more than one protocol. For example, either HTTP or WSP may provide the Hypermedia Transfer service.

Common to all five layers are sets of services that are accessible by multiple \slayers. These common services fall into two categories: security services and service discovery.

SECURITY SERVICES

The WAP specification includes mechanisms to provide confidentiality, integrity, authentication, and nonrepudiation. The security services include the following.

A. Cryptographic libraries: This application framework level library provides services for signing of data for integrity and non-repudiation purposes.

B. Authentication: WAP provides various mechanisms for client and server authentication. At the Session Services layer, HTTP Client Authentication (RFC2617) may be used to authenticate clients to proxies and application servers. At the Transport Services layer, WTLS and TLS handshakes may be used to authenticate clients and servers.

C. Identity: The WAP Identity Module (WIM) provides the functions that store and process information needed for user identification and authentication.

D. PKI: The set of security services that enable the use and management of public-key cryptography and certificates.
E. Secure transport: The Transport Services layer protocols are defined for secure transport over datagrams and connections. WTLS is defined for secure transport over datagrams and TLS is defined for secure transport over connections (i.e., TCP) (i.e., TCP).
F. Secure bearer: Some bearer networks provide bearer-level security. For example, IP networks (especially in the context of IPv6) provide bearer-level security with IPsec.

SERVICE DISCOVERY

There is a collection of service discovery services that enable the WAP client and the Web server to determine capabilities and services. Examples of service discovery services include the following.

1. **EFI:** The External Functionality Interface (EFI) allows applications to discover what external functions/services are available on the device.
2. **Provisioning:** This service allows a device to be provisioned with the parameters necessary to access network services.
3. **Navigation discovery:** This service allows a device to discover new network services (e.g., secure pull proxies) during the course of navigation such as when downloading resources from a hypermedia server. The WAP TransportLevel End-to-End Security specification, defines one navigation discovery protocol.
4. Service lookup: This service provides for the discovery of a service's parameters through a directory lookup by name. One example of this is the Domain Name System (DNS) (DNS).

Wireless Application Environment

The WAE specifies an application framework for wireless devices such as mobile telephones, pagers, and PDAs. In essence, the WAE consists of tools and formats that are intended to ease the task of developing applications and devices supported by WAP. The major elements of the WAE model

A. WAE user agents: Software that executes in the user's wireless device and that provides specific functionality (e.g., display content) to the end user.
B. Wireless telephony applications (WTA): A collection of telephony-specific extensions for call and feature control mechanisms that provide authors advanced mobile network services. Using WTA, applications developers can use the microbrowser to originate telephone calls and to respond to events from the telephone network.
C. Standard content encoding: Defined to allow a WAE user agent (e.g., a browser) to conveniently navigate Web content. On the server side are content generators. These are applications (or services) on origin servers (e.g., CGI scripts) that produce standard content formats in response to

requests from user agents in the mobile terminal. WAE does not specify any standard content generators but expects that there will be a variety available running \son typical HTTP origin servers commonly used in WWW today.

D. Push: A service to receive push transmissions from the server, i.e., transmissions that are not in response to a Web client request but are sent on the initiative of the server. This service is supported by the Push-OTA (Push over the Air) session service.
E. Multimedia messaging: Provides for the transfer and processing of multimedia messages, such as e-mail and instant messages, to WAP devices.

WAP Protocol Architecture

The WAP architecture a collection of services at each level and provides interface specifications at the boundary between each pair of layers. Because several of the services in the WAP stack can be provided using different protocols based on the circumstances, there are more than one possible stack configurations. A common protocol stack configuration in which a WAP client device connects to a Web server via a WAP gateway. This configuration is common with devices that implement version 1 of the WAP specification but is also used in version 2 devices (WAP2) if the bearer network does not support TCP/IP. In the remainder of this subsection, we provide an overview of the WAP protocols, with the exception of WTLS, which is treated [7], [8].

WIRELESS SESSION PROTOCOL

WSP provides applications with an interface for two session services. The connection-oriented session service operates above WTP, and the connectionless session service operates above the unreliable transport protocol WDP. In essence,WSP is based on HTTP with some additions and modifications to optimize \sits use over wireless channels. The principal limitations addressed are low data rate \sand susceptibility to loss of connection due to poor coverage or cell overloading.

WSP is a transaction-oriented protocol based on the concept of a request and reply. Each WSP protocol data unit (PDU) consists of a body, which may contain WML, WM Script, or images; and a header, which contains information about the data in the body and about the transaction. WSP also defines a server push operation, in which the server sends unrequested content to a client device. This may be used for broadcast messages or for services, such as news headlines or stock quotes that may be tailored to each client device.

WIRELESS TRANSACTION PROTOCOL

WTP manages transactions by conveying requests and responses between a user agent (such as a WAP browser) and anapplication server for such activities as browsing and e-commerce transactions.

WTP provides a reliable transport service but dispenses with much of the overhead \sof TCP, resulting in a lightweight protocol that is suitable for implementation in \s“thin” clients (e.g.,

mobile nodes) and suitable for use over low-bandwidth wireless \slinks. WTP includes the following features.

Three classes of transaction service.

1. Optional user-to-user reliability: WTP user triggers the confirmation of each received message.
2. Optional out-of-band data on acknowledgments.
3. PDU concatenation and delayed acknowledgment to reduce the number of messages sent.

Asynchronous transactions:

WTP is transaction oriented rather than connection oriented. With WTP, there is \no explicit connection setup or teardown but rather a reliable connectionless service. WTP provides three transaction classes that may be invoked by WSP or another higher layer protocol:

Class 0: Unreliable invoke message with no result message

Class 1: Reliable invoke message with no result message

Class 2: Unreliable invoke message with one reliable result message

Class 0 provides an unreliable datagram service, which can be used for an unreliable push operation. Data from a WTP user are encapsulated by WTP (the initiator, or client) in an invoke PDU and transmitted to the target WTP (the responder, or server) with no acknowledgment. The responder WTP delivers the data to the target WTP user.

WIRELESS NETWORK SECURITY

Class 1 provides a reliable datagram service, which can be used for a reliable push operation. Data from an initiator are encapsulated in an invoke PDU and transmitted to the responder. The responder delivers the data to the target WTP user and acknowledges receipt of the data by sending back an ACK PDU to the WTP entity on the initiator side, which confirms the transaction to the source WTP user. The responder WTP maintains state information for some time after the ACK has been sent to handle possible retransmission of the ACK if it gets lost and/or the initiator retransmits the invoke PDU.

Class 2 provides a request/response transaction service and supports the execution of multiple transactions during one WSP session. Data from an initiator are encapsulated in an invoke PDU and transmitted to the responder, which delivers the data to the target WTP user. The target WTP user prepares response data, which are handed down to the local WTP entity. The responder WTP entity sends these data back in a result PDU. If there is a delay in generating the response data beyond a timer threshold, the responder may send an ACK PDU before sending the result PDU. This prevents the initiator from unnecessarily retransmitting the invoke message.

WIRELESS DATAGRAM PROTOCOL WDP is used to adapt a higher-layer WAP protocol to the communication mechanism called the bearer used between the mobile node and the WAP

gateway. Adaptation may include partitioning data into segments of appropriate size for the bearer and interfacing with the bearer network.

WDP hides details of the various bearer networks from the other layers of WAP. In some instances, WAP is implemented on top of IP.

WIRELESS TRANSPORT LAYER SECURITY

WTLS provides security services between the mobile device (client) and the WAP gateway. WTLS is based on the industry-standard Transport Layer Security (TLS) \sProtocol,3 which is a refinement of the Secure Sockets Layer (SSL) protocol. TLS is the standard security protocol used between Web browsers and Web servers. WTLS \sis more efficient that TLS, requiring fewer message exchanges. To provide end-to-end security, WTLS is used between the client and the gateway, and TLS is used between the gateway and the target server (Figure 6.14). (Figure 6.14). WAP systems translate between WTLS and TLS within the WAP gateway. Thus, the gateway is a point of vulnerability and must be given a high level of security from external attacks.

WTLS provides the following features.

Data integrity: Uses message authentication to ensure that data sent between \sthe client and the gateway are not modified.

Privacy: Uses encryption to ensure that the data cannot be read by a third party.

Authentication: Uses digital certificates to authenticate the two parties.

Denial-of-service protection: Detects and rejects messages that are replayed \sor not successfully verified.

WIRELESS TRANSPORT LAYER SECURITY

Two important WTLS concepts are the secure session and the secure connection, \swhich are defined in the specification as:

Secure connection: A connection is a transport (in the OSI layering model definition) that provides a suitable type of service. For SSL, such connections are peer-to-peer relationships. The connections are transient. Every connection is associated with one session.

Secure session: An SSL session is an association between a client and a server.

Sessions are created by the Handshake Protocol. Sessions define a set of cryptographic security parameters, which can be shared among multiple connections. Sessions are used to avoid the expensive negotiation of new security parameters for each connection.

