CNS

Unit 1

1. Security Goals of Cryptography

1. Confidentiality:

The degree of confidentiality determines the secrecy of the information. The principle specifies that only the sender and receiver will be able to access the information shared between them. Confidentiality compromises if an unauthorized person is able to access a message.

For example, let us consider sender A wants to share some confidential information with receiver B and the information gets intercepted by the attacker C. Now the confidential information is in the hands of an intruder C.

2. Authentication:

Authentication is the mechanism to identify the user or system or the entity. It ensures the identity of the person trying to access the information. The authentication is mostly secured by using username and password. The authorized person whose identity is preregistered can prove his/her identity and can access the sensitive information.

3. Integrity:

Integrity gives the assurance that the information received is exact and accurate. If the content of the message is changed after the sender sends it but before reaching the intended receiver, then it is said that the integrity of the message is lost.

System Integrity: System Integrity assures that a system performs its intended function in an unimpaired manner, free from deliberate or inadvertent unauthorized manipulation of the system.

Data Integrity: Data Integrity assures that information (both stored and in transmitted packets) and programs are changed only in a specified and authorized manner.

4. Non-Repudiation:

Non-repudiation is a mechanism that prevents the denial of the message content sent through a network. In some cases the sender sends the message and later denies it. But the non-repudiation does not allow the sender to refuse the receiver.

- **Cryptographical Attacks**

1. **Brute force attack**

   Public and private keys play a significant role in encrypting and decrypting the data in a cryptographic system. In a brute force attack, the cybercriminal tries various private keys to decipher an encrypted message or data. If the key size is 8-bit, the possible keys will be 256 (i.e., 28). The cybercriminal must know the algorithm (usually found as open-source programs) to try all the 256 possible keys in this attack technique.

2. **Ciphertext-only attack**

   In this attack vector, the attacker gains access to a collection of ciphertext. Although the attacker cannot access the plaintext, they can successfully determine the ciphertext from the collection. Through this attack technique, the attacker can occasionally determine the key.

3. **Chosen plaintext attack**

   In this attack model, the cybercriminal can choose arbitrary plaintext data to obtain the ciphertext. It simplifies the attacker's task of resolving the encryption key. One well-known example of this type of attack is the differential cryptanalysis performed on block ciphers.

4. **Chosen ciphertext attack**

   In this attack model, the cybercriminal analyzes a chosen ciphertext corresponding to its plaintext. The attacker tries to obtain a secret key or the details about the system. By analyzing the chosen ciphertext and relating it to the plaintext, the attacker attempts to guess the key. Older versions of RSA encryption were prone to this attack.

5. **Known plaintext attack**

   In this attack technique, the cybercriminal finds or knows the plaintext of some portions of the ciphertext using information gathering techniques. Linear cryptanalysis in block cipher is one such example.

- **Monoalphabetic Ciphers**

A monoalphabetic cipher is a type of substitution cipher in which each letter in the plaintext is replaced with a fixed corresponding letter in the cipher-text. In other words, the same substitution key is used consistently throughout the encryption process. Monoalphabetic ciphers are relatively simple and easy to understand, but they are also quite weak in terms of security. They are susceptible to frequency analysis and other cryptanalysis techniques because they preserve the frequency distribution of letters in the plaintext.

One of the most well-known monoalphabetic ciphers is the Caesar cipher, also known as the shift cipher, which is a special case of the monoalphabetic cipher. In a Caesar cipher, each letter in the plaintext is shifted a fixed number of positions down or up the alphabet.

Here's an example of a Caesar cipher with a right shift of 3:

- Plaintext: ABCDEFGHIJKLMNOPQRSTUVWXYZ
- Cipher-text: DEFGHIJKLMNOPQRSTUVWXYZABC

In this example, "A" is replaced with "D," "B" is replaced with "E," and so on. If you wanted to encrypt the word "HELLO," it would become "KHOOR" using this Caesar cipher with a right shift of 3.

Monoalphabetic ciphers, including the Caesar cipher, are not secure for several reasons:

1. Frequency Analysis: Because each letter maps to a fixed letter in the cipher-text, the frequency distribution of letters in the plaintext is preserved in the cipher-text. Attackers can analyze the frequency of letters and patterns in the cipher-text to make educated guesses about the substitutions and potentially decrypt the message.

2. Lack of Key Variability: Monoalphabetic ciphers use a single substitution key for all letters, making them easy to break. In contrast, polyalphabetic ciphers use multiple substitution keys, which adds complexity and makes cryptanalysis more challenging.

3. Vulnerable to Known-Plaintext Attacks: If an attacker knows or can guess some portions of the plaintext, they can deduce the substitutions used in the cipher, which further weakens the security of monoalphabetic ciphers.

Due to their inherent weaknesses, monoalphabetic ciphers are primarily used for educational purposes or in situations where strong security is not a concern. In practice, more advanced and secure encryption techniques, such as polyalphabetic ciphers, block ciphers, and modern cryptographic algorithms like AES (Advanced Encryption Standard), are used to protect sensitive information.

2. Traditional Symmetric Key Ciphers

Traditional symmetric key ciphers are a class of cryptographic algorithms that use the same key for both encryption and decryption. These ciphers have been used for centuries to secure information and are still relevant today. Here are some of the most well-known traditional symmetric key ciphers:

1. Caesar Cipher: The Caesar cipher is one of the earliest known ciphers. It involves shifting each letter in the plaintext by a fixed number of positions down or up the alphabet. For example, a Caesar cipher with a right shift of 3 would turn "HELLO" into "KHOOR."

2. Vigenère Cipher:The Vigenère cipher is an extension of the Caesar cipher that uses a keyword to determine the amount of shifting for each letter in the plaintext. It is a polyalphabetic cipher, which means it employs multiple alphabets for encryption. The keyword is repeated to match the length of the plaintext.

3. Playfair Cipher: The Playfair cipher is a substitution cipher that encrypts pairs of letters at a time. It uses a 5x5 matrix of letters (excluding duplicates) called a "key-table" to perform substitutions based on specific rules.

4. Hill Cipher: The Hill cipher is a mathematical cipher that operates on blocks of letters (usually pairs or triples). It uses a matrix as a key to perform linear transformations on the plaintext. The size of the matrix determines the block size.

5. Transposition Cipher: Transposition ciphers do not change the letters in the plaintext but rearrange their positions. One common example is the Rail Fence cipher, which writes the plaintext diagonally on a set number of "rails" and then reads it off row by row.

6. Substitution Cipher: Substitution ciphers replace each letter or symbol in the plaintext with another letter or symbol. Examples include the Atbash cipher (which substitutes letters with their reverse counterparts) and the simple substitution cipher (which uses a fixed substitution key).

7. Rotor Machines (e.g., Enigma): Rotor machines were used during World War II for secure communications. The most famous rotor machine is the German Enigma machine. These machines used rotating disks (rotors) to perform complex letter substitutions with each keystroke.

8. One-Time Pad: The one-time pad is an unbreakable encryption technique when used correctly. It involves using a random key as long as the plaintext, and each character in the plaintext is combined with the corresponding character in the key using modular addition.

While these traditional symmetric key ciphers played essential roles in the history of cryptography, most of them are no longer considered secure for modern applications. Modern symmetric key ciphers, such as the Advanced Encryption Standard (AES), are designed with higher security standards and are widely used in secure communication and data encryption. AES, for example, is a widely adopted symmetric key cipher that offers strong security and efficiency.

3. Addition and Multiplication Ciphers

Addition and multiplication ciphers are basic forms of encryption techniques that fall under the category of classical symmetric key ciphers. They are relatively simple and not secure for most modern applications but are useful for educational purposes and understanding basic cryptographic concepts.

1. Addition Cipher (Caesar Cipher):

- The addition cipher, often referred to as the Caesar cipher, is one of the oldest and simplest encryption methods. It works by shifting each letter in the plaintext a fixed number of positions down or up the alphabet.

- In this cipher, each letter in the plaintext is replaced by a letter that is a fixed number of positions down or up the alphabet.

- For example, with a right shift of 3 (commonly known as ROT13):

  - Plaintext: ABCDEFGHIJKLMNOPQRSTUVWXYZ

  - Cipher-text: XYZABCDEFGHIJKLMNOPQRSTUVW

- To decrypt the message, you simply reverse the process by shifting the letters in the opposite direction (e.g., left by 3 positions).

- The Caesar cipher is easily breakable through frequency analysis and is considered insecure for any practical application. However, it serves as a fundamental introduction to encryption.

2. Multiplication Cipher (Affine Cipher):

- The multiplication cipher, also known as the affine cipher, is another classical encryption technique that operates on the mathematical relationship of modular arithmetic.

- In this cipher, each letter in the plaintext is mapped to a numerical value (e.g., A=0, B=1, C=2, ..., Z=25). It is then multiplied by a fixed value 'a' and added to another fixed value 'b', modulo 26 (the number of letters in the English alphabet).

- The mathematical formula for encryption is: Ciphertext = (a * Plaintext + b) mod 26

- To decrypt, you need to find the modular multiplicative inverse of 'a' (if it exists), which allows you to recover the plaintext: Plaintext = (a^(-1) * (Ciphertext- b)) mod 26.

- The security of the multiplication cipher depends on the values of 'a' and 'b'. Some combinations can be secure, but others can be easily broken through cryptanalysis.

Both the addition (Caesar) and multiplication (affine) ciphers are considered weak by modern cryptographic standards because they are vulnerable to various attacks, especially brute-force and frequency analysis. For practical purposes, modern symmetric key ciphers like AES (Advanced Encryption Standard) are used, which provide much stronger security and are widely employed in secure communication and data encryption.

---

4. Eulers Totient Function

Euler's totient function, often denoted as ϕ(n), is a mathematical function used in number theory. It calculates the count of positive integers less than or equal to a given integer n that are coprime (relatively prime) to n. Two numbers are considered coprime if their greatest common divisor (GCD) is 1.

The Euler's totient function has several important applications in number theory and cryptography, particularly in the RSA encryption algorithm. Here's how Euler's totient function is defined and some of its properties:

1. Definition: Euler's totient function ϕ(n) is defined as the count of positive integers k ($1 \leq k \leq n$) for which GCD(n, k) = 1. In other words, it counts the number of positive integers that are coprime to n.

2. Euler's Totient Function Formula: Euler's totient function ϕ(n) can be calculated using the following formula:

$\phi(n) = n * (1- 1/p_1) * (1- 1/p_2) * ... * (1- 1/p_r)$

where p_1, p_2, ..., p_r are the distinct prime factors of n.

For example, if n = 12, which factors into 2^2 * 3, then $\phi(12) = 12 * (1- 1/2) * (1- 1/3) = 4$.

3. Properties:

- $\phi(p) = p- 1$ for a prime number p. This is because all positive integers less than p are coprime to p.

- $\phi(p^k) = p^k- p^(k-1)$ for a prime power p^k. This formula counts the positive integers less than or equal to p^k that are coprime to p^k.

- If m and n are coprime (GCD(m, n) = 1), then $\phi(m * n) = \phi(m) * \phi(n)$. This is a consequence of the multiplicative property of Euler's totient function.

4. Applications:

- In RSA cryptography, Euler's totient function is used to generate public and private keys.

- In number theory, Euler's totient function is used to study modular arithmetic and the properties of integers.

- In combinatorics, $\phi(n)$ can be used to count the number of reduced fractions with a given denominator n.

- In solving certain diophantine equations, Euler's totient function plays a role in finding solutions.

Euler's totient function is a fundamental concept in number theory and modular arithmetic, with applications in various areas of mathematics and cryptography. It helps mathematicians and cryptographers understand the properties of integers and design secure encryption algorithms.

---

5. Fermat's Little Theorem

Fermat's Little Theorem is a fundamental result in number theory, named after the French mathematician Pierre de Fermat. It provides a relationship between modular arithmetic and prime numbers. The theorem is stated as follows:

Fermat's Little Theorem: If p is a prime number and a is an integer not divisible by p, then:

$a^(p-1) \equiv 1 \pmod p$

In this statement:

- "a^(p-1)" represents "a raised to the power of (p-1)."

- "≡" denotes "is congruent to."

- "1 (mod p)" means that the result is congruent to 1 when taken modulo p.

Key points and implications of Fermat's Little Theorem:

1. Fermat's Little Theorem applies to prime numbers p and integers a that are not divisible by p (i.e., a is coprime to p).

2. The theorem asserts that if you raise an integer a (not divisible by the prime p) to the power of (p-1) and then take the result modulo p, the outcome will always be congruent to 1.

3. The theorem is widely used in number theory and modular arithmetic, and it serves as the basis for various algorithms and tests, including primality tests.

4. A special case of Fermat's Little Theorem is known as the Fermat Primality Test. It states that if p is a prime number, then for any integer a (1 < a < p), the result of a^(p-1) modulo p is always 1. Therefore, if the result is not 1, p is definitely not prime.

5. Fermat's Little Theorem is a powerful tool for finding modular inverses. If you know p is prime and a is not divisible by p, you can use the theorem to find the modular multiplicative inverse of a modulo p.

6. It has applications in cryptography, particularly in the RSA encryption algorithm, where it helps ensure the security of the encryption process.

7. While Fermat's Little Theorem is a useful tool, it is essential to note that it does not work for composite numbers. There exist composite numbers that satisfy the congruence a^(p-1) ≡ 1 (mod p), even though they are not prime. These composite numbers are called Carmichael numbers.

Fermat's Little Theorem is a foundational result in number theory and plays a crucial role in various branches of mathematics and computer science, especially in the field of cryptography and number theory-based algorithms.

User's Authentication

User authentication is a process by which a system or application verifies the identity of a user to ensure that they are who they claim to be before granting access to specific resources or services. Authentication is a critical aspect of information security and is used to protect sensitive data, systems, and applications from unauthorized access. There are several methods

and factors used in user authentication, depending on the level of security required and the specific use case. Here are some common methods and factors used in user authentication:

1. **Username and Password:**
 - This is the most common form of authentication.
 - Users provide a username (or email address) and a secret password.
 - The system checks whether the entered password matches the stored password associated with the username.
2. **Two-Factor Authentication (2FA) and Multi-Factor Authentication (MFA):**
 - 2FA and MFA require users to provide two or more authentication factors to gain access.
 - Common factors include something you know (password), something you have (e.g., a smartphone or hardware token), and something you are (biometric data like fingerprints or facial recognition).
 - The combination of multiple factors enhances security.
3. **Biometric Authentication:**
 - Biometric authentication uses unique physical or behavioral characteristics to verify identity.
 - Common biometric factors include fingerprints, facial recognition, iris scans, voice recognition, and even keystroke dynamics.
 - Biometric data is difficult to forge, making it a strong form of authentication.
4. **Smart Cards and Tokens:**
 - Smart cards and hardware tokens are physical devices that generate one-time passwords or cryptographic keys.
 - Users must possess the physical token or card and often enter a PIN for authentication.
 - These are commonly used in corporate environments and for secure remote access.
5. **Single Sign-On (SSO):**
 - SSO allows users to log in once and gain access to multiple connected systems or applications without re-entering credentials.
 - SSO systems use various protocols like OAuth, SAML, and OpenID Connect for secure authentication and authorization.
6. **Passwordless Authentication:**
 - Passwordless authentication eliminates the need for traditional passwords.
 - Methods include sending one-time codes via email or SMS, using biometrics, or employing hardware tokens.
 - This approach aims to improve security and user experience.
7. **Risk-Based Authentication:**
 - Risk-based authentication assesses the risk associated with a login attempt.
 - Factors such as location, device, and user behavior are considered to determine if additional authentication steps are necessary.
8. **Captcha and Challenge-Response:**
 - Captcha challenges require users to complete tests or puzzles to prove they are human.
 - Challenge-response authentication presents users with a specific question or challenge that only they should know the answer to.