Between any pair of parties applications such as HTTP on client and server, there may be multiple secure connections. In theory, there may also be multiple simultaneous sessions between parties, but this feature is not used in practice. There are a number of states associated with each session. Once a session is established, there is a current operating state for both read

and write (i.e., receive \sand send). In addition, during the Handshake Protocol, pending read and write \states are created. Upon successful conclusion of the Handshake Protocol, the \spending states become the current states.

A session state is defined by the following parameters:

A. Session identifier: An arbitrary byte sequence chosen by the server to identify \san active or resumable session state.
B. Protocol version: WTLS protocol version number.
C. Peer certificate: Certificate of the peer. This element of the state may be null.
D. Compression method: The algorithm used to compress data prior to encryption.
E. Cipher spec: Specifies the bulk data encryption algorithm (such as null, RC5, DES, etc.) and a hash algorithm (such as MD5 or SHA-1) used for MAC calculation. It also defines cryptographic attributes such as the hash size.
F. Master secret: A 20-byte secret shared between the client and server.
G. Sequence number: Which sequence numbering scheme (off, implicit, or explicit) is used in this secure connection.
H. Key refresh: Defines how often some connection state values (encryption key, MAC secret, and IV) calculations are performed.
I. Is resumable: A flag indicating whether the session can be used to initiate new connections.

The connection state is the operating environment of the record protocol. It includes all parameters that are needed for the cryptographic operations (encryption/decryption and MAC calculation/verification).

WTLS Protocol Architecture

WTLS is not a single protocol but rather two layers of protocols. The WTLS Record Protocol provides basic security services to various higher-layer protocols. In particular, the Hypertext Transfer Protocol (HTTP), which provides the transfer service for Web client/server interaction, can operate on \stop of WTLS. Three higher-layer protocols are defined as part of WTLS: the Handshake Protocol, The Change Cipher Spec Protocol, and the Alert Protocol. These WTLS-specific protocols are used in the management of WTLS exchanges \sand are examined subsequently in this section[9], [10].

WTLS RECORD PROTOCOL

The WTLS Record Protocol takes user data from the next higher layer (WTP, WTLS Handshake Protocol, WTLS Alert Protocol, and WTLS Change Cipher Spec Protocol) and encapsulates these data in a PDU.

- Step 1. The payload is compressed using a lossless compression algorithm.
- Step 2. A message authentication code (MAC) is computed over the compressed data, using HMAC. One of several hash algorithms can be used with HMAC, including MD-5 and SHA-1. The length of the hash code is 0, 5, or 10 bytes. The MAC is added after the compressed data.

- Step 3. The compressed message plus the MAC code are encrypted using a symmetric encryption algorithm. The allowable encryption algorithms are DES, triple DES, RC5, and IDEA.
- Step 4. The Record Protocol prepends a header to the encrypted payload.

The Record Protocol header consists of the following fields.

1. Record type (8 bits): Consisting of the subfields:

- Record length field indicator (1 bit): Indicates whether a record length field \sis present.
- Sequence number field indicator (1 bit): Indicates whether a sequence number field is present.
- Cipher spec indicator (1 bit): If this bit is zero, it indicates that no compression, MAC protection, or encryption is used.
- Content type (4 bits): The higher-layer protocol above the WTLS Record Protocol.

2. Sequence number (16 bits): A sequence number associated with this record.

This provides reliability over an unreliable transport service. Record length (16 bits):The length in bytes of the plaintext data (or compressed data if compression is used).

CHANGE CIPHER SPEC PROTOCOL

Associated with the current transaction is a cipher spec, which specifies the encryption algorithm, the hash algorithm used as User Data Compresspart of HMAC, and cryptographic attributes, such as MAC code size. There are two states associated with each session. Once a session is established, there is a current operating state for both read and write (i.e., receive and send) (i.e., receive and send). In addition, during the Handshake Protocol, pending read and write states are created.

The Change Cipher Spec Protocol is one of the three WTLS-specific protocols that use the WTLS Record Protocol, and it is the simplest. This protocol consists of a single message, which consists of a single byte with the value 1. The sole purpose of this message is to cause the pending state to be copied into the current state, which updates the cipher suite to be used on this connection. Thus, when the Change Cipher Spec message arrives, the sender of the message sets the current write state to the pending state and the receiver sets the current read state to the pending state.ALERT PROTOCOL The Alert Protocol is used to convey WTLS-related alerts to the peer entity. As with other applications that use WTLS, alert messages are compressed and encrypted, as specified by the current state.

r = reserved

C = cipher spec indicator

S = sequence number field indicator

L = record length field indicator

MAC = message authentication code

Content type

Sequence number

Record length rCSL Plaintext \s(optionally compressed)

MAC (0, 16, or 20 bytes) (0, 16, or 20 bytes)

Encrypted Scope of MAC

Each message in this protocol consists of 2 bytes. The first byte takes the value warning (1), critical (2), or fatal (3) to convey the severity of the message. The second byte contains a code that indicates the specific alert. If the level is fatal, WTLS immediately terminates the connection. Other connections on the same session may continue, but no new connections on this session may be established. A critical alert message results in termination of the current secure connection. Other connections using the secure session may continue and the secure identifier may also be used for establishing new secure connections.The connection is closed using the alert messages. Either party may initiate the exchange of the closing messages. If a closing message is received, then any data after this message is ignored. It is also required that the notified party verifies termination of the session by responding to the closing message. Error handling in the WTLS is based on the alert messages. When an error is detected, the detecting party sends an alert message containing the occurred error. Further procedures depend on the level of the error that occurred.

Examples of fatal alerts:

A. **Session close notify:** notifies the recipient that the sender will \snot send any more messages using this connection state or the secure session.
B. **Unexpected message:** An inappropriate message was received.
C. **Bad record mac:** An incorrect MAC was received.
D. **Decompression failure:** The decompression function received improper input (e.g., unable to decompress or decompress to greater than maximum allowable length).
E. **Handshake failure:** Sender was unable to negotiate an acceptable set of security parameters given the options available.
F. **Illegal parameter:** A field in a handshake message was out of range or inconsistent with other fields.

Examples of nonfatal alerts:

A. **Connection close notify:** Notifies the recipient that the sender will \snot send any more messages using this connection state.
B. **Bad certificate:** A received certificate was corrupt (e.g., contained a signature that did not verify) (e.g., contained a signature that did not verify).
C. **Unsupported certificate:** The type of the received certificate is not supported.
D. **Certificate revoked:** A certificate has been revoked by its signer.
E. **Certificate expired:** A certificate has expired.

F. **Certificate unknown:** Some other unspecified issue arose in processing the certificate, rendering it unacceptable.

HANDSHAKE PROTOCOL

The most complex part of WTLS is the Handshake Protocol. This protocol allows the server and client to authenticate each other and to negotiate an encryption and MAC algorithms and cryptographic keys.

REFERENCES:

[1] I. Unwala, Z. Taqvi, and J. Lu, "Thread: An IoT protocol," in *IEEE Green Technologies Conference*, 2018. doi: 10.1109/GreenTech.2018.00037.

[2] J. Cai *et al.*, "A Handshake Protocol with Unbalanced Cost for Wireless Updating," *IEEE Access*, 2018, doi: 10.1109/ACCESS.2018.2820086.

[3] Y. Yilmaz, S. R. Gunn, and B. Halak, "Lightweight PUF-based authentication protocol for IoT devices," in *2018 IEEE 3rd International Verification and Security Workshop, IVSW 2018*, 2018. doi: 10.1109/IVSW.2018.8494884.

[4] O. Y. Erlikaya and G. Dalkiltc, "Authentication and Authorization Mechanism on Message Queue Telemetry Transport Protocol," in *UBMK 2018 - 3rd International Conference on Computer Science and Engineering*, 2018. doi: 10.1109/UBMK.2018.8566599.

[5] A. Tsetse, E. Bonniord, P. Appiah-Kubi, and S. Tweneboah-Kodua, "Performance Study of the Impact of Security on 802.11ac Networks," in *Advances in Intelligent Systems and Computing*, 2018. doi: 10.1007/978-3-319-77028-4_3.

[6] S. Sundar and S. Sumathy, "Security stipulations on iot networks," in *Lecture Notes on Data Engineering and Communications Technologies*, 2018. doi: 10.1007/978-3-319-70688-7_12.

[7] M. M. Islam, S. Paul, and M. M. Haque, "Reducing network overhead of IoTDTLS protocol employing ChaCha20 and Poly1305," in *20th International Conference of Computer and Information Technology, ICCIT 2017*, 2018. doi: 10.1109/ICCITECHN.2017.8281857.

[8] H. Ahmadi, K. Katzis, and M. Z. Shakir, "A novel airborne self-organising architecture for 5G+ networks," in *IEEE Vehicular Technology Conference*, 2018. doi: 10.1109/VTCFall.2017.8288095.