9. **Time-Based One-Time Passwords (TOTP):**

- TOTP is a form of 2FA that generates one-time passwords that expire after a short period.
- Users typically use a mobile app like Google Authenticator to generate these codes.

10. **Security Questions:**
- Users answer predefined security questions during the registration process.
- This method is less secure due to the potential for users to forget or guess answers.

The choice of authentication method depends on various factors, including the level of security required, user convenience, and the specific use case. In practice, many systems and applications employ a combination of these methods to balance security and usability.

Kerberos

Kerberos is a network authentication protocol and system that provides secure authentication for users and services in a distributed computing environment. It was developed by MIT as part of Project Athena and is widely used for authentication and secure communication in many networks, particularly in Windows Active Directory domains.

Here are the key components and concepts of the Kerberos authentication system:

1. **Authentication Server (AS):**

- The Authentication Server is responsible for initial authentication requests.

- It verifies the identity of users and issues temporary credentials called Ticket Granting Tickets (TGTs).

2. **Ticket Granting Server (TGS):**

- The Ticket Granting Server is responsible for granting access to specific services.

- It accepts TGTs from authenticated users and issues service tickets that allow access to particular network services.

3. **Key Distribution Center (KDC):**

- The Key Distribution Center is a centralized server that combines the functions of the Authentication Server and Ticket Granting Server.

- It stores user credentials (password hashes) and is responsible for authentication and ticket generation.

4. **Realm:**

- A realm is a Kerberos administrative domain. Realms are typically associated with an organization or network.

- Users and services within the same realm trust each other's authentication mechanisms.

5. **Principal:**

- A principal is a unique entity in a Kerberos realm, typically representing a user or a service.

- Each principal has a unique identifier and a secret key associated with it.

6. **Ticket:**

- Tickets are used to prove a user's identity to various services in the network.

- They are time-stamped and encrypted with a shared session key.

- There are two main types of tickets: TGTs (obtained from the AS) and service tickets (obtained from the TGS).

7. **Session Key:**

- A session key is a symmetric encryption key used to secure communication between a user and a service.

- It is generated during the authentication process and is only known to the user and the service.

The Kerberos authentication process involves the following steps:

1. **Authentication:** When a user logs in, they request authentication from the Authentication Server (AS). The AS verifies the user's identity and issues a TGT encrypted with a session key.

2. **Ticket Granting:** The user sends the TGT to the Ticket Granting Server (TGS) and requests access to a specific service. The TGS validates the TGT and issues a service ticket encrypted with a session key for the requested service.

3. **Accessing Services:** The user sends the service ticket to the desired network service. The service validates the ticket and, if it's valid, establishes a secure session with the user using the session key.

Kerberos provides several advantages:

- Strong security: It uses symmetric key cryptography to secure authentication and communication.

- Single sign-on (SSO): Users need to authenticate once to obtain a TGT and can then access multiple services without re-entering credentials.

- Mutual authentication: Both the user and the service authenticate each other during the ticket exchange.

- Time-based tickets: Tickets have a limited validity period, reducing the risk of replay attacks.

Kerberos is widely used in enterprise environments for secure authentication and authorization and plays a significant role in Windows Active Directory domains for user and service authentication.

Advanced Encryption Standard (AES)

Advanced Encryption Standard (AES) is a specification for the encryption of electronic data established by the U.S National Institute of Standards and Technology (NIST) in 2001. AES is widely used today as it is a much stronger than DES and triple DES despite being harder to implement.

Points to remember

- AES is a block cipher.
- The key size can be 128/192/256 bits.
- Encrypts data in blocks of 128 bits each.

That means it takes 128 bits as input and outputs 128 bits of encrypted cipher text as output. AES relies on substitution-permutation network principle which means it is performed using a series of linked operations which involves replacing and shuffling of the input data.

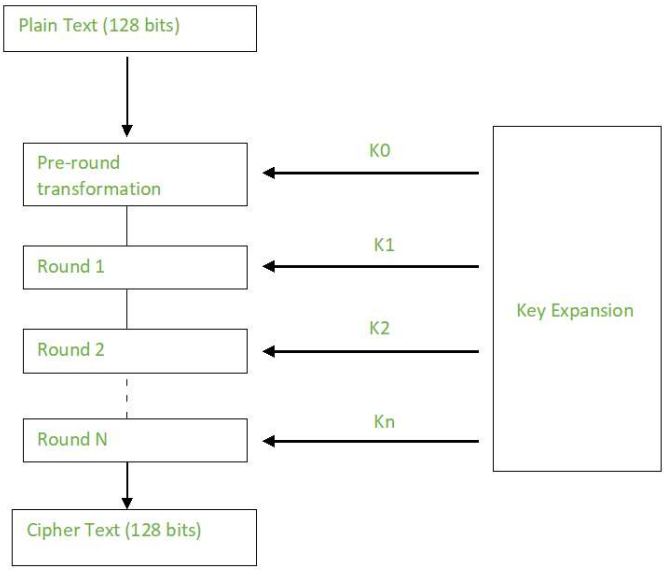
Working of the cipher :
AES performs operations on bytes of data rather than in bits. Since the block size is 128 bits, the cipher processes 128 bits (or 16 bytes) of the input data at a time.

The number of rounds depends on the key length as follows :

- 128 bit key – 10 rounds
- 192 bit key – 12 rounds
- 256 bit key – 14 rounds

Creation of Round keys :

A Key Schedule algorithm is used to calculate all the round keys from the key. So the initial key is used to create many different round keys which will be used in the corresponding round of the encryption.



Encryption :

AES considers each block as a 16 byte (4 byte x 4 byte = 128) grid in a column major arrangement.

```
[ b0 | b4 | b8 | b12 |
| b1 | b5 | b9 | b13 |
| b2 | b6 | b10| b14 |
| b3 | b7 | b11| b15 ]
```

Each round comprises of 4 steps :

- SubBytes
- ShiftRows
- MixColumns
- Add Round Key

The last round doesn't have the MixColumns round.

The SubBytes does the substitution and ShiftRows and MixColumns performs the permutation in the algorithm.

SubBytes :
This step implements the substitution.

In this step each byte is substituted by another byte. Its performed using a lookup table also called the S-box. This substitution is done in a way that a byte is never substituted by itself and also not substituted by another byte which is a compliment of the current byte. The result of this step is a 16 byte (4 x 4) matrix like before.

The next two steps implement the permutation.

ShiftRows :
This step is just as it sounds. Each row is shifted a particular number of times.

- The first row is not shifted
- The second row is shifted once to the left.
- The third row is shifted twice to the left.
- The fourth row is shifted thrice to the left.

(A left circular shift is performed.)

```
[ b0  | b1  | b2  | b3  ]          [ b0  | b1  | b2  | b3  ]
| b4  | b5  | b6  | b7  |    ->    | b5  | b6  | b7  | b4  |
```

```
| b8  | b9  | b10 | b11 |          | b10 | b11 | b8  | b9  |
[ b12 | b13 | b14 | b15 ]          [ b15 | b12 | b13 | b14 ]
```

MixColumns :

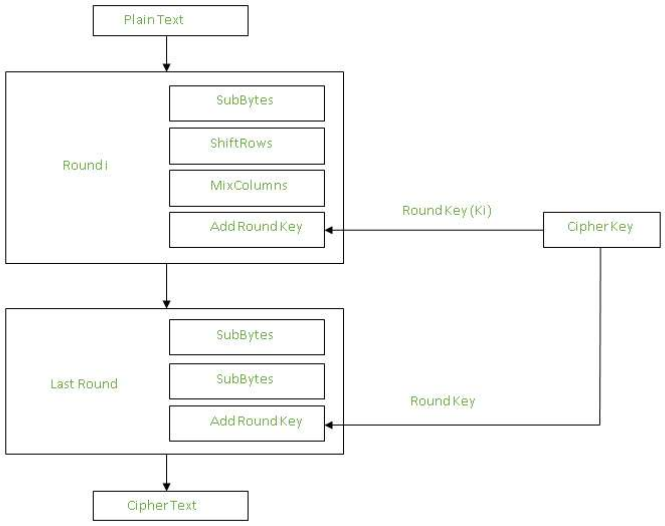
This step is basically a matrix multiplication. Each column is multiplied with a specific matrix and thus the position of each byte in the column is changed as a result.

This step is skipped in the last round.

```
[ c0 ]         [ 2  3  1  1 ]  [ b0 ]
| c1 |  =      | 1  2  3  1 |     | b1 |
| c2 |       | 1  1  2  3 |     | b2 |
[ c3 ]       [ 3  1  1  2 ]     [ b3 ]
```

Add Round Keys :

Now the resultant output of the previous stage is XOR-ed with the corresponding round key. Here, the 16 bytes is not considered as a grid but just as 128 bits of data.



After all these rounds 128 bits of encrypted data is given back as output. This process is repeated until all the data to be encrypted undergoes this process.

Decryption :

The stages in the rounds can be easily undone as these stages have an opposite to it which when performed reverts the changes.Each 128 blocks goes through the 10,12 or 14 rounds depending on the key size.

The stages of each round in decryption is as follows :

- Add round key
- Inverse MixColumns
- ShiftRows
- Inverse SubByte

The decryption process is the encryption process done in reverse so i will explain the steps with notable differences.

Inverse MixColumns :

This step is similar to the MixColumns step in encryption, but differs in the matrix used to carry out the operation.

```
[ b0 ]         [ 14  11  13  9  ]  [ c0 ]
| b1 |  =      | 9   14  11  13 |     | c1 |
| b2 |       | 13  9   14  11 |     | c2 |
[ b3 ]         [ 11  13  9   14 ]     [ c3 ]
```

Inverse SubBytes :

Inverse S-box is used as a lookup table and using which the bytes are substituted during decryption.

Applications:

AES is widely used in many applications which require secure data storage and transmission. Some common use cases include:

- **Wireless security:** AES is used in securing wireless networks, such as Wi-Fi networks, to ensure data confidentiality and prevent unauthorized access.

- **Database Encryption:** AES can be applied to encrypt sensitive data stored in databases. This helps protect personal information, financial records, and other confidential data from unauthorized access in case of a data breach.
- **Secure communications:** AES is widely used in protocols like such as internet communications, email, instant messaging, and voice/video calls.It ensures that the data remains confidential.
- **Data storage:** AES is used to encrypt sensitive data stored on hard drives, USB drives, and other storage media, protecting it from unauthorized access in case of loss or theft.
- **Virtual Private Networks (VPNs):** AES is commonly used in VPN protocols to secure the communication between a user's device and a remote server. It ensures that data sent and received through the VPN remains private and cannot be deciphered by eavesdroppers.
- **Secure Storage of Passwords:** AES encryption is commonly employed to store passwords securely. Instead of storing plaintext passwords, the encrypted version is stored. This adds an extra layer of security and protects user credentials in case of unauthorized access to the storage.
- **File and Disk Encryption:** AES is used to encrypt files and folders on computers, external storage devices, and cloud storage. It protects sensitive data stored on devices or during data transfer to prevent unauthorized access.

Summary :

AES instruction set is now integrated into the CPU (offers throughput of several GB/s)to improve the speed and security of applications that use AES for encryption and decryption. Even though its been 20 years since its introduction we have failed to break the AES algorithm as it is infeasible even with the current technology. Till date the only vulnerability remains in the implementation of the algorithm.

Last Updated : 22 May, 2023

88

Message Authentication Requirements

Data is prone to various attacks. One of these attacks includes message authentication. This threat arises when the user does not have any information about the originator of the message. Message authentication can be achieved using cryptographic methods which further make use of keys.

Authentication Requirements:

- **Revelation:** It means releasing the content of the message to someone who does not have an appropriate cryptographic key.
- **Analysis of Traffic:** Determination of the pattern of traffic through the duration of connection and frequency of connections between different parties.
- **Deception:** Adding out of context messages from a fraudulent source into a communication network. This will lead to mistrust between the parties communicating and may also cause loss of critical data.
- **Modification in the Content:** Changing the content of a message. This includes inserting new information or deleting/changing the existing one.
- **Modification in the sequence:** Changing the order of messages between parties. This includes insertion, deletion, and reordering of messages.
- **Modification in the Timings:** This includes replay and delay of messages sent between different parties. This way session tracking is also disrupted.
- **Source Refusal:** When the source denies being the originator of a message.
- **Destination refusal:** When the receiver of the message denies the reception.

Message Authentication Functions:

DEFINITION

block cipher

By **TechTarget Contributor**

What is a block cipher?

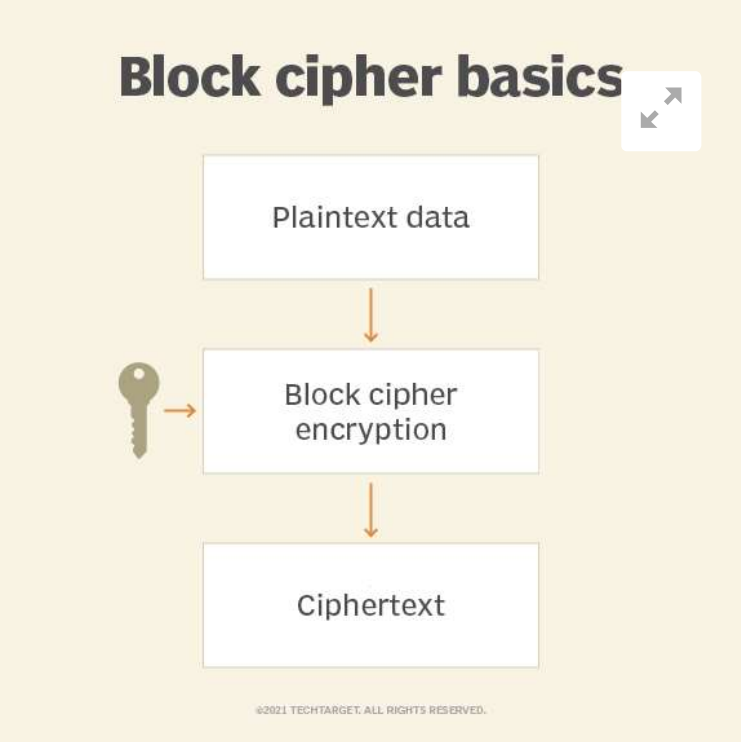
A block cipher is a method of encrypting data in blocks to produce ciphertext using a cryptographic key and algorithm. The block cipher processes fixed-size blocks simultaneously, as opposed to a stream cipher, which encrypts data one bit at a time. Most modern block ciphers are designed to encrypt data in fixed-size blocks of either 64 or 128 bits.

How does a block cipher work?

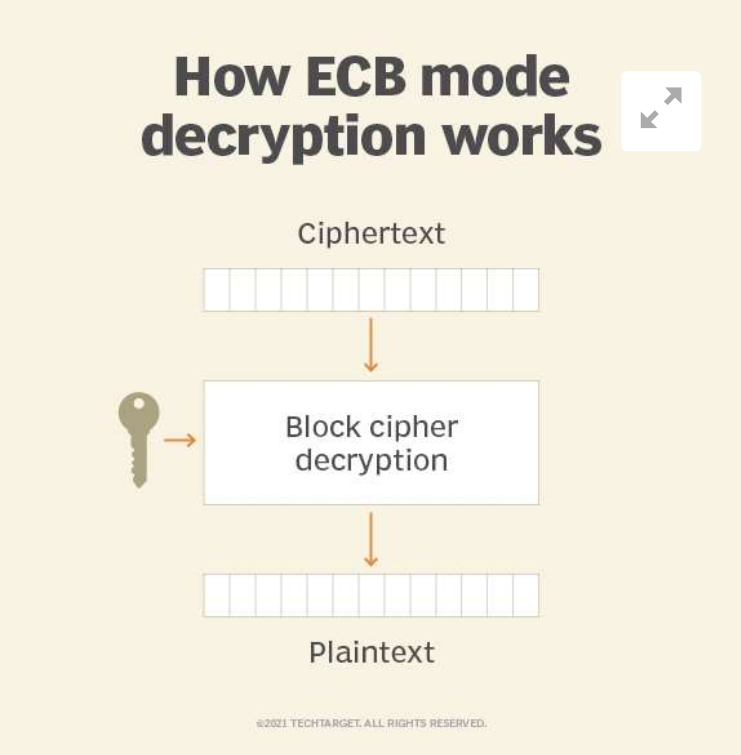
A block cipher uses a symmetric key and algorithm to encrypt and decrypt a block of data. A block cipher requires an initialization vector (IV) that is added to the input plaintext in order to increase the keyspace of the cipher and make it more difficult to use brute force to break the key. The IV is derived from a random number generator, which is combined with text in the first block and the key to ensure all subsequent blocks result in ciphertext that does not match that of the first encryption block.

The *block size* of a block cipher refers to the number of bits that are processed together. Data Encryption Standard (DES) and Advanced Encryption Standard (AES) are both symmetric block ciphers.

The DES block cipher was originally designed by IBM in 1975 and consisted of 64-bit blocks and a 56-bit key. This cipher is not considered secure anymore, due to the short key size, and was replaced in 1998 by AES. AES uses a 128-bit block size and a 128-, 192- or 256-bit key size.



How a block cipher works



Electronic codebook mode decryption

What are the different modes of operation in block cipher?

Block ciphers only encrypt messages that are the same size as their block length, so each block of plaintext with more or less blocks needs to be encrypted separately. The following block cipher modes of operation define how these blocks are encrypted:

- **Electronic codebook (ECB) mode.** ECB mode is used to electronically code messages as their plaintext form. It is the simplest of all block cipher modes of operation. It does not add any randomness to the key stream, and it is the only mode that can be used to encrypt a single-bit stream. This means that each plaintext symbol, such as a character from the plaintext alphabet, is converted into a ciphertext symbol using the cipher's key and a substitution alphabet. Each plaintext block is encrypted independently of all the other blocks. If a plaintext block is only 8 bytes, only 8 bytes of the key are used; if a plaintext block is 100 bytes, all 100 bytes of the key are used.

- **Cipher block chaining (CBC) mode.** CBC mode is a method of encrypting data that ensures that each block of plaintext is combined with the previous ciphertext block before being encrypted. The symmetric key algorithm creates a ciphertext that depends on all plaintext blocks processed before it in a data stream. This is done to ensure that each block of the ciphertext is dependent on all of the previous blocks. Each plaintext block is XORed (exclusive OR) with the previous ciphertext block before being encrypted with the cipher algorithm. CBC mode is used in a variety of security applications. For example, Secure Sockets Layer/Transport Layer Security uses CBC mode to encrypt data that is transferred over the internet.
- **Ciphertext feedback (CFB) mode.** In contrast to CBC mode, which encrypts a set number of bits of plaintext at a time, it is sometimes necessary to encrypt and transfer plaintext values instantly, one at a time. Like CBC, CFB also uses an IV. CFB uses a block cipher as a component of a random number generator. In CFB mode, the previous ciphertext block is encrypted, and the output is XORed with the current plaintext block to create the current ciphertext block. The XOR operation conceals plaintext patterns.
- **Output feedback (OFB) mode.** OFB mode can be used with any block cipher and is similar in some respects to CBC mode. It uses a feedback mechanism, but instead of XORing the previous block of ciphertext with the plaintext before encryption, in OFB mode, the previous block of ciphertext is XORed with the plaintext after it is encrypted.
- **Counter (CTR) mode.** CTR mode uses a block chaining mode of encryption as a building block. The process of encrypting data is performed by XORing the plaintext with a sequence of pseudorandom values, each of which is generated from the ciphertext using a feedback function. The CTR encryption process can be visualized as a series of XORs between blocks of plaintext and corresponding blocks of ciphertext.

Authenticated encryption with additional data modes

The following modes provide message encryption and can supply additional data -- including sequence number or header -- that is not included in the ciphertext:

- **Galois/Counter Mode (GCM).** In GCM mode, blocks are combined with an IV and encrypted with AES. The result is then XORed with the plaintext to generate the

ciphertext.

- **Counter Mode with CBC Message Authentication Code Protocol (CCMP).** CCMP mode is for use with AES. It uses a 128-bit block size and a 128-bit key size and is capable of handling messages up to 16 bytes. CCMP mode was designed to address some of the problems with the CBC mode of operation in which the same block of plaintext can encrypt to different ciphertexts.
- **Synthetic IV (SIV).** SIV is a byte-oriented (8-bit) substitution-permutation network AES algorithm. It takes a plaintext message and a secret key and encrypts the plaintext into ciphertext. However, it differs from other cipher modes in that it does not use a random key stream; instead, it uses a fixed key stream that is generated from a pseudorandom number generator.
- **AES-GCM-SIV.** AES-GCM-SIV is a combination of the AES block cipher and GCM, with the added security feature of a SIV. This enables more messages to be encrypted with the same key than with GCM-SIV.

Learn how cloud providers are tackling multi-cloud key challenges using key management as a service.

This was last updated in May 2021

Continue Reading About block cipher

- The difference between AES and DES encryption
- Weighing double key encryption challenges, payoffs
- Symmetric vs. asymmetric encryption: Decipher the differences
- Cryptography basics: Symmetric key encryption algorithms

Related Terms

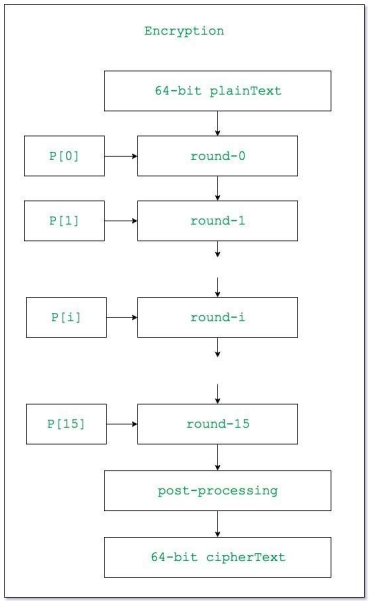
Blowfish Algorithm with Examples

Blowfish is an encryption technique designed by **Bruce Schneier** in 1993 as an alternative to DES Encryption Technique. It is significantly faster than DES and provides a good encryption rate with no effective cryptanalysis technique found to date. It is one of the first, secure block cyphers not subject to any patents and hence freely available for anyone to use. It is symmetric block cipher algorithm.

1. **blockSize**: 64-bits
2. **keySize**: 32-bits to 448-bits variable size
3. **number of subkeys**: 18 [P-array]
4. **number of rounds**: 16
5. **number of substitution boxes**: 4 [each having 512 entries of 32-bits each]

Blowfish Encryption Algorithm

The entire encryption process can be elaborated as:

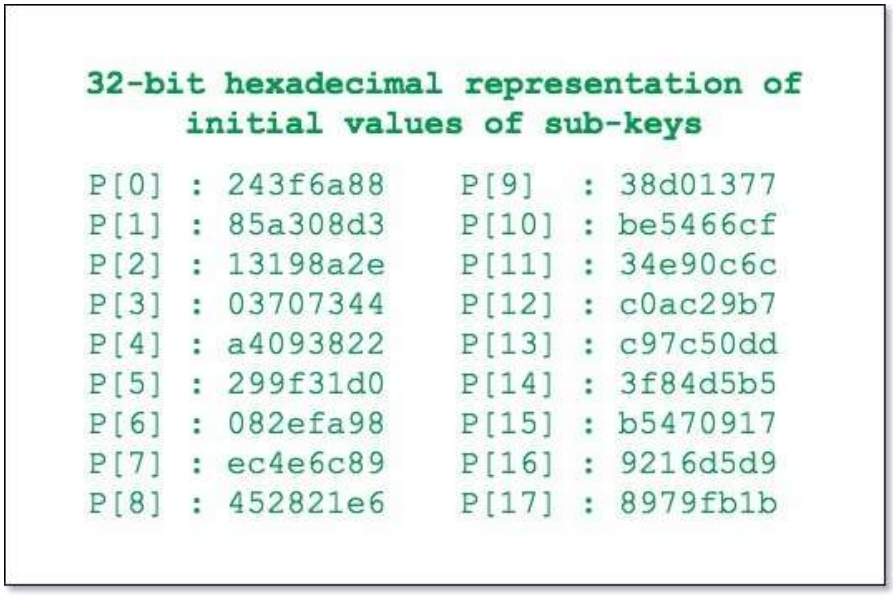


Lets see each step one by one:

Step1: Generation of subkeys:

- 18 subkeys{P[0]...P[17]} are needed in both encryption as well as decryption process and the same subkeys are used for both the processes.
- These 18 subkeys are stored in a P-array with each array element being a 32-bit entry.
- It is initialized with the digits of pi(?).
- The hexadecimal representation of each of the subkeys is given by:

```
P[0] = "243f6a88"
P[1] = "85a308d3"
.
.
.
P[17] = "8979fb1b"
```



- Now each of the subkey is changed with respect to the input key as:

```
P[0] = P[0] xor 1st 32-bits of input key
P[1] = P[1] xor 2nd 32-bits of input key
.
.
.
P[i] = P[i] xor (i+1)th 32-bits of input key
(roll over to 1st 32-bits depending on the key length)
.
.
.
P[17] = P[17] xor 18th 32-bits of input key
(roll over to 1st 32-bits depending on key length)
```

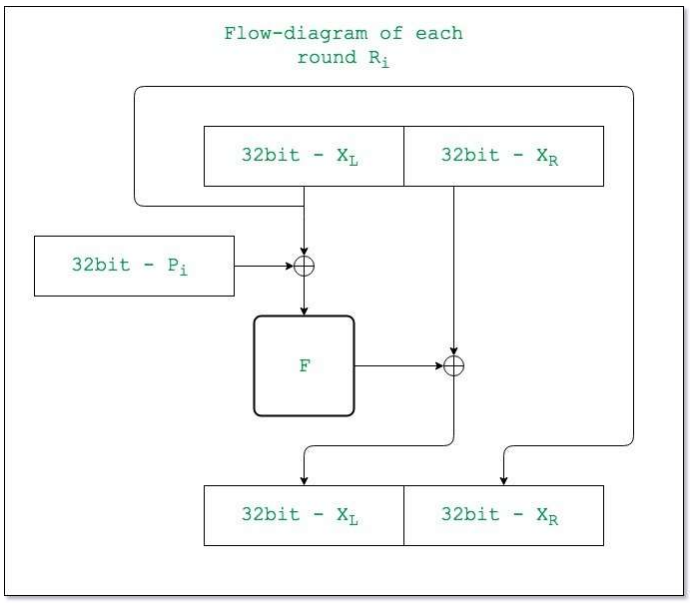
The resultant P-array holds 18 subkeys that is used during the entire encryption process

Step2: initialise Substitution Boxes:

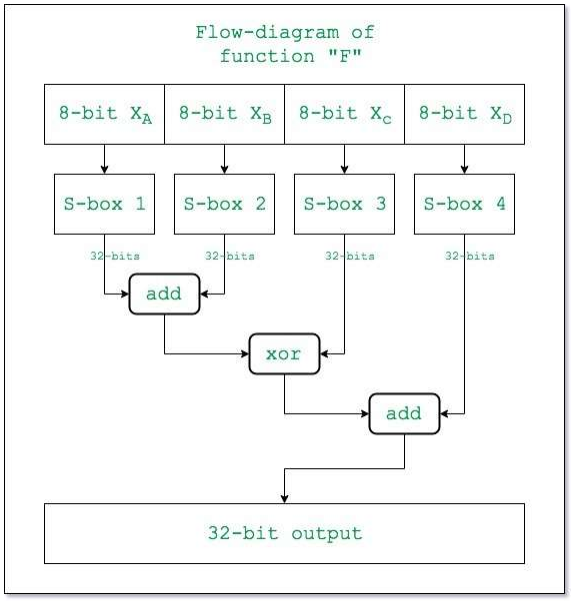
- 4 Substitution boxes(S-boxes) are needed{S[0]...S[4]} in both encryption aswell as decryption process with each S-box having 256 entries{S[i][0]...S[i][255], 0&lei&le4} where each entry is 32-bit.
- It is initialized with the digits of pi(?) after initializing the P-array. You may find the **s-boxes** in here!

Step3: Encryption:

- The encryption function consists of two parts:
  a. Rounds: The encryption consists of 16 rounds with each round(Ri) taking inputs the plainText(P.T.) from previous round and corresponding subkey(Pi). The description of each round is as follows:

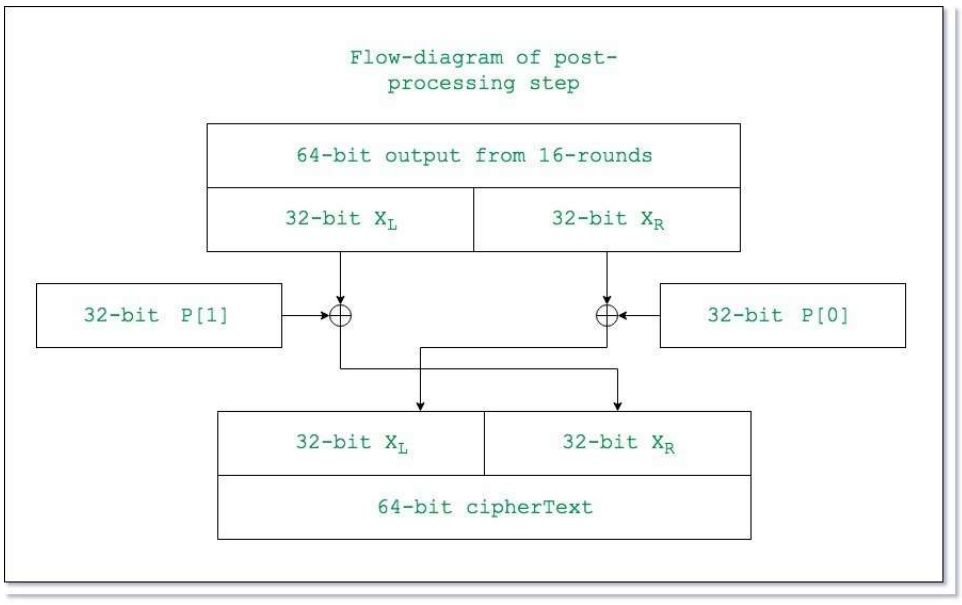


The description of the function ” F ” is as follows:



Here the function “add” is addition modulo 2^32.

b. Post-processing: The output after the 16 rounds is processed as follows:



Below is a Java Program to demonstrate Blowfish encryption:

Java

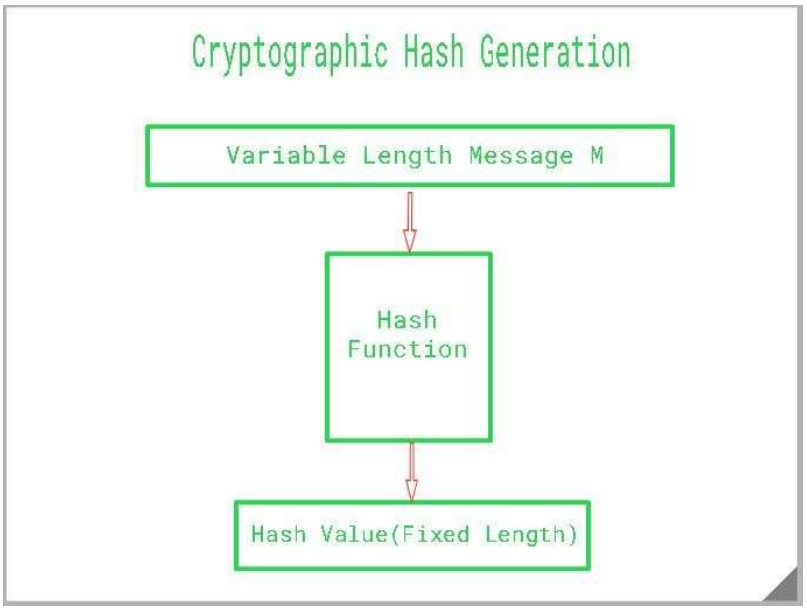
```
// Java Program to demonstrate Blowfish encryption

import java.util.*;

public class Main {
```

Cryptographic Hash Function in Java

Cryptographic Hash is a Hash function that takes random size input and yields a fixed-size output. It is easy to calculate but challenging to retrieve the original data. It is strong and difficult to duplicate the same hash with unique inputs and is a one-way function so revert is not possible. Hashing is also known by different names such as Digest, Message Digest, Checksum, etc.