[9] M. Radha and M. Nagabhushana Rao, "Detection and prevention of internal external and chain of black hole attack using Iecbhdp methodology," *J. Adv. Res. Dyn. Control Syst.*, 2018.

[10] A. H. Gerez, K. Kamaraj, R. Nofal, Y. Liu, and B. Dezfouli, "Energy and Processing Demand Analysis of TLS Protocol in Internet of Things Applications," in *IEEE Workshop on Signal Processing Systems, SiPS: Design and Implementation*, 2018. doi: 10.1109/SiPS.2018.8598334.

CHAPTER 18

ELECTRONIC MAIL SECURITY

Mr. Aishwary Awasthi, Research Scholar
Department of Mechanical Engineering, Sanskriti University, Mathura, Uttar Pradesh, India
Email Id- aishwary@sanskriti.edu.in

Electronic mail (email) is one of the most popular means of communication in the modern world. While email has many benefits, it also poses significant security risks. Email security is essential to protect sensitive and confidential information from cyber attackers who can intercept, modify or steal messages. In this article, we will discuss email security and the measures you can take to keep your emails secure.

Threats to Email Security

There are many types of email security threats. The most common are:

1. **Phishing:** This is a social engineering attack where an attacker poses as a legitimate person or organization to trick the recipient into clicking on a malicious link or opening a malicious attachment.
2. **Malware:** Malware is a type of malicious software that can infect your computer through an email attachment or link. Once the malware is on your computer, it can steal data, delete files, or even take control of your system.
3. **Email Spoofing:** Email spoofing is a technique used by attackers to make the email appear to come from a legitimate sender. The attacker uses a fake email address, which makes it difficult for the recipient to determine whether the email is legitimate or not.
4. **Man-in-the-middle attack:** A man-in-the-middle (MITM) attack is where an attacker intercepts the communication between the sender and the recipient. The attacker can then modify the email or steal information from it.

Email Security Measures

There are several email security measures you can take to protect yourself from the above-mentioned threats. These measures include:

1. **Use strong passwords:** Use strong, unique passwords for your email account. Avoid using easy-to-guess passwords such as birth dates, pet names, or common words. Instead, use a combination of upper and lowercase letters, numbers, and special characters.
2. **Enable two-factor authentication:** Two-factor authentication (2FA) adds an extra layer of security to your email account. With 2FA, you will need to enter a verification code in addition to your password to log in to your email account.

3. **Use encryption:** Encryption is a process that converts data into an unreadable format that can only be decoded with a key. Encryption can be used to protect email messages, attachments, and even the entire email account.
4. **Beware of phishing emails:** Be cautious of emails that request sensitive information or contain suspicious links or attachments. Always verify the sender's email address and never click on links or download attachments from unknown sources.
5. **Install anti-malware software:** Install anti-malware software on your computer to protect against malware that can infect your computer through email attachments or links.
6. **Enable spam filters:** Enable spam filters to automatically filter out unwanted emails that may contain spam, phishing emails, or other types of malware.
7. **Use a secure email service:** Choose a secure email service that uses encryption to protect your messages and attachments. Examples of secure email services include ProtonMail, Tutanota, and Hushmail[1], [2].

Email security is essential to protect sensitive and confidential information from cyber attackers. There are several email security measures you can take to keep your email secure, including using strong passwords, enabling two-factor authentication, using encryption, being cautious of phishing emails, installing anti-malware software, enabling spam filters, and using a secure email service. By implementing these measures, you can reduce the risk of your email being hacked and protect your privacy and data. In addition to the email security measures outlined in the previous section, there are several other important factors to consider when it comes to email security.

1. **Keep Your Email Software Up to Date**

One of the most effective ways to improve email security is to keep your email software up to date. This includes both the email client on your computer or mobile device and the server software used by your email provider.

Software updates often include security patches that fix known vulnerabilities and address newly discovered security threats. By keeping your email software up to date, you can reduce the risk of your email being hacked or compromised.

2. **Limit Your Use of Public Wi-Fi**

Public Wi-Fi networks can be convenient when you need to check your email on the go, but they are often unsecured and vulnerable to hacking. If you must use public Wi-Fi to access your email, use a virtual private network (VPN) to encrypt your traffic and protect your privacy.

3. **Avoid Sending Sensitive Information via Email**

Email is not a secure method of transmitting sensitive information such as credit card numbers, social security numbers, or passwords. If you must send sensitive information via email, use encryption to protect the message and its contents. You can use a secure email service, such as

ProtonMail, that offers end-to-end encryption or use a file encryption program to encrypt the information before sending it.

4. **Be Careful with Email Attachments**

Email attachments can be a major source of security threats, as they can contain viruses, malware, or other types of malicious software. Be careful when opening email attachments, especially from unknown sources. If you receive an attachment that you were not expecting or that seems suspicious, do not open it. Instead, contact the sender and confirm that they intended to send the attachment.

5. **Use Disposable Email Addresses**

Disposable email addresses are temporary email addresses that you can use for a specific purpose, such as signing up for a newsletter or creating an account on a website. These email addresses can help protect your privacy and reduce the risk of spam and other unwanted email. Services such as Mailinator or 10 Minute Mail provide disposable email addresses that can be used for this purpose.

6. **Back Up Your Emails**

Regularly backing up your emails can help protect your data and ensure that you can recover your messages in case of a security breach or system failure. You can back up your email messages to an external hard drive, a cloud storage service, or another secure location.

7. **Train Your Employees on Email Security Best Practices**

If you run a business, it is important to train your employees on email security best practices. This includes educating them on the risks of phishing, malware, and other email-based security threats, as well as providing guidance on how to use email safely and securely.

Training should cover best practices for creating and managing passwords, how to recognize and avoid phishing scams, how to handle email attachments, and other key email security topics. By educating your employees on email security best practices, you can help protect your business from security breaches and other types of cyber attacks.

Email security is an important consideration for anyone who uses email to communicate or exchange sensitive information. By taking steps to protect your email, you can reduce the risk of your data being compromised or stolen. Some of the most important email security measures include using strong passwords, enabling two-factor authentication, using encryption, being cautious of phishing emails, installing anti-malware software, and using a secure email service.

It is also important to keep your email software up to date, avoid sending sensitive information via email, be careful with email attachments, use disposable email addresses, back up your emails, and train your employees on email security best practices. By following these email security best practices, you can help protect your data and maintain the privacy of your communications.

Email is the most widely utilized network-based application in almost all dispersed systems. Users anticipate being able to send emails to those who are connected to the Internet either directly or indirectly and do so independent of the communications software or host operating system. There is a rising need for authentication and secrecy services as e-mail use skyrockets. In terms of widely used techniques, Pretty Good Privacy (PGP) and S/MIME stand out as the two most popular systems. In this chapter, both are investigated. A discussion of DomainKeys Identified Mail brings the chapter to a conclusion[3], [4].

EXCELLENT PRIVACY

PGP is an astonishing occurrence. PGP, which is mostly the result of the work of one individual, Phil Zimmermann, offers an authentication and secrecy service that can be used to applications for electronic communication and file storage. Basically, Zimmermann has accomplished the following:

A. Selected the top cryptographic building blocks currently on the market.
B. Added these algorithms to a general-purpose program that relies on a manageable number of simple instructions and is independent of the operating system and CPU.
C. Distributed the package's documentation and source code without charge over the Internet, message boards, and paid networks like AOL (America On Line).
D. Signed a contract with a business (Viacrypt, now Network Associates) to provide a fully functional, reasonably priced commercial version of PGP.

PGP has rapidly expanded and is currently extensively used. There are several explanations for this expansion.

1. It runs on many different systems, including Windows, UNIX, Macintosh, and many more, and is freely accessible everywhere. Additionally, those who want a product with vendor support will find satisfaction with the commercial version.
2. It is based on algorithms that are very secure and have withstood intense public scrutiny. The package specifically contains CAST-128, IDEA, and 3DES for symmetric encryption, RSA, DSS, and Diffie-Hellman for public-key encryption, and SHA-1 for hash coding.
3. It has a broad variety of applications, from people who want to securely interact with others across the globe through the Internet and other networks to companies that want to choose and enforce a standardized system for encrypting data and communications.
4. No governmental or standards body established it or has any authority over it. This makes PGP appealing to people with a natural mistrust of "the establishment".
5. PGP is now on a road to become an Internet standard (RFC 3156; MIME Security with OpenPGP). PGP yet still has an air of an anti-establishment project.

We start by taking a broad view of PGP's functionality. Next, we look at the generation and storage of cryptographic keys. The crucial topic of public-key management is then covered [5], [6].

Notation

A few words are new, although the majority of the notation used in this chapter has been used previously. Perhaps it would be preferable to list them first. The symbols listed below are employed. The symmetric encryption method's session key

A. Using user A's private key in a public-key encryption technique
B. Using user A's public key in a public-key encryption technique
C. Public-key cryptography
D. Decryption using public keys
E. Heterogeneous encryption

The PGP documentation often refers to a key that is associated with a public key in a public-key encryption scheme as a secret key. This method runs the danger of being confused with a secret key used for symmetric encryption, as was already noted. Therefore, we refer to private keys instead.