Properties Of Cryptography Hash Function

The ideal cryptographic hash function has the following main properties:

1. **Deterministic:** This means that the same message always results in the same hash.
2. **Quick:** It is quick to compute the hash value for any given message.
3. **Avalanche Effect:** This means that every minor change in the message results in a major change in the hash value.

4. **One-Way Function:** You cannot reverse the cryptographic hash function to get to the data.
5. **Collision Resistance:** It is infeasible to find two different messages that produce the same hash value.
6. **Pre-Image Resistance:** The hash value shouldn't be predictable from the given string and vice versa.
7. **Second Pre-Image Resistance:** Given an input, it should be difficult to find another input that has the same hash value.

We often hear the term Cracking a Hash, there are a couple of ways to do that:

- Find an algorithm to generate a collision between two hashes. The more advance the algorithm is, the more difficult it is to crack the hash.
- Another way is to find an algorithm to identify a unique and different input that will produce a given hash. It is similar to a collision, but instead of colliding, we are focusing on finding the input using an algorithm.
- Some common hashes we still use today that are considered "cracked" from a cryptographic point of view are MD5(Message-Digest Algorithm) and SHA-1(Secure Hash Algorithm 1). Keep in mind that these are technically broken Hashes and never use for security purposes.

How to create a Cryptographic Hash

- Create a random salt value using SecureRandom class, SecureRandom class generates strong random values. The engineNextBytes(byte[] bytes) method is used to generate a user-specified number of random bytes.
- Convert two sets of bytes into one using ByteArrayOutputStream class and create it to ByteArray.
- Create an instance of a message-digest passing SHA2_ALGORITHM which returns a hash of the given input value.
- UUID is used to genmessage-digested to a string and passed as input.
- The returned object can be converted to a hex binary format using DatatypeConverter.

Java

```
// Java program to demonstrate
// how to create a Hash

package java_cryptography;

import java.io.ByteArrayOutputStream;
import java.security.MessageDigest;
import java.util.UUID;
import javax.xml.bind.DatatypeConverter;
import sun.security.provider.SecureRandom;

public class Hashing {

    // Initializing the final string variable
    private static final String SHA2_ALGORITHM
        = "SHA-256";

    // Creating a random salt value to prevent
    // attacks from the Rainbow table.
    public static byte[] Creating_Random_Salt()
    {
        byte[] salt = new byte[16];
        SecureRandom secure_random
```

Digital signatures are the public-key primitives of message authentication. In the physical world, it is common to use handwritten signatures on handwritten or typed messages. They are used to bind signatory to the message.

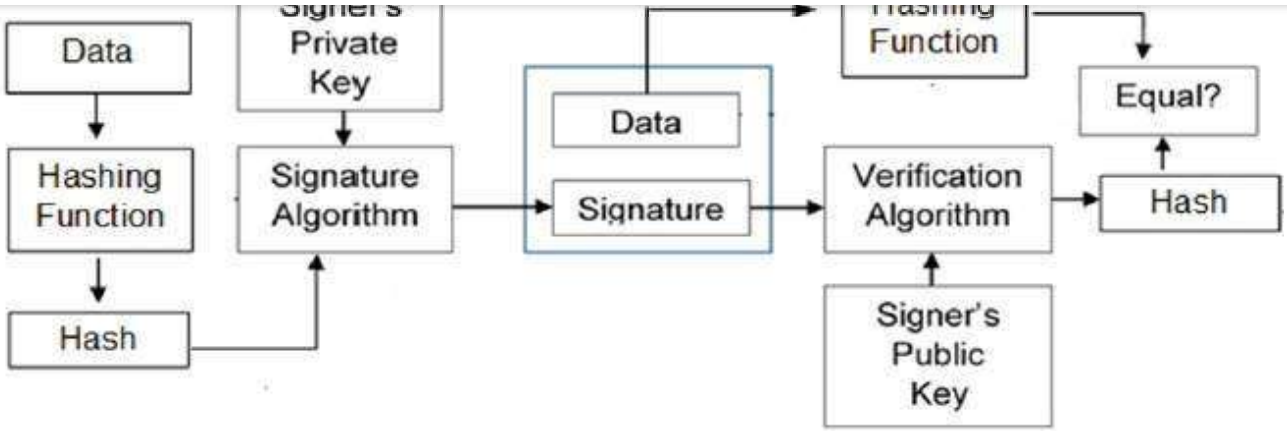
Similarly, a digital signature is a technique that binds a person/entity to the digital data. This binding can be independently verified by receiver as well as any third party.

Digital signature is a cryptographic value that is calculated from the data and a secret key known only by the signer.

In real world, the receiver of message needs assurance that the message belongs to the sender and he should not be able to repudiate the origination of that message. This requirement is very crucial in business applications, since likelihood of a dispute over exchanged data is very high.

Model of Digital Signature

As mentioned earlier, the digital signature scheme is based on public key cryptography. The model of digital signature scheme is depicted in the following illustration –



The following points explain the entire process in detail –

Each person adopting this scheme has a public-private key pair.

Generally, the key pairs used for encryption/decryption and signing/verifying are different. The private key used for signing is referred to as the signature key and the public key as the verification key.

Signer feeds data to the hash function and generates hash of data.

Hash value and signature key are then fed to the signature algorithm which produces the digital signature on given hash. Signature is appended to the data and then both are sent to the verifier.

Verifier feeds the digital signature and the verification key into the verification algorithm. The verification algorithm gives some value as output.

Verifier also runs same hash function on received data to generate hash value.

For verification, this hash value and output of verification algorithm are compared. Based on the comparison result, verifier decides whether the digital signature is valid.

Since digital signature is created by 'private' key of signer and no one else can have this key; the signer cannot repudiate signing the data in future.

It should be noticed that instead of signing data directly by signing algorithm, usually a hash of data is created. Since the hash of data is a unique

efficiency of the scheme.

Let us assume RSA is used as the signing algorithm. As discussed in public key encryption chapter, the encryption/signing process using RSA involves modular exponentiation.

Signing large data through modular exponentiation is computationally expensive and time consuming. The hash of the data is a relatively small digest of the data, hence **signing a hash is more efficient than signing the entire data**.

Importance of Digital Signature

Out of all cryptographic primitives, the digital signature using public key cryptography is considered as very important and useful tool to achieve information security.

Apart from ability to provide non-repudiation of message, the digital signature also provides message authentication and data integrity. Let us briefly see how this is achieved by the digital signature –

Message authentication – When the verifier validates the digital signature using public key of a sender, he is assured that signature has been created only by sender who possess the corresponding secret private key and no one else.

Data Integrity – In case an attacker has access to the data and modifies it, the digital signature verification at receiver end fails. The hash of modified data and the output provided by the verification algorithm will not match. Hence, receiver can safely deny the message assuming that data integrity has been breached.

Non-repudiation – Since it is assumed that only the signer has the knowledge of the signature key, he can only create unique signature on a given data. Thus the receiver can present data and the digital signature to a third party as evidence if any dispute arises in the future.

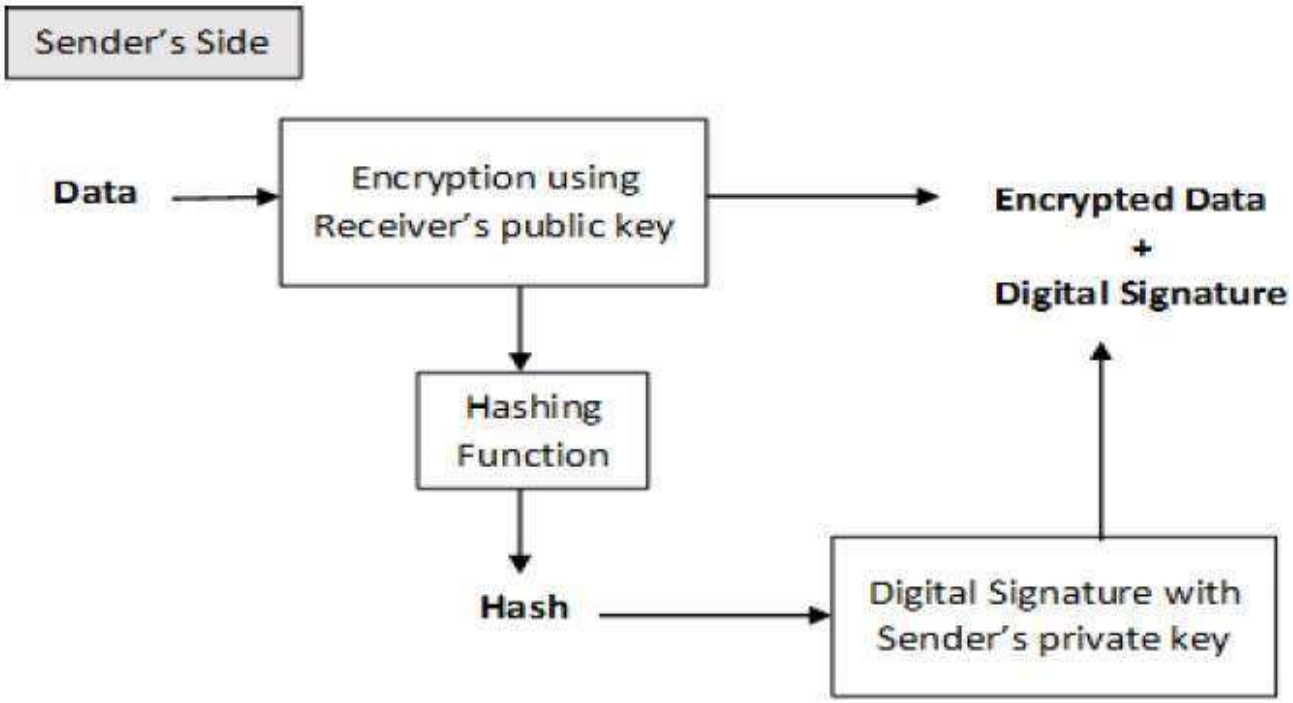
By adding public-key encryption to digital signature scheme, we can create a cryptosystem that can provide the four essential elements of security namely – Privacy, Authentication, Integrity, and Non-repudiation.

In many digital communications, it is desirable to exchange an encrypted messages than plaintext to achieve confidentiality. In public key encryption scheme, a public (encryption) key of sender is available in open domain, and hence anyone can spoof his identity and send any encrypted message to the receiver.

This makes it essential for users employing PKC for encryption to seek digital signatures along with encrypted data to be assured of message authentication and non-repudiation.

This can archived by combining digital signatures with encryption scheme. Let us briefly discuss how to achieve this requirement. There are **two possibilities, sign-then-encrypt** and **encrypt-then-sign**.

However, the crypto system based on sign-then-encrypt can be exploited by receiver to spoof identity of sender and sent that data to third party. Hence, this method is not preferred. The process of encrypt-then-sign is more reliable and widely adopted. This is depicted in the following illustration –



The receiver after receiving the encrypted data and signature on it, first verifies the signature using sender's public key. After ensuring the validity of the signature, he then retrieves the data through decryption using his private key.

DEFINITION

elliptical curve cryptography (ECC)

By **Andrew Froehlich,** West Gate Networks

What is elliptical curve cryptography (ECC)?

Elliptical curve cryptography (ECC) is a public key encryption technique based on elliptic curve theory that can be used to create faster, smaller and more efficient cryptographic keys.

ECC is an alternative to the Rivest-Shamir-Adleman (RSA) cryptographic algorithm and is most often used for digital signatures in cryptocurrencies, such as Bitcoin and Ethereum, as well as one-way encryption of emails, data and software.

An elliptic curve is not an ellipse, or oval shape, but it is represented as a looping line intersecting two axes, which are lines on a graph used to indicate the position of a point. The curve is completely symmetric, or mirrored, along the x-axis of the graph.

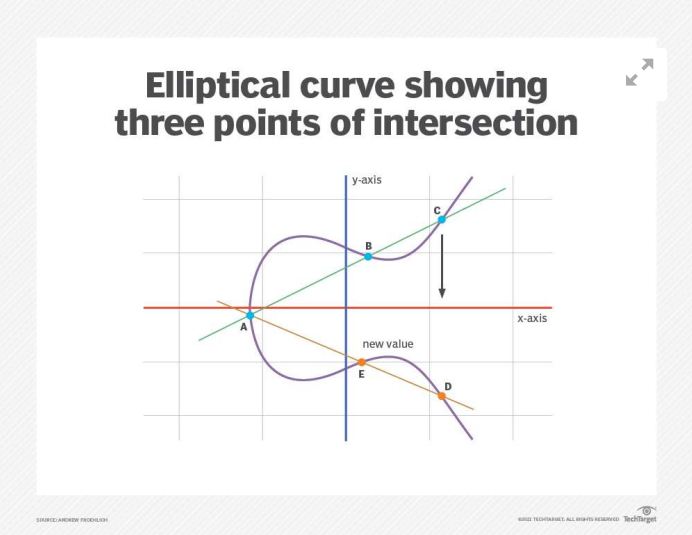
Public key cryptography systems, like ECC, use a mathematical process to merge two distinct keys and then use the output to encrypt and decrypt data. One is a public key that is known to anyone, and the other is a private key that is only known by the sender and receiver of the data.

ECC generates keys through the properties of an elliptic curve equation instead of the traditional method of generation as the product of large prime numbers. From a cryptographic perspective, the points along the graph can be formulated using the following equation:

$y^2=x^3 + ax + b$

ECC is like most other public key encryption methods, such as the RSA algorithm and Diffie-Hellman. Each of these cryptography mechanisms uses the concept of a one-way, or trapdoor, function. This means that a mathematical equation with a public and private key can be used to easily get from point A to point B. But, without knowing the private key and depending on the key size used, getting from B to A is difficult, if not impossible, to achieve.

ECC is based on the properties of a set of values for which operations can be performed on any two members of the group to produce a third member, which is derived from points where the line intersects the axes as shown with the green line and three blue dots in the below diagram labeled A, B and C. Multiplying a point on the curve by a number produces another point on the curve (C). Taking point C and bringing it to the mirrored point on the opposite side of the x-axis produces point D. From here, a line is drawn back to our original point A, creating an intersection at point E. This process can be completed *n* number of times within a defined max value. The *n* is the private key value, which indicates how many times the equation should be run, ending on the final value that is used to encrypt and decrypt data. The maximum defined value of the equation relates to the key size used.



This elliptical curve diagram shows three intersection points.

Comparing RSA vs. elliptical curve cryptography

ECC can yield a level of security that requires fewer computing resources to encrypt and decrypt data compared to alternative methods, like RSA. For example, ECC using a 256-bit key would require a 3,072-bit RSA key to achieve equivalent protection. Because ECC establishes equivalent security with lower computing power and battery resource usage than RSA, it is widely used for mobile applications and internet of things (IoT) devices with limited central processing unit (CPU) resources.

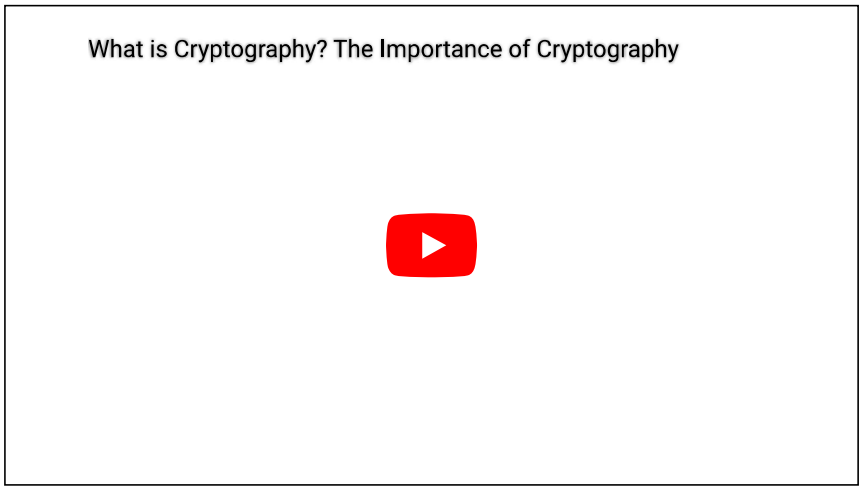
ECC offers several benefits compared to RSA:

- It operates on devices with low CPU and memory resources.
- It encrypts and decrypts faster.
- Larger key sizes can be used without significantly increasing the key size or CPU and memory requirements.

How secure is elliptical curve cryptography?

ECC is thought to be highly secure if the key size used is large enough. The U.S. government requires the use of ECC with a key size of either 256 or 384 bits for internal communications, depending on the sensitivity level of the information being transmitted.

But ECC is not necessarily any more or less secure compared to alternatives such as RSA. The primary benefit of ECC is the inherent efficiencies gained when encrypting and decrypting data.



The history of elliptical curve cryptography

The properties and functions of elliptic curves in mathematics have been studied for more than 150 years. Their use within cryptography was first proposed in 1985, separately by Neal Koblitz from the University of Washington and Victor Miller at IBM.

ECC was first developed by Certicom, a mobile e-business security provider, and was then licensed by Hifn, a manufacturer of integrated circuitry and network security products. Vendors, including 3Com, Cylink Corp., Motorola, Pitney Bowes, Siemens, TRW Inc. (acquired by Northrop Grumman) and Verifone, supported ECC in their products.

The use of ECC in public and private sectors has increased over the past few years. While RSA continues to be more widely used and is easier to understand compared to ECC, the efficiency benefits of ECC make it appealing for many enterprise use cases. These include speeding up secure access to Secure Sockets Layer-encrypted websites and streaming encrypted data from IoT devices with limited computing power.

Data encryption is a must-have for cloud environments, but organizations face many challenges as they try to secure their data. Learn what cloud encryption strategies organizations can implement to overcome these limitations.

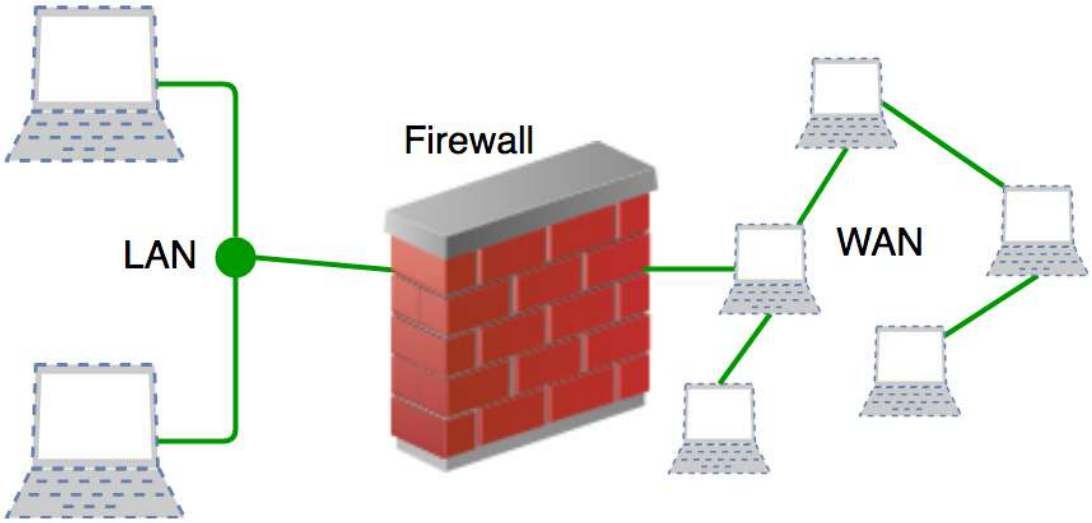
This was last updated in January 2022

Continue Reading About elliptical curve cryptography (ECC)

- EncroChat hearings delayed as lawyers seek disclosure on police hacking
- Crypto-agility: Strategies and best practices to get there
- Learn the basics of cryptography in IoT
- Mobile app security beset practices for 4 vulnerability types
- An introductory guide to mobile app security testing

Introduction of Firewall in Computer Network

A firewall is a network security device, either hardware or software-based, which monitors all incoming and outgoing traffic and based on a defined set of security rules it accepts, rejects or drops that specific traffic. **Accept :** allow the traffic **Reject :** block the traffic but reply with an “unreachable error” **Drop :** block the traffic with no reply A firewall establishes a barrier between secured internal networks and outside untrusted network, such as the Internet.



History and Need for Firewall

Before Firewalls, network security was performed by Access Control Lists (ACLs) residing on routers. ACLs are rules that determine whether network access should be granted or denied to specific IP address. But ACLs cannot determine the nature of the packet it is blocking. Also, ACL alone does not have the capacity to keep threats out of the network. Hence, the Firewall was introduced. Connectivity to the Internet is no longer optional for organizations. However, accessing the Internet provides benefits to the organization; it also enables the outside world to interact with the internal network of the

organization. This creates a threat to the organization. In order to secure the internal network from unauthorized traffic, we need a Firewall.

How does Firewall work?

Spotlight

Firewall match the network traffic against the rule set defined in its table. Once the rule is matched, associate action is applied to the network traffic. For example, Rules are defined as any employee from HR department cannot access the data from code server and at the same time another rule is defined like system administrator can access the data from both HR and technical department. Rules can be defined on the firewall based on the necessity and security policies of the organization. From the perspective of a server, network traffic can be either outgoing or incoming. Firewall maintains a distinct set of rules for both the cases. Mostly the outgoing traffic, originated from the server itself, allowed to pass. Still, setting a rule on outgoing traffic is always better in order to achieve more security and prevent unwanted communication. Incoming traffic is treated differently. Most traffic which reaches on the firewall is one of these three major Transport Layer protocols- TCP, UDP or ICMP. All these types have a source address and destination address. Also, TCP and UDP have port numbers. ICMP uses *type code* instead of port number which identifies purpose of that packet. **Default policy:** It is very difficult to explicitly cover every possible rule on the firewall. For this reason, the firewall must always have a default policy. Default policy only consists of action (accept, reject or drop). Suppose no rule is defined about SSH connection to the server on the firewall. So, it will follow the default policy. If default policy on the firewall is

set to *accept*, then any computer outside of your office can establish an SSH connection to the server. Therefore, setting default policy as *drop* (or reject) is always a good practice.

Generation of Firewall

Firewalls can be categorized based on their generation.

1. **First Generation- Packet Filtering Firewall:** Packet filtering firewall is used to control network access by monitoring outgoing and incoming packets and allowing them to pass or stop based on source and destination IP address, protocols, and ports. It analyses traffic at the transport protocol layer (but mainly uses first 3 layers). Packet firewalls treat each packet in isolation. They have no ability to tell whether a packet is part of an existing stream of traffic. Only It can allow or deny the packets based on unique packet headers. Packet filtering firewall maintains a filtering table that decides whether the packet will be forwarded or discarded. From the given filtering table, the packets will be filtered according to the following rules:

	Source IP	Dest. IP	Source Port	Dest. Port	Action
1	192.168.21.0	--	--	--	deny
2	--	--	--	23	deny
3	--	192.168.21.3	--	--	deny
4	--	192.168.21.0	--	>1023	Allow

Sample Packet Filter Firewall Rule

1. Incoming packets from network 192.168.21.0 are blocked.
2. Incoming packets destined for the internal TELNET server (port 23) are blocked.
3. Incoming packets destined for host 192.168.21.3 are blocked.
4. All well-known services to the network 192.168.21.0 are allowed.
5. **Second Generation- Stateful Inspection Firewall:** Stateful firewalls (performs Stateful Packet Inspection) are able to determine the connection

state of packet, unlike Packet filtering firewall, which makes it more efficient. It keeps track of the state of networks connection travelling across it, such as TCP streams. So the filtering decisions would not only be based on defined rules, but also on packet's history in the state table.

6. **Third Generation- Application Layer Firewall :** Application layer firewall can inspect and filter the packets on any OSI layer, up to the application layer. It has the ability to block specific content, also recognize when certain application and protocols (like HTTP, FTP) are being misused. In other words, Application layer firewalls are hosts that run proxy servers. A proxy firewall prevents the direct connection between either side of the firewall, each packet has to pass through the proxy. It can allow or block the traffic based on predefined rules. *Note: Application layer firewalls can also be used as Network Address Translator(NAT).*
7. **Next Generation Firewalls (NGFW):** Next Generation Firewalls are being deployed these days to stop modern security breaches like advance malware attacks and application-layer attacks. NGFW consists of Deep Packet Inspection, Application Inspection, SSL/SSH inspection and many functionalities to protect the network from these modern threats.

What is Magic Firewall?

"Magic Firewall" is a term used to describe a security feature provided by the web hosting and security company Cloudflare. It is a cloud-based firewall that provides protection against a wide range of security threats, including DDoS attacks, SQL injections, cross-site scripting (XSS), and other types of attacks that target web applications.

The Magic Firewall works by analyzing traffic to a website and using a set of predefined rules to identify and block malicious traffic. The rules are based on threat intelligence from a variety of sources, including the company's own threat intelligence network, and can be customized by website owners to meet their specific security needs.

The Magic Firewall is considered "magic" because it is designed to work seamlessly and invisibly to website visitors, without any noticeable impact on

website performance. It is also easy to set up and manage, and can be accessed through Cloudflare's web-based control panel.

Overall, the Magic Firewall is a powerful security tool that provides website owners with an additional layer of protection against a variety of security threats.

Types of Firewall

Firewalls are generally of two types: *Host-based* and *Network-based.*

1. **Host- based Firewalls :** Host-based firewall is installed on each network node which controls each incoming and outgoing packet. It is a software application or suite of applications, comes as a part of the operating system. Host-based firewalls are needed because network firewalls cannot provide protection inside a trusted network. Host firewall protects each host from attacks and unauthorized access.
2. **Network-based Firewalls :** Network firewall function on network level. In other words, these firewalls filter all incoming and outgoing traffic across the network. It protects the internal network by filtering the traffic using rules defined on the firewall. A Network firewall might have two or more network interface cards (NICs). A network-based firewall is usually a dedicated system with proprietary software installed.

Advantages of using Firewall

1. **Protection from unauthorized access:** Firewalls can be set up to restrict incoming traffic from particular IP addresses or networks, preventing hackers or other malicious actors from easily accessing a network or system. Protection from unwanted access.
2. **Prevention of malware and other threats:** Malware and other threat prevention: Firewalls can be set up to block traffic linked to known malware or other security concerns, assisting in the defense against these kinds of attacks.
3. **Control of network access:** By limiting access to specified individuals or groups for particular servers or applications, firewalls can be used to restrict access to particular network resources or services.

4. **Monitoring of network activity:** Firewalls can be set up to record and keep track of all network activity. This information is essential for identifying and looking into security problems and other kinds of shady behavior.
5. **Regulation compliance:** Many industries are bound by rules that demand the usage of firewalls or other security measures. Organizations can comply with these rules and prevent any fines or penalties by using a firewall.
6. **Network segmentation:** By using firewalls to split up a bigger network into smaller subnets, the attack surface is reduced and the security level is raised.

Disadvantages of using Firewall

1. **Complexity:** Setting up and keeping up a firewall can be time-consuming and difficult, especially for bigger networks or companies with a wide variety of users and devices.
2. **Limited Visibility:** Firewalls may not be able to identify or stop security risks that operate at other levels, such as the application or endpoint level, because they can only observe and manage traffic at the network level.
3. **False sense of security:** Some businesses may place an excessive amount of reliance on their firewall and disregard other crucial security measures like endpoint security or intrusion detection systems.
4. **Limited adaptability:** Because firewalls are frequently rule-based, they might not be able to respond to fresh security threats.
5. **Performance impact:** Network performance can be significantly impacted by firewalls, particularly if they are set up to analyze or manage a lot of traffic.
6. **Limited scalability:** Because firewalls are only able to secure one network, businesses that have several networks must deploy many firewalls, which can be expensive.
7. **Limited VPN support:** Some firewalls might not allow complex VPN features like split tunneling, which could restrict the experience of a remote worker.
8. **Cost:** Purchasing many devices or add-on features for a firewall system can be expensive, especially for businesses.

Real-Time Applications of Firewall

1. **Corporate networks:** Many businesses employ firewalls to guard against unwanted access and other security risks on their corporate networks. These firewalls can be set up to only permit authorized users to access particular resources or services and to prevent traffic from particular IP addresses or networks.
2. **Government organizations:** Government organizations frequently employ firewalls to safeguard sensitive data and to adhere to rules like HIPAA or PCI-DSS. They might make use of cutting-edge firewalls like Next-generation firewalls (NGFW), which can detect and stop intrusions as well as manage access to particular data and apps.
3. **Service providers:** Firewalls are used by service providers to safeguard their networks and the data of their clients, including ISPs, cloud service providers, and hosting firms. They might make use of firewalls that accommodate enormous volumes of traffic and support advanced features such as VPN and load balancing.
4. **Small enterprises:** Small firms may use firewalls to separate their internal networks, restrict access to specific resources or applications, and defend their networks from external threats.
5. **Networks at home:** To guard against unwanted access and other security risks, many home users employ firewalls. A firewall that many routers have built in can be set up to block incoming traffic and restrict access to the network.
6. **Industrial Control Systems (ICS):** Firewalls are used to safeguard industrial control systems against illegal access and cyberattacks in many vital infrastructures, including power plants, water treatment facilities, and transportation systems.

What is HMAC(Hash based Message Authentication Code)?

HMAC (Hash-based Message Authentication Code) is a type of a message authentication code (MAC) that is acquired by executing a cryptographic hash function on the data (that is) to be authenticated and a secret shared key. Like any of the MAC, it is used for both data integrity and authentication. Checking data integrity is necessary for the parties involved in communication. HTTPS, SFTP, FTPS, and other transfer protocols use HMAC. The cryptographic hash function may be MD-5, SHA-1, or SHA-256. Digital signatures are nearly similar to HMACs i.e they both employ a hash function and a shared key. The difference lies in the keys i.e HMACs use symmetric key(same copy) while Signatures use asymmetric (two different keys).

History

Processes and decisions pertinent to business are greatly dependent on integrity. If attackers tamper this data, it may affect the processes and business decisions. So while working online over the internet, care must be taken to ensure integrity or least know if the data is changed. That is when HMAC comes into use.