Description of the operation

Four services make up PGP's real functioning, as opposed to key management: email compatibility, confidentiality, and authentication. We look at each of them individually.

AUTHENTICATION

This is the digital signature system. The order is as stated below.

1. The communication is made by the sender.
2. A 160-bit hash code of the message is produced using SHA-1.
3. The result of the RSA encryption, which uses the sender's private key, is prepended to the message.
4. To decode and get the hash code, the receiver employs RSA along with the sender's public key.
5. The recipient creates a fresh hash code for the message and contrasts it with the original, before decrypting it. The message is recognized as legitimate if the two match.

The use of SHA-1 and RSA together creates a strong digital signature system. The receiver is certain that only the owner of the matching private key can create the signature because to RSA's strength. The receiver is certain that no one else could create a new message that matches the hash code and, therefore, the signature of the original message due to the strength of SHA-1.

Instead, DSS/SHA-1 may be used to create signatures.

Despite the fact that signatures are often contained in the message or file they sign, this is not always the case: It is possible to use detached signatures. The message that a detachable

signature signs may be saved and transferred independently. This is helpful in a variety of situations. It may be desirable for a user to keep a separate signature record of each communication transmitted or received. An executable program's detached signature may identify a future viral infection. Finally, when more than one participant is required to sign a document, such as a legal contract, detached signatures may be employed. Because each person's signature is distinct from the others, it only applies to the paper. If not, signatures would need to be nested, with the next signer having to sign both the document and the first signature before continuing.

CONFIDENTIALITY PGP also offers secrecy, which is achieved by encrypting communications that are sent over the Internet or that are saved locally as files.

The symmetric encryption method CAST-128 is applicable in both scenarios.

Alternatives include IDEA and 3DES.

It utilizes the 64-bit cipher feedback (CFB) mode.

As usual, the issue of key distribution has to be addressed. Each symmetric key in PGP is only ever used once. That is, for every communication, a fresh key is created as a random 128-bit integer. Consequently, although though this is referred to as a session key in the documentation, it is really a one-time key. The session key is attached to the message and sent with it since it can only be used once. The recipient's public key is used to encrypt the key and secure it.

A. The sender creates a message and a random 128-bit integer that will only be used for this message as a session key.
B. Using the session key and CAST-128 (or IDEA, or 3DES), the message is encrypted.
C. The message is prepended with the session key encrypted using RSA using the recipient's public key.
D. To decode and get the session key, the receiver employs RSA along with its private key.
E. The message is decrypted using the session key.

PGP offers the Diffie-Hellman encryption method as an alternative to RSA for key encryption. DiffieHellman is a key exchange method, as was stated. In actuality, PGP makes use of ElGamal, a Diffie-Hellman variation that does provide encryption and decryption. There may be several observations. First, instead of only utilizing or ElGamal to encrypt the message directly, the combination of symmetric and public-key encryption is utilized to shorten the encryption time: Compared to RSA or ElGamal, CAST-128 and the other symmetric algorithms are much quicker. Second, since only the receiver is able to retrieve the session key associated with the message, using the public-key technique answers the issue of how to distribute session keys. Note that because we are not starting an ongoing session, we do not need the kind of session-key exchange mechanism covered[7], [8].

Instead, every communication is a distinct, once-only event with its own key. Furthermore, using handshaking to ensure that both sides have the same session key is impractical given the store-

and-forward nature of electronic mail. Last but not least, the use of one-time symmetric keys reinforces the already effective symmetric encryption strategy. Each key only encrypts a little portion of plaintext, and the keys have no connection to one another. As a result, the system as a whole is secure to the degree that the public-key algorithm is safe. To do this, PGP offers the user a selection of key size choices ranging from 768 to 3072 bits (the DSS key for signatures is limited to 1024 bits).

AUTHENTICITY AND CONFIDENTIALITY

Both services may be utilized for the same message. First, a plaintext message's signature is created and prepended to the message. Then, CAST-128 (or IDEA or 3DES) is used to encrypt the plaintext message and signature, and RSA is used to encrypt the session key (or ElGamal). The order described here is superior to the alternative, which involves encrypting the message before producing a signature for it. Generally speaking, it is more practical to store a signature with a plaintext version of a communication.

A third party need not worry about the symmetric key while confirming the signature if the signature is done first for third-party verification reasons. While using both services, the sender signs the message with its own private key first, then encrypts it with a session key, and then encrypts the session key with the recipient's public key.

COMPRESSION PGP

By default compresses the message after signing it but before to encrypting it. This offers the advantage of conserving space for file storage as well as for sending and receiving emails. The positioning of the compression method, shown in Figure 7.1 by the letters Z for compression and Z-1 for decompression, is crucial.

1. For two reasons, the signature is created before compression:

In order to keep just the uncompressed message and the signature for future verification, it is better to sign an uncompressed message. It would be essential to either keep a compressed version of the message for subsequent verification or to recompress the message when verification is necessary if one signed a compressed document. PGP's compression technique poses a challenge, even if one were ready to dynamically construct a recompressed message for verification. The process is not deterministic; different implementations of the algorithm output different compressed forms as a consequence of varying tradeoffs in running performance vs compression ratio. The fact that any version of the algorithm may successfully decompress the output of any other version makes these various compression techniques compatible. After compression, all PGP implementations would be required to use the same version of the hash 228 CHAPTER 7 / ELECTRONIC MAIL SECURITY function and signature.

2. **To increase cryptographic security, message encryption is used after compression.**

Cryptanalysis is more challenging since the compressed message has less redundancy than the original plaintext.The ZIP compression method is used; further information is provided in Appendix G.

COMPATIBILITY WITH E-MAIL

When PGP is utilized, the block that is being transferred is at least partially encrypted. The message digest is encrypted using the sender's private key if just the signature service is employed. When using the secrecy service, the message and any current signatures are encrypted with a one-time symmetric key. Thus, a stream of random 8-bit octets makes up all or a portion of the final block.

However, many electronic mail platforms only allow the usage of ASCII text blocks. PGP offers the service of transforming the unprocessed 8-bit binary stream into a stream of writable ASCII characters in order to get around this constraint. The conversion method utilized for this is radix-64. Four ASCII characters are assigned to each set of three binary octets. A CRC is also added to this format in order to identify transmission issues. For a description, see Appendix 7A.

A message may be expanded by 33% when using radix 64. Fortunately, the plaintext message has been compressed, and the message's session key and signature are both rather small. The compression should really more than make up for the radix-64 growth. For instance, [HELD96] indicates that using ZIP, the average compression ratio is about 2.0. If the signature and other important components are ignored, the average overall result of compression and expansion on a file this size would be. As a result, there is still a roughly one-third compression overall.

The radix-64 method is notable for its blind conversion of the input stream to the radix-64 format regardless of the content, even if the input is ASCII text. The result will be illegible to a casual observer if a message is signed but not encrypted and the conversion is done to the full block, providing a certain measure of anonymity. PGP may be set up to solely convert the signature part of signed plaintext communications to radix-64 format as an option. As a result, the communication may be read by the human receiver without the need for PGP.

The four services that have been addressed so far are related. A signature is created upon transmission (if necessary) using the uncompressed plaintext's hash code. The plaintext is then compressed together with the signature, if any. The block (compressed plaintext or compressed signature + plaintext) is then encrypted and prepended with the public-keyencrypted symmetric encryption key if secrecy is necessary. The incoming block is initially transformed back to binary after being received in radix-64 format. The receiver then finds the session key and decrypts the message if it was encrypted. The block that results is then uncompressed. If the message is signed, the receiver finds the hash code that was sent and compares it to the hash code it computed on its own.

Key chains and keys with encryption

PGP uses four different key types: passphrase-based symmetric keys, public keys, private keys, and one-time session symmetric keys (explained subsequently). With regard to these keys, three different needs may be found.

- It is necessary to have a method for generating random session keys.
- We want to let a user have more than one public-key/private-key combination. One explanation is that the user could sometimes want to switch out their key combination.

Any messages in the pipeline at that point will have been built using an outdated key. Furthermore, until an update reaches them, receivers will only be aware of the previous public key. A user may want to have many key pairs available at once in addition to the requirement to periodically change keys in order to communicate with various correspondent groups or just to increase security by restricting the quantity of data that may be encrypted with a particular key. The end result of all of this is that users and public keys do not always correlate exactly. Therefore, a method for detecting certain keys is required.

Each PGP entity must have a file of both its own public/private key pairs and a file of correspondents' public keys.

We look at each of these demands separately.

GENERATION OF SESSION KEYS

Each session key is connected to a single message and is only used to encrypt and decode that particular communication. Remember that symmetric encryption algorithms are used for both message encryption and decryption.