Applications

- Verification of e-mail address during activation or creation of an account.
- Authentication of form data that is sent to the client browser and then submitted back.
- HMACs can be used for Internet of things (IoT) due to less cost.
- Whenever there is a need to reset the password, a link that can be used once is sent without adding a server state.
- It can take a message of any length and convert it into a fixed-length message digest. That is even if you got a long message, the message digest will be small and thus permits maximizing bandwidth.

Working of HMAC

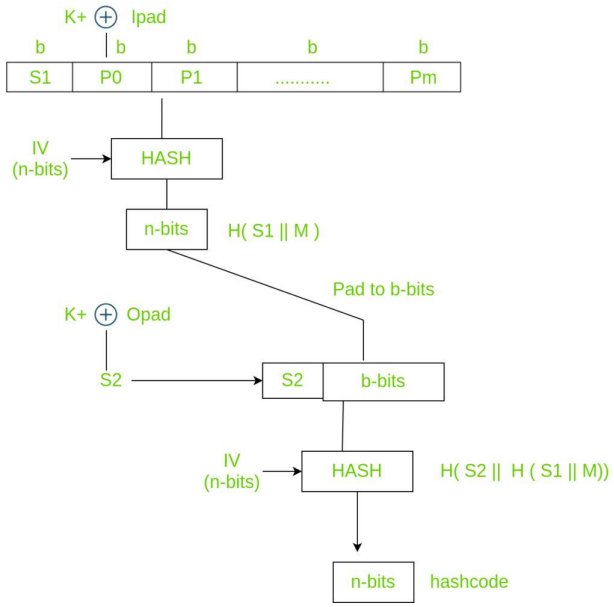
HMACs provides client and server with a shared private key that is known only to them. The client makes a unique hash (HMAC) for every

request. When the client requests the server, it hashes the requested data with a private key and sends it as a part of the request. Both the message and key are hashed in separate steps making it secure. When the server receives the request, it makes its own HMAC. Both the HMACS are compared and if both are equal, the client is considered legitimate.

The formula **for HMAC:**

```
HMAC = hashFunc(secret key + message)
```

There are three types of authentication functions. They are message encryption, message authentication code, and hash functions. The major difference between MAC and hash (HMAC here) is the dependence of a key. In HMAC we have to apply the hash function along with a key on the plain text. The hash function will be applied to the plain text message. But before applying, we have to compute S bits and then append it to plain text and after that apply the hash function. For generating those S bits we make use of a key that is shared between the sender and receiver.



Using key K (0 < K < b), K+ is generated by padding O's on left side of key K until length becomes b bits. The reason why it's not padded on right is change(increase) in the length of key. b bits because it is the block size of plain text. There are two predefined padding bits called ipad and opad. All this is done before applying hash function to the plain text message.

```
ipad - 00110110
opad - 01011100
```

Now we have to calculate S bits

K+ is EXORed with ipad and the result is S1 bits which is equivalent to b bits since both K+ and ipad are b bits. We have to append S1 with plain text messages. Let P be the plain text message.

S1, p0, p1 upto Pm each is b bits. m is the number of plain text blocks. P0 is plain text block and b is plain text block size. After appending S1 to Plain text we have to apply HASH algorithm (any variant).

Simultaneously we have to apply initialization vector (IV) which is a buffer of size n-bits. The result produced is therefore n-bit hashcode i.e H(S1 || M).

Similarly, n-bits are padded to b-bits And K+ is EXORed with opad producing output S2 bits. S2 is appended to the b-bits and once again hash function is applied with IV to the block. This further results into n-bit hashcode which is H(S2 || H(S1 || M)).

Summary:

1. Select K.
   If K < b, pad 0's on left until k=b. K is between 0 and b (0 < K < b)
2. EXOR K+ with ipad equivalent to b bits producing S1 bits.
3. Append S1 with plain text M
4. Apply SHA-512 on (S1 || M)
5. Pad n-bits until length is equal to b-bits

6. EXOR K+ with opad equivalent to b bits producing S2 bits.
7. Append S2 with output of step 5.
8. Apply SHA-512 on step 7 to output n-bit hashcode.

Advantages

- HMACs are ideal for high-performance systems like routers due to the use of hash functions which are calculated and verified quickly unlike the public key systems.
- Digital signatures are larger than HMACs, yet the HMACs provide comparably higher security.
- HMACs are used in administrations where public key systems are prohibited.

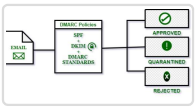
Disadvantages

- HMACs uses shared key which may lead to non-repudiation. If either sender or receiver's key is compromised then it will be easy for attackers to create unauthorized messages.

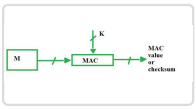
Last Updated : 31 Aug, 2021

15

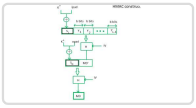
Similar Reads



Domain based Message Authentication, Reporting and Conformance…

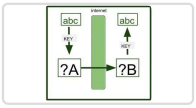


How message authentication code works?

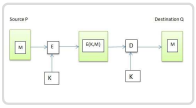


HMAC Algorithm in Computer Network

Difference between single-factor authentication and mul…



Message Authentication Codes



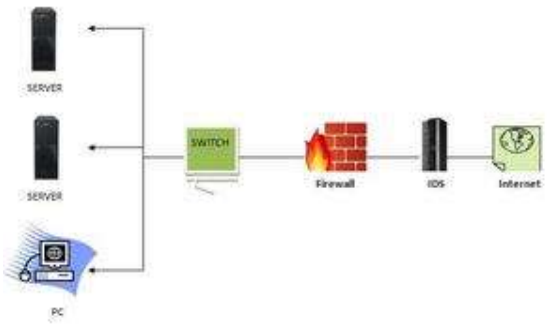
Message Authentication Requirements

Session vs Token Based Authentication

Bug in SHA-512 Hash Generation Java code

Intrusion Detection System (IDS)

A system called an intrusion detection system (IDS) observes network traffic for malicious transactions and sends immediate alerts when it is observed. It is software that checks a network or system for malicious activities or policy violations. Each illegal activity or violation is often recorded either centrally using a SIEM system or notified to an administration. IDS monitors a network or system for malicious activity and protects a computer network from unauthorized access from users, including perhaps insiders. The intrusion detector learning task is to build a predictive model (i.e. a classifier) capable of distinguishing between 'bad connections' (intrusion/attacks) and 'good (normal) connections'.



How does an IDS work?

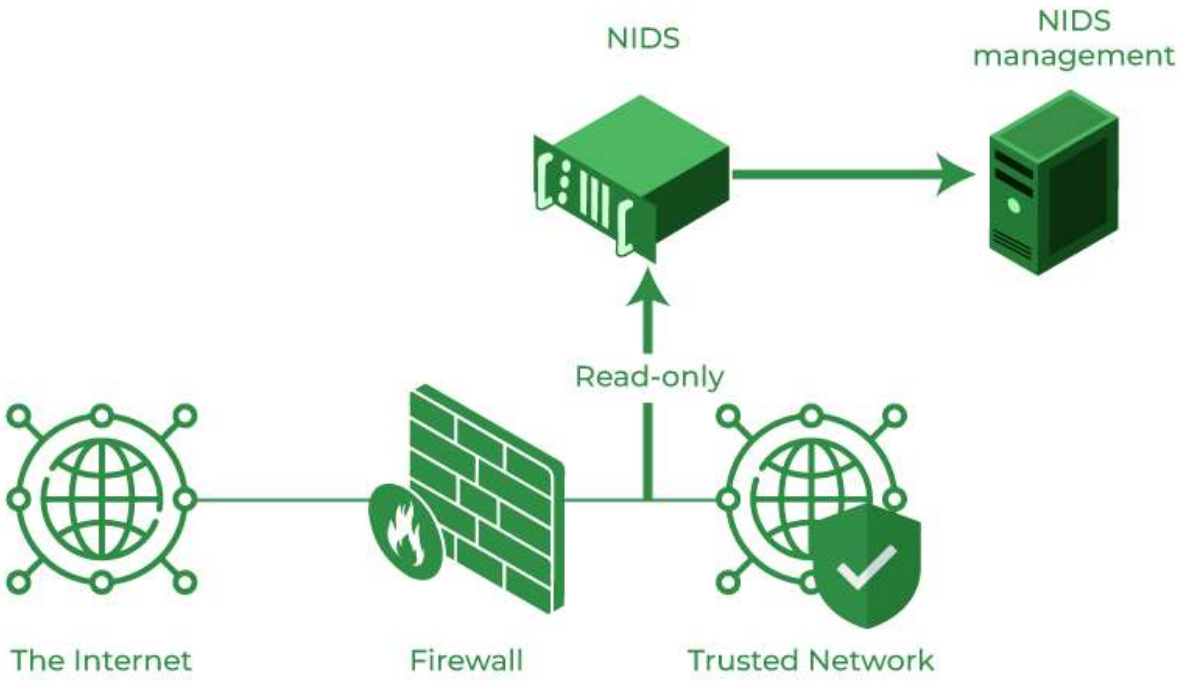
- An IDS (Intrusion Detection System) monitors the traffic on a computer network to detect any suspicious activity.
- It analyzes the data flowing through the network to look for patterns and signs of abnormal behavior.
- The IDS compares the network activity to a set of predefined rules a patterns to identify any activity that might indicate an attack or intrusion.

- If the IDS detects something that matches one of these rules or patterns, it sends an alert to the system administrator.
- The system administrator can then investigate the alert and take action to prevent any damage or further intrusion.

Classification of Intrusion Detection System

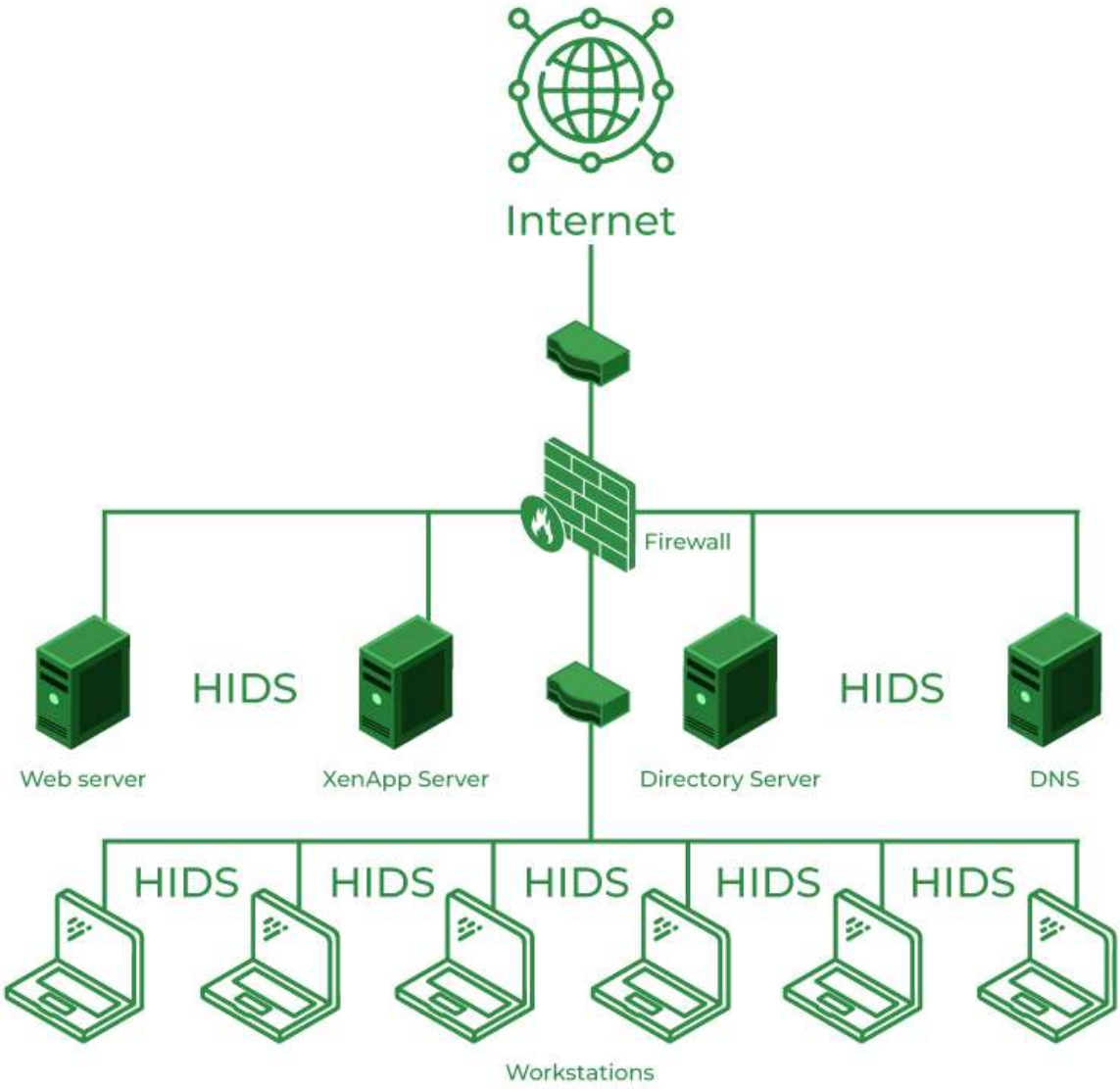
IDS are classified into 5 types:

- **Network Intrusion Detection System (NIDS):** Network intrusion detection systems (NIDS) are set up at a planned point within the network to examine traffic from all devices on the network. It performs an observation of passing traffic on the entire subnet and matches the traffic that is passed on the subnets to the collection of known attacks. Once an attack is identified or abnormal behavior is observed, the alert can be sent to the administrator. An example of a NIDS is installing it on the subnet where firewalls are located in order to see if someone is trying to crack the firewall.



- **Host Intrusion Detection System (HIDS):** Host intrusion detection systems (HIDS) run on independent hosts or devices on the network. A HIDS monitors the incoming and outgoing packets from the device only and will

alert the administrator if suspicious or malicious activity is detected. It takes a snapshot of existing system files and compares it with the previous snapshot. If the analytical system files were edited or deleted, an alert is sent to the administrator to investigate. An example of HIDS usage can be seen on mission-critical machines, which are not expected to change their layout.



- **Protocol-based Intrusion Detection System (PIDS):** Protocol-based intrusion detection system (PIDS) comprises a system or agent that would consistently reside at the front end of a server, controlling and interpreting the protocol between a user/device and the server. It is trying to secure the web server by regularly monitoring the HTTPS protocol stream and

accepting the related HTTP protocol. As HTTPS is unencrypted and before instantly entering its web presentation layer then this system would need to reside in this interface, between to use the HTTPS.

- **Application Protocol-based Intrusion Detection System (APIDS):** An application Protocol-based Intrusion Detection System (APIDS) is a system or agent that generally resides within a group of servers. It identifies the intrusions by monitoring and interpreting the communication on application-specific protocols. For example, this would monitor the SQL protocol explicitly to the middleware as it transacts with the database in the web server.
- **Hybrid Intrusion Detection System:** Hybrid intrusion detection system is made by the combination of two or more approaches to the intrusion detection system. In the hybrid intrusion detection system, the host agent or system data is combined with network information to develop a complete view of the network system. The hybrid intrusion detection system is more effective in comparison to the other intrusion detection system. Prelude is an example of Hybrid IDS.

Benefits of IDS

- **Detects malicious activity:** IDS can detect any suspicious activities and alert the system administrator before any significant damage is done.
- **Improves network performance:** IDS can identify any performance issues on the network, which can be addressed to improve network performance.
- **Compliance requirements:** IDS can help in meeting compliance requirements by monitoring network activity and generating reports.
- **Provides insights:** IDS generates valuable insights into network traffic, which can be used to identify any weaknesses and improve network security.

Detection Method of IDS

1. **Signature-based Method:** Signature-based IDS detects the attacks on the basis of the specific patterns such as the number of bytes or a number of 1s or the number of 0s in the network traffic. It also detects on the basis of the already known malicious instruction sequence that is used by the malware.

IP security (IPSec)

Pre-Requisite: Types of Internet Protocol

IP Sec (Internet Protocol Security) is an Internet Engineering Task Force (IETF) standard suite of protocols between two communication points across the IP network that provide data authentication, integrity, and confidentiality. It also defines the encrypted, decrypted, and authenticated packets. The protocols needed for secure key exchange and key management are defined in it.

Uses of IP Security

IPsec can be used to do the following things:

- To encrypt application layer data.
- To provide security for routers sending routing data across the public internet.
- To provide authentication without encryption, like to authenticate that the data originates from a known sender.
- To protect network data by setting up circuits using IPsec tunneling in which all data being sent between the two endpoints is encrypted, as with a Virtual Private Network(VPN) connection.

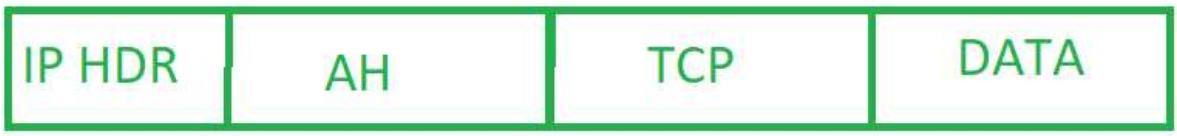
Components of IP Security

It has the following components:

1. Encapsulating Security Payload (ESP)
2. Authentication Header (AH)
3. Internet Key Exchange (IKE)

1. Encapsulating Security Payload (ESP): It provides data integrity, encryption, authentication, and anti-replay. It also provides authentication for payload.

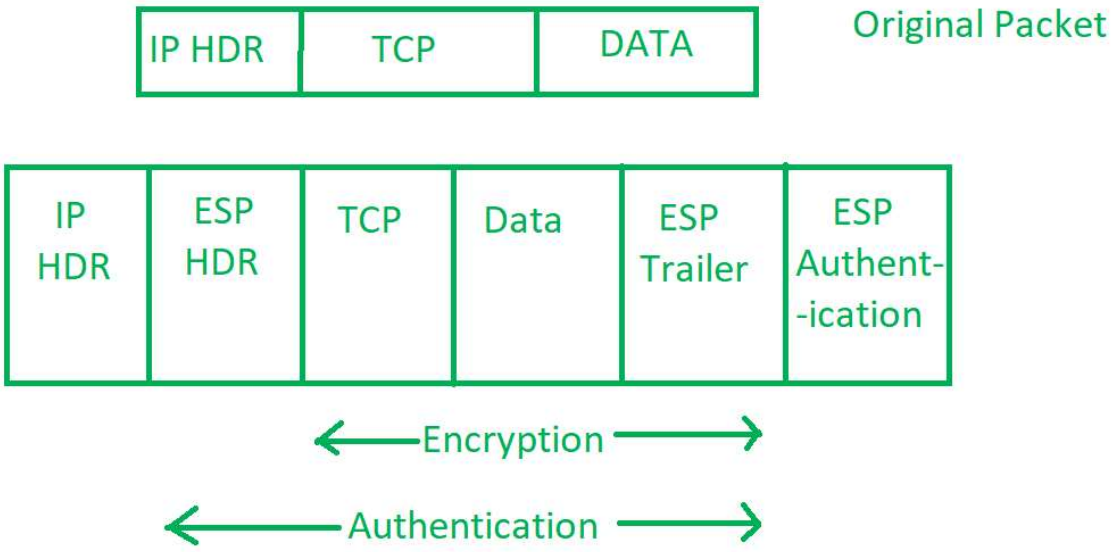
2. Authentication Header (AH): It also provides data integrity, authentication, and anti-replay and it does not provide encryption. The anti-replay protection protects against the unauthorized transmission of packets. It does not protect data confidentiality.



IP Header

3. Internet Key Exchange (IKE): It is a network security protocol designed to dynamically exchange encryption keys and find a way over Security Association (SA) between 2 devices. The Security Association (SA) establishes shared security attributes between 2 network entities to support secure communication. The Key Management Protocol (ISAKMP) and Internet Security Association provides a framework for authentication and key exchange. ISAKMP tells how the setup of the Security Associations (SAs) and how direct connections between two hosts are using IPsec. Internet Key Exchange (IKE) provides message content

protection and also an open frame for implementing standard algorithms such as SHA and MD5. The algorithm's IP sec users produce a unique identifier for each packet. This identifier then allows a device to determine whether a packet has been correct or not. Packets that are not authorized are discarded and not given to the receiver.

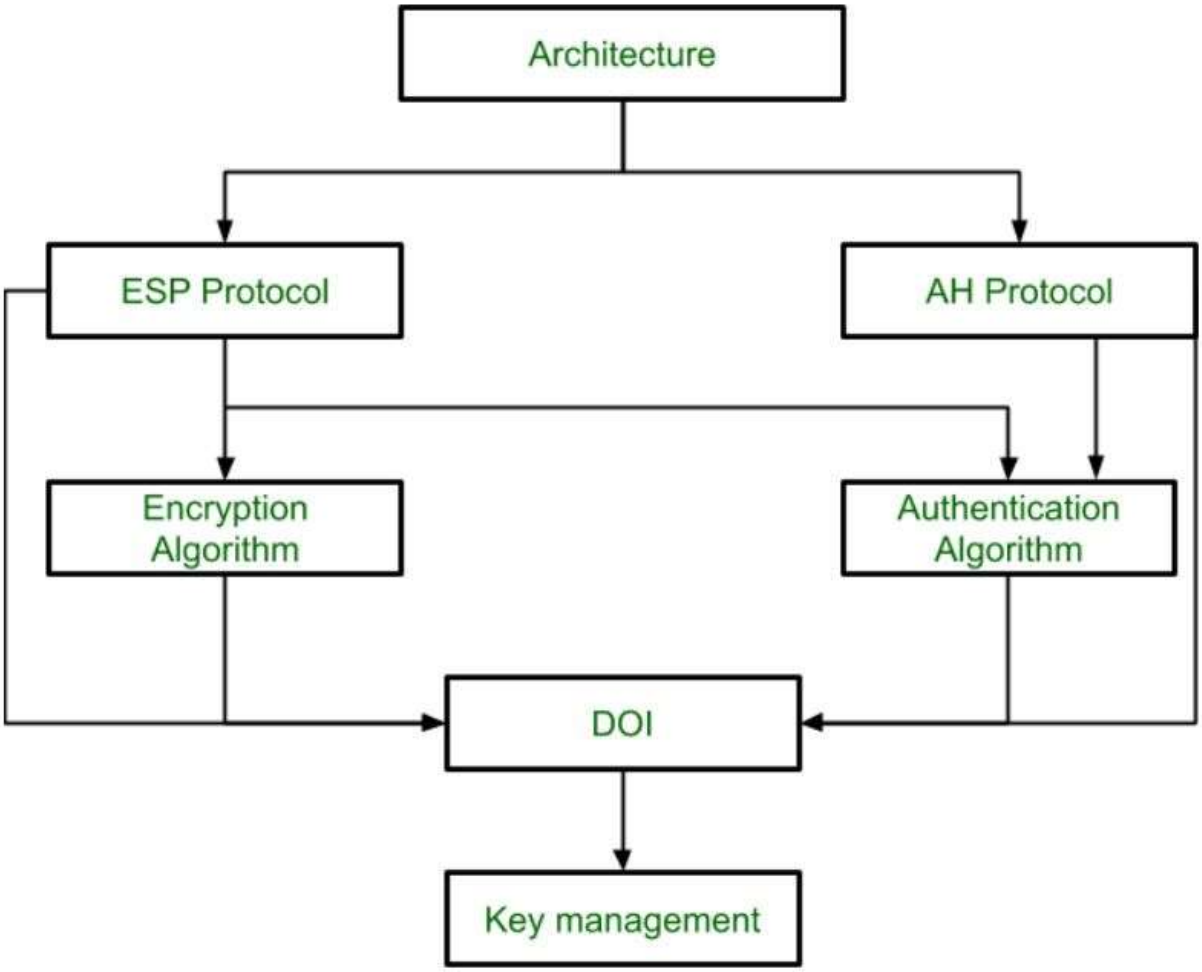


Packets in Internet Protocol

IP Security Architecture

IPSec (IP Security) architecture uses two protocols to secure the traffic or data flow. These protocols are ESP (Encapsulation Security Payload) and AH (Authentication Header). IPSec Architecture includes protocols, algorithms, DOI, and Key Management. All these components are very important in order to provide the three main services:

- Confidentiality
- Authenticity
- Integrity



IP Security Architecture

Working on IP Security

- The host checks if the packet should be transmitted using IPsec or not. This packet traffic triggers the security policy for itself. This is done when the system sending the packet applies appropriate encryption. The incoming packets are also checked by the host that they are encrypted properly or not.
- Then IKE Phase 1 starts in which the 2 hosts(using IPsec) authenticate themselves to each other to start a secure channel. It has 2 modes. The Main mode provides greater security and the Aggressive mode which enables the host to establish an IPsec circuit more quickly.
- The channel created in the last step is then used to securely negotiate the way the IP circuit will encrypt data across the IP circuit.
- Now, the IKE Phase 2 is conducted over the secure channel in which the two hosts negotiate the type of cryptographic algorithms to use on

the session and agree on secret keying material to be used with those algorithms.
- Then the data is exchanged across the newly created IPsec encrypted tunnel. These packets are encrypted and decrypted by the hosts using IPsec SAs.
- When the communication between the hosts is completed or the session times out then the IPsec tunnel is terminated by discarding the keys by both hosts.

Features of IPSec

1. **Authentication:** IPSec provides authentication of IP packets using digital signatures or shared secrets. This helps ensure that the packets are not tampered with or forged.
2. **Confidentiality:** IPSec provides confidentiality by encrypting IP packets, preventing eavesdropping on the network traffic.
3. **Integrity:** IPSec provides integrity by ensuring that IP packets have not been modified or corrupted during transmission.
4. **Key management:** IPSec provides key management services, including key exchange and key revocation, to ensure that cryptographic keys are securely managed.
5. **Tunneling:** IPSec supports tunneling, allowing IP packets to be encapsulated within another protocol, such as GRE (Generic Routing Encapsulation) or L2TP (Layer 2 Tunneling Protocol).
6. **Flexibility:** IPSec can be configured to provide security for a wide range of network topologies, including point-to-point, site-to-site, and remote access connections.
7. **Interoperability:** IPSec is an open standard protocol, which means that it is supported by a wide range of vendors and can be used in heterogeneous environments.

Advantages of IPSec

1. **Strong security:** IPSec provides strong cryptographic security services that help protect sensitive data and ensure network privacy and integrity.
2. **Wide compatibility:** IPSec is an open standard protocol that is widely supported by vendors and can be used in heterogeneous environments.
3. **Flexibility:** IPSec can be configured to provide security for a wide range of network topologies, including point-to-point, site-to-site, and remote access connections.
4. **Scalability:** IPSec can be used to secure large-scale networks and can be scaled up or down as needed.
5. **Improved network performance:** IPSec can help improve network performance by reducing network congestion and improving network efficiency.

Disadvantages of IPSec

1. **Configuration complexity:** IPSec can be complex to configure and requires specialized knowledge and skills.
2. **Compatibility issues:** IPSec can have compatibility issues with some network devices and applications, which can lead to interoperability problems.
3. **Performance impact:** IPSec can impact network performance due to the overhead of encryption and decryption of IP packets.
4. **Key management:** IPSec requires effective key management to ensure the security of the cryptographic keys used for encryption and authentication.
5. **Limited protection:** IPSec only provides protection for IP traffic, and other protocols such as ICMP, DNS, and routing protocols may still be vulnerable to attacks.

Key Management in Cryptography

In cryptography, it is a very tedious task to distribute the public and private keys between sender and receiver. If the key is known to the third party (forger/eavesdropper) then the whole security mechanism becomes worthless. So, there comes the need to secure the exchange of keys.

There are two aspects for Key Management:

1. Distribution of public keys.
2. Use of public-key encryption to distribute secrets.

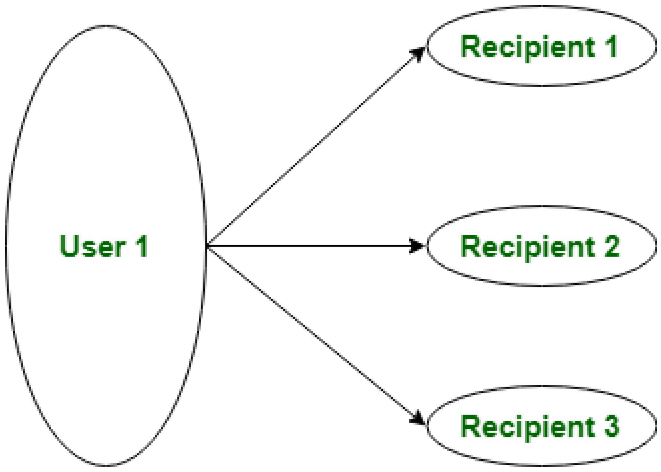
Distribution of Public Key:

The public key can be distributed in four ways:

1. Public announcement
2. Publicly available directory
3. Public-key authority
4. Public-key certificates.

These are explained as following below:

1. Public Announcement: Here the public key is broadcasted to everyone. The major weakness of this method is a forgery. Anyone can create a key claiming to be someone else and broadcast it. Until forgery is discovered can masquerade as claimed user.



Public Key Announcement

2. Publicly Available Directory: In this type, the public key is stored in a public directory. Directories are trusted here, with properties like Participant Registration, access and allow to modify values at any time, contains entries like {name, public-key}. Directories can be accessed electronically still vulnerable to forgery or tampering.

3. Public Key Authority: It is similar to the directory but, improves security by tightening control over the distribution of keys from the directory. It requires users to know the public key for the directory. Whenever the keys are needed, real-time access to the directory is made by the user to obtain any desired public key securely.

4. Public Certification: This time authority provides a certificate (which binds an identity to the public key) to allow key exchange without real-time access to the public authority each time. The certificate is

accompanied by some other info such as period of validity, rights of use, etc. All of this content is signed by the private key of the certificate authority and it can be verified by anyone possessing the authority's public key.

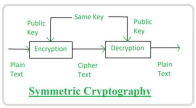
First sender and receiver both request CA for a certificate which contains a public key and other information and then they can exchange these certificates and can start communication.

Last Updated : 13 Jan, 2022

46

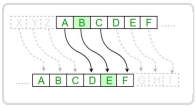
Similar Reads


Custom Building Cryptography Algorithms (Hybrid Cryptography)



Classical Cryptography and Quantum Cryptography


What Should Be the Length of the Symmetric Key in Cryptography?



Caesar Cipher in Cryptography


RSA Algorithm in Cryptography


Cryptography Introduction

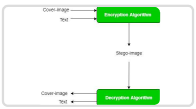
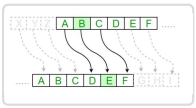


Image Steganography in Cryptography


DNA Cryptography


Birthday attack in Cryptography



History of Cryptography

Previous

Public Key Encryption

Next

Implementation of Diffie-Hellman Algorithm

Article Contributed By :

Krishna_Yadav

Key Management in Cryptography

In cryptography, it is a very tedious task to distribute the public and private keys between sender and receiver. If the key is known to the third party (forger/eavesdropper) then the whole security mechanism becomes worthless. So, there comes the need to secure the exchange of keys.