3DES employs a 168-bit key, while CAST-128 and IDEA use 128-bit keys. The debate that follows will be predicated on CAST-128. CAST-128 is used to create random 128-bit integers. Two 64-bit blocks that are regarded as plaintext before being encrypted and a 128-bit key make up the input to the random number generator. The CAST-128 encrypter generates two blocks of 64-bit cipher text in cipher feedback mode, which are combined to provide the 128-bit session key. The ANSI X12.17 algorithm is the basis for the algorithm that is employed.

Two 64-bit blocks that make up the "plaintext" input to the random number generator are created from a stream of 128-bit randomized values. These figures are based on the user's keyboard input. The randomized stream is generated using both the time of the keystrokes and the actual keys pressed. As a result, if the user presses random keys at their usual rate, a fairly "random" input will be produced. The previous session key output from CAST-128 is used with this random input to create the key input for the generator. The end result is an effectively unpredictable sequence of session keys due to the effective scrambling of CAST-128.

More information on PGP random number generating algorithms is provided in Appendix H.

KEY IDENTIFICATIONS

As we have explained, the session key that was used to encrypt the message is included with every encrypted communication. The recipient's public key is used to encrypt. So, in order to retrieve the session key and subsequently the message, only the receiver will be able to do so. The receiver would instantly know which key to use to decrypt the session key the recipient's

particular private key if each user used a single public/private key combination. However, we have made it clear that it is OK for a single user to own several public/private key combinations.

How can the receiver discover which of its public keys was used to encrypt the session key, then? The public key might be sent together with the message as a straightforward approach. The receiver might then continue after confirming that this is one of its public keys. This plan might work, but it wastes space needlessly. The length of an RSA public key may reach hundreds of decimal digits.

Another option would be to provide each public key a unique identification that is only used by that person. In other words, a key might be uniquely identified by its combination of user ID and key ID. Only the considerably smaller key ID would then need to be sent. However, this method brings up a management and administrative issue: In order for the sender and the receiver to be able to map from the key ID to the public key, key IDs must be issued and preserved. This feels like an unnecessary burden.

The approach used by PGP is to provide each public key a key ID that, quite likely, is distinct inside a user ID. Each public key's least significant 64 bits make up the key ID that is connected with it. In other words, the public key's key ID is (mod). Given this length, there is extremely little chance that two key IDs will exist simultaneously.

Additionally, a key ID is necessary for the PGP digital signature. The receiver must be aware of which public key is meant to be used since the sender may encrypt the message digest using any number of different private keys. As a result, the 64-bit key ID of the necessary public key is included in the message's digital signature component. When a message is received, the receiver first confirms that the key ID belongs to a sender whose public key it is familiar with before checking the signature[9], [10].

The 160-bit SHA-1 digest of the message, encrypted using the sender's private signature key. The signature timestamp is multiplied by the message component's data part to generate the digest. The signature timestamp's inclusion in the digest protects against replay-style attacks. The message component's filename and date are excluded, guaranteeing that detached signatures are identical to attached signatures prefixed to the message. Signatures that are detached are 264 PUa PUa 232.

A. The first two octets of the message digest: By contrasting this plaintext copy with the first two octets of the encrypted digest, the receiver may establish whether the right public key was used to decode the message digest for authentication. These octets also function as the message's 16-bit frame check sequence.
B. Public key of the sender's key ID: reveals the private key that was used to encrypt the message digest as well as the public key that should be used to decode the message digest.
C. The session key may be used to encrypt the message component and optional signature component once they have been compressed using ZIP.

The recipient's public key identification and the session key, which was encrypted by the sender using the recipient's public key, are both included in the session key component. Key chains we have seen that key IDs are essential to PGP's functionality and that each PGP message that offers both secrecy and authentication must include two key IDs. In order for all parties to utilize these keys effectively and efficiently, they must be saved and structured in a methodical manner. In PGP, each node is equipped with two data structures, one for storing the public/private key pairs belonging to that node and the other for storing the public keys of other users who are known to this node. The terms "private-key ring" and "public-key ring" are used to refer to these two types of data structures, respectively.

It makes sense to make the value of the private key as safe as feasible, even while it is intended that the private-key ring be kept only on the workstation of the person who generated and owns the key pairs and that it be available only to that user. In light of this, the private key itself is not kept on the key chain. This key is instead CAST-128 encrypted (or IDEA or 3DES). The steps are as follows:

A. The user decides on a password for private key encryption.
B. The user is prompted for the passphrase when the system produces a new public/private key pair using RSA. The password is converted into a 160-bit hash code using SHA-1, which is then discarded.
C. The system uses CAST-128 to encrypt the private key, utilizing the hash code's 128 bits as the key.

The encrypted private key is then saved in the private-key ring once the hash code has been destroyed.

The user must then enter the password in order to access the private-key ring and get a private key. PGP will extract the encrypted private key, create the passphrase hash code, and then use the hash code and CAST-128 to decode the private key.

This plan is incredibly efficient and condensed. The security of this system rests on the security of the password, just as it does with any system that uses passwords. The user should choose a passphrase that is difficult to figure out yet simple to remember in order to resist the urge to write it down.

A public-key ring's overall structure is also seen in Figure 7.4. The public keys of other users who are familiar to this user are stored in this data structure. Let's overlook several of the fields in the figure for the time being and talk about the following ones.

1. Timestamp: The time and date at which this entry was produced.
2. Key ID: This entry's public key's least significant 64 bits.
3. Public Key: The public key for this entry.
4. User ID: This number identifies the key's owner. A single public key may have several user IDs attached to it.

We shall see why both methods of indexing are necessary later. The public-key ring may be indexed by either the User ID or the Key ID. We can now demonstrate the transmission and

receiving of messages using these key rings. For the sake of simplicity, we exclude compression and radix-64 conversion from the explanation that follows. First, think about message transmission (Figure 7.5) and presumptively sign and encrypt the message. The sender PGP entity completes the subsequent actions.

1. Signing the message:

   a. Using your userid as an index, PGP locates the sender's private key in the private-key ring. The command retrieves the first private key on the ring if your userid was omitted.
   b. PGP asks the user for the password in order to get the private key that is not encrypted.
   c. The message's signature part is put together.

The communication is encrypted using a session key that is generated by PGP. Using her userid as an index, PGP locates the recipient's public key in the public-key ring. The message's session key component is created. The receiving PGP entity completes the subsequent actions.

Decrypting the message: Using the Key ID field in the session key component of the message as an index, PGP obtains the receiver's private key from the private-key ring. PGP asks the user for the password in order to get the private key that is not encrypted. PGP decrypts the message after recovering the session key.

PRIVACY IS PRETTY GOOD

PGP offers a framework for resolving this issue with a number of potential solutions. No fixed public-key management strategy is established since PGP is meant to be used in a range of official and informal settings, as we shall see when we explore S/MIME later in this chapter.

PUBLIC-KEY MANAGEMENT APPROACHES

The crux of the issue is as follows: To interact with other PGP users, User A must create a public-key ring containing their public keys. Consider the scenario where A's key ring includes a public key that is shown as belonging to B but is really held by C. This would occur, for instance, if A obtained the key from a bulletin board system (BBS) that had been hacked by C but had been used by B to publish the public key. As a consequence, there are now two dangers. First, C may send messages to A while impersonating B in order to get A to acknowledge the message as originating from B. Second, C can decrypt any communication sent from A to B. There are many methods for reducing the possibility that a user's public-key ring includes fake public keys. Let's say A wants to get B a trustworthy public key. The following are some potential strategies.

A. Take the key directly from B. B may provide A a floppy disk with her public key stored on it. The key may then be loaded onto A's machine via the floppy disk. Although incredibly safe, this system clearly has certain practical drawbacks.
B. Telephone-verify a key. If A can identify B over the phone, A may call B and ask her to speak the key aloud in radix-64 format. The more realistic option would be for B to provide A her key by email. The "fingerprint" of the key is a 160-bit

SHA-1 digest of the key that PGP may produce and show in hexadecimal format. Then, A might call B's number and request that she narrate the fingerprint over the phone. The key is confirmed if the two fingerprints are identical.

C. Obtain B's public key from D, who is a mutually trusted party. The introducer, D, generates a signed certificate for this reason. The certificate contains the public key of B, the date the key was generated, and the key's duration. D creates a SHA-1 digest of this certificate, encrypts it using her personal key, and appends her signature to it. No one else can produce a fake public key and claim that it is signed by D since only D could have produced the signature. The signed certificate might either be placed on a bulletin board or given directly to A by B or D.

D. Get hold of B's public key from a reliable certifying body. Once again, the authority generates and signs a public-key certificate. A might then use the authority, provide a user name, and get a certificate that has been signed.

In situations 3 and 4, A would already need a copy of the introducer's public key and the assurance that it is legitimate. A is ultimately responsible for deciding how much faith to place in each introducer.

USAGE OF TRUST

Despite the fact that PGP does not provide any specifications for creating certifying authorities or for building trust, it does offer a practical method for employing trust, connecting trust to public keys, and making use of trust data.

These are the fundamental building blocks. According to the description in the previous paragraph, every item in the public-key ring is a public key certificate. Each of these entries has a key legitimacy field that shows how much PGP will believe that this is a legitimate public key for this user; the greater the degree of confidence, the more strongly this user ID will be bound to this key. By PGP, this field is calculated. Zero or more signatures that the owner of the key ring has gathered and that sign this certificate are also included with the entry. Each signature, in turn, has a signature trust field attached to it that shows how much the PGP user trusts the signer to certify public keys. The collection of signature trust fields in the entry is where the key legitimacy field is obtained. Each item describes a public key linked with a certain owner, and it also includes a user-assigned owner trust field that indicates how much this public key is trusted to sign additional public-key certificates. The owner trust field from another entry may be seen as a cached copy in the signature trust fields.

REFERENCES:

[1] K. Waltermire and W. Barker, “Domain Name System-Based Electronic Mail Security,” *Natl. Inst. Stand. Technol.*, 2018.

[2] L. Mohan and S. Elayidon, “Secure and Privacy Preserving Mail Servers using Modified Homomorphic Encryption (MHE) Scheme,” *Int. J. Adv. Comput. Sci. Appl.*, 2018, doi: 10.14569/ijacsa.2018.090316.

[3] D. Anggraini and S. Juanita, "Aplikasi E-Arsip Pengamanan Pesan Elektronik Berbasis Web dengan Mengimplementasikan Algoritma Kriptografi RSA dan Elgamal pada Klinik Dr. H. Hartono," *J. TICOM*, 2018.

[4] N. Innab, H. Al-Rashoud, R. Al-Mahawes, and W. Al-Shehri, "Evaluation of the Effective Anti-Phishing Awareness and Training in Governmental and Private Organizations in Riyadh," in *21st Saudi Computer Society National Computer Conference, NCC 2018*, 2018. doi: 10.1109/NCG.2018.8593144.

[5] Y. Chandu, K. S. Rakesh Kumar, N. V. Prabhukhanolkar, A. N. Anish, and S. Rawal, "Design and implementation of hybrid encryption for security of IOT data," in *Proceedings of the 2017 International Conference On Smart Technology for Smart Nation, SmartTechCon 2017*, 2018. doi: 10.1109/SmartTechCon.2017.8358562.

[6] D. Maldonado-Ruiz, E. Loza-Aguirre, and J. Torres, "A proposal for an improved distributed architecture for open pgp's web of trust," in *Proceedings - 2018 International Conference on Computational Science and Computational Intelligence, CSCI 2018*, 2018. doi: 10.1109/CSCI46756.2018.00022.

[7] S. Soni, D. Mehta, N. K. Mehta, and R. K. Mehta, "CONSUMERS' ATTITUDES TOWARDS ONLINE SHOPPING IN NOIDA CITY," *Prestig. Int. J. Manag. Res.*, 2018.

[8] A. Bamrara, "Identifying and analyzing the latent cyber threats in developing economies," in *Cyber Security and Threats: Concepts, Methodologies, Tools, and Applications*, 2018. doi: 10.4018/978-1-5225-5634-3.ch051.

[9] W. Stallings, "E-mail Security Using Pretty Good Privacy," in *Handbook of Heterogeneous Networking*, 2018. doi: 10.1201/9781351072625-72.

[10] K. Dhevan and M. Vidya, "Digital Buyer Behaviour in B2C Model - An Empirical Research Study in Trichirappalli District of Tamil Nadu," *Sumedha J. Manag.*, 2018.

CHAPTER 19

IP SECURITY

Dr. Narendra Kumar Sharma, Assistant Professor
Department of Computer Science and Engineering, Sanskriti University, Mathura, Uttar Pradesh, India
Email Id- narendra@sanskriti.edu.in

IP security, also known as Internet Protocol security or IPsec, is a protocol suite designed to provide secure communication over IP networks. IPsec provides confidentiality, integrity, and authentication for data transmitted over IP networks, including the internet. IPsec can be used to secure communication between two hosts, between a host and a network, or between two networks.

The need for IPsec arises from the fact that IP networks are inherently insecure, with no built-in security features. IP packets can be intercepted, modified, or replayed, making it easy for an attacker to eavesdrop on network traffic or steal sensitive data. IPsec addresses these security concerns by providing encryption and authentication of IP packets, ensuring that they cannot be read or modified by unauthorized parties.

IPsec provides several security services, including:

1. Confidentiality: IPsec can encrypt IP packets, ensuring that the contents of the packet cannot be read by unauthorized parties.
2. Integrity: IPsec can provide data integrity by adding a message authentication code (MAC) to IP packets. This ensures that the contents of the packet have not been modified during transmission.
3. Authentication: IPsec can provide authentication by using digital certificates or pre-shared keys to verify the identity of the communicating parties.
4. Replay protection: IPsec can prevent replay attacks by using sequence numbers and timestamps to ensure that packets are not duplicated or replayed.

IPsec operates at the network layer of the OSI model and can be used with any higher-level protocol, including TCP, UDP, and ICMP. IPsec can be implemented in two modes: transport mode and tunnel mode.

Transport mode is used to secure communication between two hosts. In transport mode, only the payload of the IP packet is encrypted and authenticated, leaving the IP header intact. This mode is used when both hosts are trusted and only the payload needs to be secured.

Tunnel mode is used to secure communication between two networks. In tunnel mode, the entire IP packet is encapsulated within a new IP packet, with the new IP header providing the necessary security features. This mode is used when the two networks are not trusted and need to be isolated from each other.

IPsec can be implemented in two different ways: manually or automatically. Manual implementation requires manual configuration of each device on the network, including the configuration of encryption keys, digital certificates, and other security parameters. This method is time-consuming and error-prone, but it provides greater control over the security configuration.

Automatic implementation uses a protocol called Internet Key Exchange (IKE) to automatically negotiate the security parameters between two devices. This method is faster and less error-prone, but it requires a compatible implementation on both devices[1], [2].

IPsec has several components, including:

1. Security Association (SA): A security association is a set of security parameters, including encryption keys, digital certificates, and other security attributes, that are used to secure communication between two devices.
2. Security Policy Database (SPD): The security policy database is a database of security policies that specify how IP packets are to be secured. The SPD is used to determine which security association should be used for a particular IP packet.
3. Key Management Protocol (KMP): The key management protocol is used to manage the encryption keys used by IPsec. The KMP can be used to generate and distribute encryption keys, as well as to revoke and update keys when necessary.
4. Authentication Header (AH): The authentication header is a part of the IPsec protocol that provides data integrity and authentication. The AH adds a MAC to the IP packet, ensuring that the contents of the packet have not been modified during transmission.
5. Encapsulating Security Payload (ESP): The encapsulating security payload is a part of the IPsec protocol that provides confidentiality and data integrity. ESP encrypts the entire IP packet, including the

How IPsec Works:

IPsec works by adding security services to IP packets. These security services are added by the sending device and removed by the receiving device. The security services added to the IP packet depend on the security policy specified in the security policy database.

When a device wants to send an IP packet securely, it first consults the security policy database to determine which security association should be used for the packet. The security policy database contains a list of security policies, each of which specifies the source and destination IP addresses, the type of traffic (such as TCP or UDP), and the security association to be used

Once the sending device has determined the security association to be used, it applies the necessary security services to the IP packet. If the IP packet is being sent in transport mode, only the payload of the IP packet is encrypted and authenticated, leaving the IP header intact. If the IP packet is being sent in tunnel mode, the entire IP packet is encapsulated within a new IP packet, with the new IP header providing the necessary security features.

The receiving device receives the IP packet and removes the security services that were added by the sending device. If the IP packet was encrypted, the receiving device decrypts the packet using the appropriate encryption key. If the IP packet contained a MAC, the receiving device verifies the MAC to ensure that the contents of the packet have not been modified during transmission.

IPsec Components:

1. **Security Association (SA):**

A security association is a set of security parameters that are used to secure communication between two devices. These security parameters include encryption keys, digital certificates, and other security attributes. Each security association is uniquely identified by a security parameter index (SPI) and a destination IP address. The SPI is a number that is used to differentiate between different security associations[3], [4].

2. **Security Policy Database (SPD):**

The security policy database is a database of security policies that specify how IP packets are to be secured. The SPD contains a list of security policies, each of which specifies the source and destination IP addresses, the type of traffic (such as TCP or UDP), and the security association to be used. When a device wants to send an IP packet, it consults the SPD to determine which security association should be used for the packet.

3. **Key Management Protocol (KMP):**

The key management protocol is used to manage the encryption keys used by IPsec. The KMP can be used to generate and distribute encryption keys, as well as to revoke and update keys when necessary. There are several key management protocols available, including Internet Key Exchange (IKE), which is the most commonly used key management protocol.

4. **Authentication Header (AH):**

The authentication header is a part of the IPsec protocol that provides data integrity and authentication. The AH adds a MAC to the IP packet, ensuring that the contents of the packet have not been modified during transmission. The MAC is calculated using a secret key that is shared between the sending and receiving devices. The AH does not provide encryption, so it should be used in conjunction with another IPsec protocol, such as ESP, if confidentiality is required.

5. **Encapsulating Security Payload (ESP):**

The encapsulating security payload is a part of the IPsec protocol that provides confidentiality and data integrity. ESP encrypts the entire IP packet, including the IP header and the payload. ESP also adds a MAC to the IP packet, ensuring that the contents of the packet have not been modified during transmission. ESP can be used in conjunction with AH to provide both confidentiality and authentication. Advantages of IPsec:

1. **Security:**

The primary advantage of IPsec is that it provides security for IP packets transmitted over IP networks.

2. **Flexibility:**

IPsec is a flexible protocol that can be used to secure a wide range of IP traffic. It can be used to secure traffic between two hosts, between two networks, or between a host and a network. It can also be used to secure traffic over different types of networks, including LANs, WANs, and the Internet.

3. **Compatibility:**

IPsec is a widely supported protocol that is available on a variety of different platforms, including routers, firewalls, and servers. This means that IPsec can be used to provide security for a wide range of devices, regardless of the platform they are running on.

4. **Scalability:**

IPsec is a scalable protocol that can be used to secure large networks. It supports the use of multiple security associations, which can be used to secure different types of traffic between different hosts and networks. This makes IPsec a good choice for securing enterprise networks.

5. **Interoperability:**

IPsec is an interoperable protocol that can be used to secure communication between different vendors' devices. This means that IPsec can be used to provide security for networks that use different hardware and software platforms.

6. **Compliance:**

IPsec is a protocol that is commonly used to comply with security standards, such as HIPAA, PCI DSS, and GDPR. These standards require that data in transit is secured using encryption and authentication, which can be achieved using IPsec.

7. **Reduced Costs:**

By using IPsec to secure network traffic, organizations can reduce their costs for maintaining a secure network. This is because IPsec can provide secure communication over the Internet, which can eliminate the need for expensive leased lines or dedicated private networks.

8. Control:

IPsec provides organizations with control over who can access their network resources. By using IPsec, organizations can create secure virtual private networks (VPNs) that allow authorized users to access their network resources from remote locations. This provides organizations with the flexibility to allow remote access to their network resources while maintaining security.

9. Simple Implementation:

IPsec is a relatively easy protocol to implement. This is because it can be implemented at the network layer, which means that it can be used to secure a wide range of applications without requiring changes to the applications themselves. This makes IPsec a good choice for organizations that want to secure their network traffic without incurring high implementation costs.

10. Better Performance:

IPsec can provide better performance than other security protocols, such as SSL or TLS. This is because IPsec operates at the network layer, which means that it can encrypt and authenticate data at wire speed. This can result in faster network performance and lower latency.

IPsec is a powerful protocol that can be used to provide security for IP traffic transmitted over IP networks. It provides a wide range of security features, including encryption, authentication, and data integrity. IPsec is a flexible protocol that can be used to secure a wide range of IP traffic, and it is compatible with a wide range of different platforms. By using IPsec, organizations can reduce their costs for maintaining a secure network, gain better control over their network resources, and comply with security standards. Additionally, IPsec can provide better performance than other security protocols, which can result in faster network performance and lower latency[5], [6].

There are application-specific security mechanisms for a number of application areas, including electronic mail (S/MIME, PGP), client/server (Kerberos), Web access (Secure Sockets Layer), and others. However, users have security concerns that cut across protocol layers. For example, an enterprise can run a secure, private IP network by disallowing links to untrusted sites, encrypting packets that leave the premises, and authenticating packets that enter the premises. By implementing security at the IP level, an organization can ensure secure networking not only for applications that have security mechanisms but also for the many security-ignorant applications.

IP-level security encompasses three functional areas: authentication, confidentiality, and key management. The authentication mechanism assures that a received packet was, in fact, transmitted by the party identified as the source in the packet header. In addition, this mechanism assures that the packet has not been altered in transit. The confidentiality facility enables communicating nodes to encrypt messages to prevent eavesdropping by third parties. The key management facility is concerned with the secure exchange of keys. We begin this chapter with an overview of IP security (IPsec) and an introduction to the IPsec architecture. We then look at each of the three functional areas in detail.

Appendix D reviews Internet protocols.

IP SECURITY OVERVIEW

In 1994, the Internet Architecture Board (IAB) issued a report titled "Security in the Internet Architecture" (RFC 1636). (RFC 1636). The report identified key areas for security mechanisms. Among these were the need to secure the network

IP SECURITY OVERVIEW

Infrastructure from unauthorized monitoring and control of network traffic and the need to secure end-user-to-end-user traffic using authentication and encryption mechanisms.

To provide security, the IAB included authentication and encryption as necessary security features in the next-generation IP, which has been issued as IPv6. Fortunately, these security capabilities were designed to be usable both with the current IPv4 and the future IPv6. This means that vendors can begin offering these features now, and many vendors now do have some IPsec capability in their products. The IPsec specification now exists as a set of Internet standards.

Applications of IPsec

IPsec provides the capability to secure communications across a LAN, across private \sand public WANs, and across the Internet. Examples of its use include:

E. **Secure branch office connectivity over the Internet:** A company can build a secure virtual private network over the Internet or over a public WAN. This enables a business to rely heavily on the Internet and reduce its need for private networks, saving costs and network management overhead.
F. **Secure remote access over the Internet:** An end user whose system is equipped with IP security protocols can make a local call to an Internet Service Provider (ISP) and gain secure access to a company network. This reduces the cost of toll charges for traveling employees and telecommuters.
G. **Establishing extranet and intranet connectivity with partners:** IPsec can be used to secure communication with other organizations, ensuring authentication and confidentiality and providing a key exchange mechanism.
H. **Enhancing electronic commerce security:** Even though some Web and electronic commerce applications have built-in security protocols, the use of IPsec enhances that security. IPsec guarantees that all traffic designated by the network administrator is both encrypted and authenticated, adding an additional layer of security to whatever is provided at the application layer.
I.

The principal feature of IPsec that enables it to support these varied applications is that it can encrypt and/or authenticate all traffic at the IP level. Thus, all distributed applications (including remote logon, client/server, e-mail, file transfer, Web access, and so on) can be secured.

Figure 8.1 is a typical scenario of IPsec usage. An organization maintains LANs at dispersed locations. No secure IP traffic is conducted on each LAN. For traffic offsite, through some sort of private or public WAN, IPsec protocols are used.

These protocols operate in networking devices, such as a router or firewall that connect each LAN to the outside world. The IPsec networking device will typically encrypt and compress all traffic going into the WAN and decrypt and decompress traffic coming from the WAN; these operations are transparent to workstations and servers on the LAN. Secure transmission is also possible with individual users whodial into the WAN. Such user workstations must implement the IPsec protocols to provide security.

IP SECURITY OVERVIEW

TRANSPORT MODE Transport mode provides protection primarily for upper-layer protocols. That is, transport mode protection extends to the payload of an IP packet. Examples include a TCP or UDP segment or an ICMP packet, all of which operate directly above IP in a host protocol stack. Typically, transport mode is used for end-to-end communication between two hosts (e.g., a client and a server, or two workstations). When a host runs AH or ESP over IPv4, the payload is the data that normally follow the IP header. For IPv6, the payload is the data that normally follow both the IP header and any IPv6 extensions headers that are present, with the possible exception of the destination options header, which may be included in the protection. ESP in transport mode encrypts and optionally authenticates the IP payload but not the IP header. AH in transport mode authenticates the IP payload and selected portions of the IP header.

TUNNEL MODE

Tunnel mode provides protection to the entire IP packet. To achieve this, after the AH or ESP fields are added to the IP packet, the entire packet plus security fields is treated as the payload of new outer IP packet with a new outer IP header. The entire original, inner, packet travels through a tunnel from one point of an IP network to another; no routers along the way are able to examine the inner IP header. Because the original packet is encapsulated, the new, larger packet may have totally different source and destination addresses, adding to the security. Tunnel mode is used when one or both ends of a security association (SA) are a security gateway, such as a firewall or router that implements IPsec.

With tunnel mode, a number of hosts on networks behind firewalls may engage in secure communications without implementing IPsec. The unprotected packets generated by such hosts are tunneled through external networks by tunnel mode SAs set up by the IPsec software in the firewall or secure router at the boundary of the local network. Here is an example of how tunnel mode IPsec operates. Host A on a network generates an IP packet with the destination address of host B on another network[7], [8].

This packet is routed from the originating host to a firewall or secure router at the boundary of A’s network. The firewall filters all outgoing packets to determine the need for IPsec processing. If this packet from A to B requires IPsec, the firewall performs IPsec processing and encapsulates the packet with an outer IP header. The source IP address of this outer IP packet is

this firewall, and the destination address may be a firewall that forms the boundary to B's local network. This packet is now routed to B's firewall, with intermediate routers examining only the outer IP header. At B's firewall, the outer IP header is stripped off, and the inner packet is delivered to B. ESP in tunnel mode encrypts and optionally authenticates the entire inner IP packet, including the inner IP header. AH in tunnel mode authenticates the entire \sinner IP packet and selected portions of the outer IP header.

ENCAPSULATING SECURITY PAYLOAD

• Integrity Check Value (variable): A variable-length field (must be an integral number of 32-bit words) that contains the Integrity Check Value computed over the ESP packet minus the Authentication Data field.

When any combined mode algorithm is employed, the algorithm itself is expected to return both decrypted plaintext and a pass/fail indication for the integrity check. For combined mode algorithms, the ICV that would normally appear at the end of the ESP packet when integrity is selected may be omitted. When the ICV is omitted and integrity is selected, it is the responsibility of the combined mode algorithm to encode within the Payload Data an ICV-equivalent means of verifying the integrity of the packet.

An initialization value (IV), or nonce, is present if this is required by the encryption or authenticated encryption algorithm used for ESP. If tunnel mode is being used, then the IPsec implementation may add traffic flow confidentiality (TFC) padding after the Payload Data and before the Padding field, as explained subsequently.

Encryption and Authentication Algorithms

The Payload Data, Padding, Pad Length, and Next Header fields are encrypted by the ESP service. If the algorithm used to encrypt the payload requires cryptographic synchronization data, such as an initialization vector (IV), then these data may be carried explicitly at the beginning of the Payload Data field. If included, an IV is usually not encrypted, although it is often referred to as being part of the ciphertext.

The ICV field is optional. It is present only if the integrity service is selected \sand is provided by either a separate integrity algorithm or a combined mode algorithm that uses an ICV. The ICV is computed after the encryption is performed. This order of processing facilitates rapid detection and rejection of replayed or bogus packets by the receiver prior to decrypting the packet, hence potentially reducing the impact of denial of service (DoS) attacks. It also allows for the possibility of parallel processing of packets at the receiver, i.e., decryption can take place in parallel with integrity checking. Note that because the ICV is not protected by encryption, a keyed integrity algorithm must be employed to compute the ICV.

Padding

The Padding field serves several purposes:

J. If an encryption algorithm requires the plaintext to be a multiple of some number of bytes (e.g., the multiple of a single block for a block cipher), the Padding field

is used to expand the plaintext (consisting of the Payload Data, Padding, Pad Length, and Next Header fields) to the required length.

K. The ESP format requires that the Pad Length and Next Header fields be right aligned within a 32-bit word. Equivalently, the ciphertext must be an integer multiple of 32 bits. The Padding field is used to assure this alignment.

L. Additional padding may be added to provide partial traffic-flow confidentiality by concealing the actual length of the payload.

IP SECURITY

Anti-Replay Service

A replay attack is one in which an attacker obtains a copy of an authenticated packet and later transmits it to the intended destination. The receipt of duplicate, authenticated IP packets may disrupt service in some way or may have some other undesired consequence. The Sequence Number field is designed to thwart such attacks. First, we discuss sequence number generation by the sender, and then we look at how it is processed by the recipient.

When a new SA is established, the sender initializes a sequence number counter to 0. Each time that a packet is sent on this SA, the sender increments the counter and places the value in the Sequence Number field. Thus, the first value to be used is 1. If anti-replay is enabled (the default), the sender must not allow the sequence number to cycle past 232 – 1 back to zero. Otherwise, there would be multiple valid packets with the same sequence number. If the limit of 232 – 1 is reached, the sender should terminate this SA and negotiate a new SA with a new key.

Because IP is a connectionless, unreliable service, the protocol does not guarantee that packets will be delivered in order and does not guarantee that all packets \swill be delivered. Therefore, the IPsec authentication document dictates that the receiver should implement a window of size, with a default. The right edge of the window represents the highest sequence number, so far received for a valid packet. For any packet with a sequence number in the range from to that has been correctly received (i.e., properly authenticated), the corresponding slot in the window is marked

ENCAPSULATING SECURITY PAYLOAD

In the context of IPv6, ESP is viewed as an end-to-end payload; that is, it is not examined or processed by intermediate routers.

Therefore, the ESP header appears after the IPv6 base header and the hop-by-hop, routing, and fragment extension headers.

The destination options extension header could appear before or after the ESP header, depending on the semantics desired. For IPv6, encryption covers the entire transport-level segment plus the ESP trailer plus the destination options extension header if it occurs after the ESP header. Again, authentication covers the ciphertext plus the ESP header.

Transport mode operation may be summarized as follows.

M. At the source, the block of data consisting of the ESP trailer plus the entire transport-layer segment is encrypted and the plaintext of this block is replaced with its ciphertext to form the IP packet for transmission. Authentication is added if this option is selected.
N. The packet is then routed to the destination. Each intermediate router needs to examine and process the IP header plus any plaintext IP extension headers but does not need to examine the ciphertext.
O. The destination node examines and processes the IP header plus any plaintext IP extension headers. Then, on the basis of the SPI in the ESP header, the destination node decrypts the remainder of the packet to recover the plaintext transport-layer segment.

Transport mode operation provides confidentiality for any application that uses it, thus avoiding the need to implement confidentiality in every individual application. One drawback to this mode is that it is possible to do traffic analysis on the transmitted packets[9], [10].

TUNNEL MODE ESP

Tunnel mode ESP is used to encrypt an entire IP packet. For this mode, the ESP header is prefixed to the packet and then the packet plus the ESP trailer is encrypted. This method can be used to counter traffic analysis. Because the IP header contains the destination address and possibly source routing directives and hop-by-hop option information, it is not possible simply to transmit the encrypted IP packet prefixed by the ESP header. Intermediate routers would be unable to process such a packet. Therefore, it is necessary to encapsulate the entire block (ESP header plus ciphertext plus Authentication Data, if present) with a new IP header that will contain sufficient information for routing but not for traffic analysis.

Whereas the transport mode is suitable for protecting connections between hosts that support the ESP feature, the tunnel mode is useful in a configuration that includes a firewall or other sort of security gateway that protects a trusted network from external networks. In this latter case, encryption occurs only between an external host and the security gateway or between two security gateways. This relieves hosts on the internal network of the processing burden of encryption and simplifies the key distribution task by reducing the number of needed keys. Further, it thwarts traffic analysis based on ultimate destination.

REFERENCES:

[1] F. Sönmez and B. EROL, “Survey of Research on IP-DECT and VOIP Systems Safety and a Novel Counter-Measure Approach,” *Int. J. Comput. Appl. Technol. Res.*, 2018, doi: 10.7753/ijcatr0703.1002.

[2] M. Maryanto, M. Maisyaroh, and B. Santoso, “Metode Internet Protocol Security (IPSec) Dengan Virtual Private Network (VPN) Untuk Komunikasi Data,” *PIKSEL Penelit. Ilmu Komput. Sist. Embed. Log.*, 2018, doi: 10.33558/piksel.v6i2.1508.

[3] D. Irawan1 and Fatoni, "Penerapan IP Security pada Jaringan VPN Site to Site di PT. Pertamina Ubeb Adera Pengabuan," *Univ. Bina Darma*, 2018.

[4] S. Rajashree, K. S. Soman, and P. G. Shah, "Security with IP Address Assignment and Spoofing for Smart IOT Devices," in *2018 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2018*, 2018. doi: 10.1109/ICACCI.2018.8554660.

[5] A. Chaturvedi and P. Verma, "Popularly Used Security Protocols on All Layers of Network Communication," *Int. J. Trend Sci. Res. Dev.*, 2018, doi: 10.31142/ijtsrd8312.

[6] Y. Liu, Z. Pang, G. Dan, D. Lan, and S. Gong, "A Taxonomy for the Security Assessment of IP-Based Building Automation Systems: The Case of Thread," *IEEE Trans. Ind. Informatics*, 2018, doi: 10.1109/TII.2018.2844955.

[7] A. Andhyka and F. Badri, "Security Management Implementation in Cloud Server," *Inf. J. Ilm. Bid. Teknol. Inf. dan Komun.*, 2018, doi: 10.25139/inform.v3i2.1050.

[8] H. Kim, T. Kim, and D. Jang, "An intelligent improvement of internet-wide scan engine for fast discovery of vulnerable IoT devices," *Symmetry (Basel).*, 2018, doi: 10.3390/sym10050151.

[9] Z. Zhang *et al.*, "An Overview of Security Support in Named Data Networking," *IEEE Communications Magazine*. 2018. doi: 10.1109/MCOM.2018.1701147.

[10] K. Sudeendra Kumar, S. Sahoo, A. Mahapatra, A. K. Swain, and K. K. Mahapatra, "Security enhancements to system on chip devices for IoT perception layer," in *Proceedings - 2017 IEEE International Symposium on Nanoelectronic and Information Systems, iNIS 2017*, 2018. doi: 10.1109/iNIS.2017.39.