There are two aspects for Key Management:

1. Distribution of public keys.
2. Use of public-key encryption to distribute secrets.

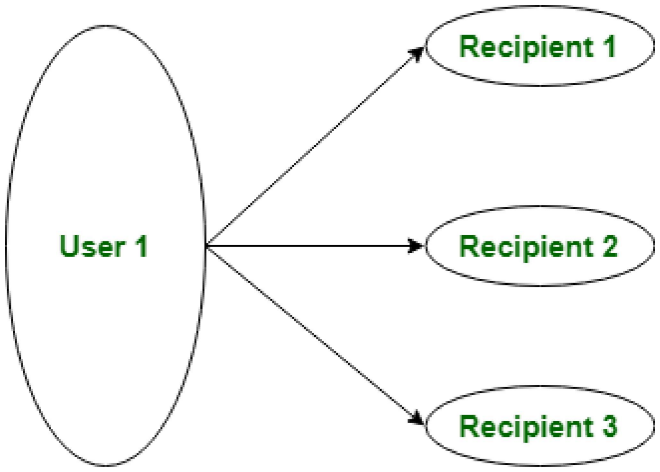
Distribution of Public Key:

The public key can be distributed in four ways:

1. Public announcement
2. Publicly available directory
3. Public-key authority
4. Public-key certificates.

These are explained as following below:

1. Public Announcement: Here the public key is broadcasted to everyone. The major weakness of this method is a forgery. Anyone can create a key claiming to be someone else and broadcast it. Until forgery is discovered can masquerade as claimed user.



Public Key Announcement

2. Publicly Available Directory: In this type, the public key is stored in a public directory. Directories are trusted here, with properties like Participant Registration, access and allow to modify values at any time, contains entries like {name, public-key}. Directories can be accessed electronically still vulnerable to forgery or tampering.

3. Public Key Authority: It is similar to the directory but, improves security by tightening control over the distribution of keys from the directory. It requires users to know the public key for the directory. Whenever the keys are needed, real-time access to the directory is made by the user to obtain any desired public key securely.

4. Public Certification: This time authority provides a certificate (which binds an identity to the public key) to allow key exchange without real-time access to the public authority each time. The certificate is

accompanied by some other info such as period of validity, rights of use, etc. All of this content is signed by the private key of the certificate authority and it can be verified by anyone possessing the authority's public key.

First sender and receiver both request CA for a certificate which contains a public key and other information and then they can exchange these certificates and can start communication.

Last Updated : 13 Jan, 2022

46

Similar Reads

- Custom Building Cryptography Algorithms (Hybrid Cryptography)
- Classical Cryptography and Quantum Cryptography
- What Should Be the Length of the Symmetric Key in Cryptography?
- RSA Algorithm in Cryptography
- Cryptography Introduction
- Image Steganography in Cryptography
- DNA Cryptography
- Birthday attack in Cryptography
- History of Cryptography
- An Overview of Cloud Cryptography

What Is a Message Authentication Code?

Message Authentication Code (MAC) Defined

Message Authentication Code (MAC), also referred to as a tag, is used to authenticate the origin and nature of a message. MACs use authentication cryptography to verify the legitimacy of data sent through a network or transferred from one person to another.

In other words, MAC ensures that the message is coming from the correct sender, has not been changed, and that the data transferred over a network or stored in or outside a system is legitimate and does not contain harmful code. MACs can be stored on a hardware security module, a device used to manage sensitive digital keys.

How Does a Message Authentication Code Work?

The first step in the MAC process is the establishment of a secure channel between the receiver and the sender. To encrypt a message, the MAC system uses an algorithm, which uses a symmetric key and the plain text message being sent. The MAC algorithm then generates authentication tags of a fixed length by processing the message. The resulting computation is the message's MAC.

This MAC is then appended to the message and transmitted to the receiver. The receiver computes the MAC using the same algorithm. If the resulting MAC the receiver arrives at equals the one sent by the sender, the message is verified as authentic, legitimate, and not tampered with.

In effect, MAC uses a secure key only known to the sender and the recipient. Without this information, the recipient will not be able to open, use, read, or even receive the data being sent. If the data is to be altered between the time the sender initiates the transfer and when the recipient receives it, the MAC information will also be affected.

Therefore, when the recipient attempts to verify the authenticity of the data, the key will not work, and the end result will not match that of the sender. When this kind of discrepancy is detected, the data packet can be discarded, protecting the recipient's system.

Types of Message Authentication Codes?

Although all MACs accomplish the same end objective, there are a few different types.

1. One-time MAC

A one-time MAC is a lot like one-time encryption in that a MAC algorithm for a single use is defined to secure the transmission of data. One-time MACs tend to be faster than other authentication algorithms.

2. Carter-Wegman MAC

A Carter-Wegman MAC is similar to a one-time MAC, except it also incorporates a pseudorandom function that makes it possible for a single key to be used many times over.

3. HMAC

With a Keyed-Hash Message Authentication Code (HMAC) system, a one-way hash is used to create a unique MAC value for every message sent. The input parameters can have various values assigned, and making them very different from each other may produce a higher level of security.

Approved Message Authentication Code Algorithms

The approved general-purpose MAC algorithms are HMAC, KECCAK Message Authentication Code (KMAC), and Cipher-based Method Authentication Code (CMAC). Message authentication in cryptography depends on hashes, which are used to verify the legitimacy of the transmission, ensuring the message has not been altered or otherwise corrupted since it was first transmitted by the sender.

Keyed-Hash Message Authentication Code (HMAC)

The HMAC is based on an approved hash function. It performs a function similar to that of the Rivest-Shamir-Adelman (RSA) cryptosystem, which is one of the oldest methods of sending data securely. The functions that can be used in HMAC are outlined in the following publications:

1. FIPS 180-4, Secure Hash Standard
2. FIPS 202, SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions

Guidelines regarding HMAC's security are outlined in NIST SP 800-107 Revision 1, Recommendation for Applications Using Approved Hash Algorithms.

KECCAK Message Authentication Code (KMAC)

KMACs consist of keyed cryptographic algorithms, and their parameters are specified in FIPS 202, SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. Two variants of KECCAK exist: KMAC256 and KMAC128.

The CMAC Mode for Authentication

As outlined in SP 800-38B, Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication, CMAC is built using an approved block cipher, which is an algorithm that uses a symmetric encryption key, similar to the NIST's Advanced Encryption Standard (AES), which also uses a symmetric key and was used to guard classified information by the U.S. government.

What Are the Benefits of Message Authentication Codes?

1. Protects Data Integrity

With MACs, you can make sure that unauthorized code, such as executable codes used by viruses, has not been put into your system. This is useful when trying to combat viruses and other malware.

2. Detects Changes in the Message Content

You can use an application to generate a MAC based on data that has been sent to you or provided via a storage device. After the application generates a MAC, it can be compared to the original one to detect changes to the data.

How Fortinet Can Help

With FortiGate next-generation firewall (NGFW), all your network traffic is filtered at a granular level, preventing unauthorized or malicious files from entering or exiting your system. FortiGate help you filter out data from unauthorized sources.

If a malicious actor that has been identified by FortiGuard, the intelligence system that powers FortiGate, inserted unauthorized messages, FortiGate prevents their data from getting into your system. Similarly, if the message was originally benevolent but was changed into something containing malicious code, FortiGate can detect the dangerous code and discard the data packet that carries it.

FAQs

What is a message authentication code?

Message Authentication Code (MAC), also referred to as a tag, is used to authenticate the origin and nature of a message. MAC ensures that the message is coming from the correct sender, has not been changed, and that the data transferred over a network or stored in or outside a system is legitimate and does not contain harmful code.

How does a message authentication code work?

The first step in the MAC process is the establishment of a secure channel between the receiver and the sender. To encrypt a message, the MAC system uses an algorithm, which uses a symmetric key and the plain text message being sent. The MAC algorithm then generates authentication tags of a fixed length by processing the message.

What are the types of message authentication codes?

Although all MAC's accomplish the same end objective, there are a few different types.

- One-time MAC

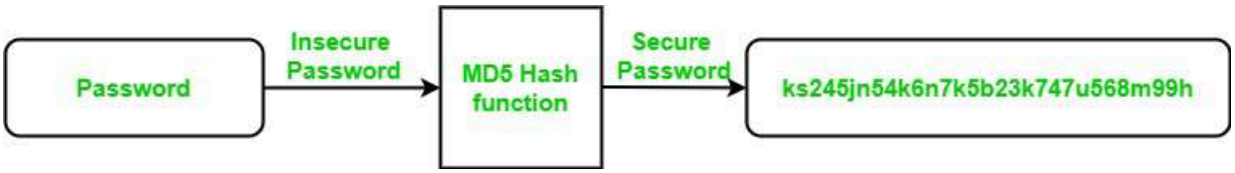
- Carter-Wegman MAC

What is the MD5 Algorithm?

MD5 is a cryptographic hash function algorithm that takes the message as input of any length and changes it into a fixed-length message of 16 bytes. MD5 algorithm stands for the **message-digest algorithm**. MD5 was developed as an improvement of MD4, with advanced security purposes. The output of MD5 (Digest size) is always **128 bits. MD5** was developed in 1991 by **Ronald Rivest.**

Use Of MD5 Algorithm:

- It is used for file authentication.
- In a web application, it is used for security purposes. e.g. Secure password of users etc.
- Using this algorithm, We can store our password in 128 bits format.



MD5 Algorithm

Working of the MD5 Algorithm:

MD5 algorithm follows the following steps

1. Append Padding Bits: In the first step, we add padding bits in the original message in such a way that the total length of the message is 64 bits less than the exact multiple of 512.

Suppose we are given a message of 1000 bits. Now we have to add padding bits to the original message. Here we will add 472 padding bits to the original message. After adding the padding bits the size of the original message/output of the first step will be 1472 i.e. 64 bits less than an exact multiple of 512 (i.e. 512*3 = 1536).

Length(original message + padding bits) = 512 * i – 64 where i = 1,2,3 . . .

2. Append Length Bits: In this step, we add the length bit in the output of the first step in such a way that the total number of the bits is the perfect multiple of 512. Simply, here we add the 64-bit as a length bit in the output of the first step.
i.e. output of first step = 512 * n – 64
length bits = 64.

After adding both we will get **512 * n** i.e. the exact multiple of 512.

3. Initialize MD buffer: Here, we use the 4 buffers i.e. J, K, L, and M. The size of each buffer is 32 bits.

- J = 0x67425301
- K = 0xEDFCBA45
- L = 0x98CBADFE
- M = 0x13DCE476

4. Process Each 512-bit Block: This is the most important step of the MD5 algorithm. Here, a total of 64 operations are performed in 4 rounds. In the 1st round, 16 operations will be performed, 2nd round 16 operations will be performed, 3rd round 16 operations will be performed, and in the 4th round, 16 operations will be performed. We apply a different function on each round i.e. for the 1st round we apply the F function, for the 2nd G function, 3rd for the H function, and 4th for the I function.
We perform OR, AND, XOR, and NOT (basically these are logic gates) for calculating functions. We use 3 buffers for each function i.e. K, L, M.

```
- F(K,L,M) = (K AND L) OR (NOT K  AND M)
- G(K,L,M) = (K AND L) OR (L AND NOT M)
- H(K,L,M) = K XOR L XOR M
- I(K,L,M) = L XOR (K OR NOT M)
```

After applying the function now we perform an operation on each block. For performing operations we need

- add modulo 2^{32}
- M[i] – 32 bit message.
- K[i] – 32-bit constant.
- <<<n – Left shift by n bits.

Now take input as initialize MD buffer i.e. J, K, L, M. Output of K will be fed in L, L will be fed into M, and M will be fed into J. After doing this now we perform some operations to find the output for J.

- In the first step, Outputs of K, L, and M are taken and then the function F is applied to them. We will add modulo 2^{32} bits for the output of this with J.
- In the second step, we add the M[i] bit message with the output of the first step.
- Then add 32 bits constant i.e. K[i] to the output of the second step.

- At last, we do left shift operation by n (can be any value of n) and addition modulo by 2^{32}.

After all steps, the result of J will be fed into K. Now same steps will be used for all functions G, H, and I. After performing all 64 operations we will get our message digest.

Output:

After all, rounds have been performed, the buffer J, K, L, and M contains the MD5 output starting with the lower bit J and ending with Higher bits M.

Code:

Python

```python
# importing the required libraries
import hashlib
# making a message
inputstring = "This is a message sent by a computer user."
# encoding the message using the library function
output = hashlib.md5(inputstring.encode())
# printing the hash function
print("Hash of the input string:")
print(output.hexdigest())
```

Output

```
Hash of the input string:
922547e866c89b8f677312df0ccec8ee
```

Application Of MD5 Algorithm:

- We use message digest to verify the integrity of files/ authenticates files.
- MD5 was used for data security and encryption.

Password Management in Cyber Security

A password is a secret word or phrase or code that you need to know in order to have access to a place or system. In technical terms, it is a series of letters or numbers that you must type into a computer or computer system in order to be able to use it. A password is a real-life implementation of challenge-response authentication (a set of protocols to protect digital assets and data).

> *A string of characters i.e letters, numbers, special characters, used to verify the identity of a user during the authentication process is known as password.*

Password Management:

Since passwords are meant to keep the files and data secret and safe so it is prevented the unauthorized access, password management refers to the practices and set of rules or principles or standards that out must follow or at least try to seek help from in order to be a good/strong password and along with its storage and management for the future requirements.

Issues Related to Managing Passwords:

The main problem with password management is that it is not safe to use the same password for multiple sites, therefore having different passwords for different sites and on top of that remembering them is quite difficult. As per the statistics, more than 65% of people reuse passwords across accounts and a majority do not change them, even after a known breach. Meanwhile, 25% reset their passwords once a month or more because they forgot them.

To escape from this situation people often tend to use password managers (A password manager is a computer program that allows users to store, generate, and manage their passwords for local applications and online services.). Password managers to a certain extent reduce the problem by having to remember only one “master password” instead of having to remember multiple passwords. The only problem with having a master password is that once it is out or known to an attacker, the rest of all the passwords become available.

The main issues related to managing passwords are as follows:

- Login spoofing
- Sniffing attack
- Brute force attack
- Shoulder surfing attack
- Data breach

Methods to Manage Password:

There are a lot of good practices that we can follow to generate a strong password and also the ways to manage them.

- **Strong and long passwords:** A minimum length of 8 to 12 characters long, also it should contain at least three different character sets (e.g., uppercase characters, lowercase characters, numbers, or symbols)
- **Password Encryption:** Using irreversible end-to-end encryption is recommended. In this way, the password remains safe even if it ends up in the hands of cybercriminals.

- **Multi-factor Authentication (MFA):** Adding some security questions and a phone number that would be used to confirm that it is indeed you who is trying to log in will enhance the security of your password.
- **Make the password pass the test:** Yes, put your password through some testing tools that you might find online in order to ensure that it falls under the strong and safe password category.
- **Avoid updating passwords frequently:** Though it is advised or even made mandatory to update or change your password as frequently as in 60 or 90 days.

Last Updated : 26 Jun, 2022

1

Similar Reads


Cyber Security and Cyber Crimes


Difference between Cyber Security and Information Security

Difference between Network Security and Cyber Security

How Security System Should Evolve to Handle Cyber Security Threats and…


Difference between Software Security and Cyber Security


Incident Management in Cyber Security


Identity and Access Management (IAM) in Cyber Security Roles


Difference Between Zombie and Logic Bomb in Cyber Security


Information Security and Cyber Laws


Cyber Security in Context to Organisations

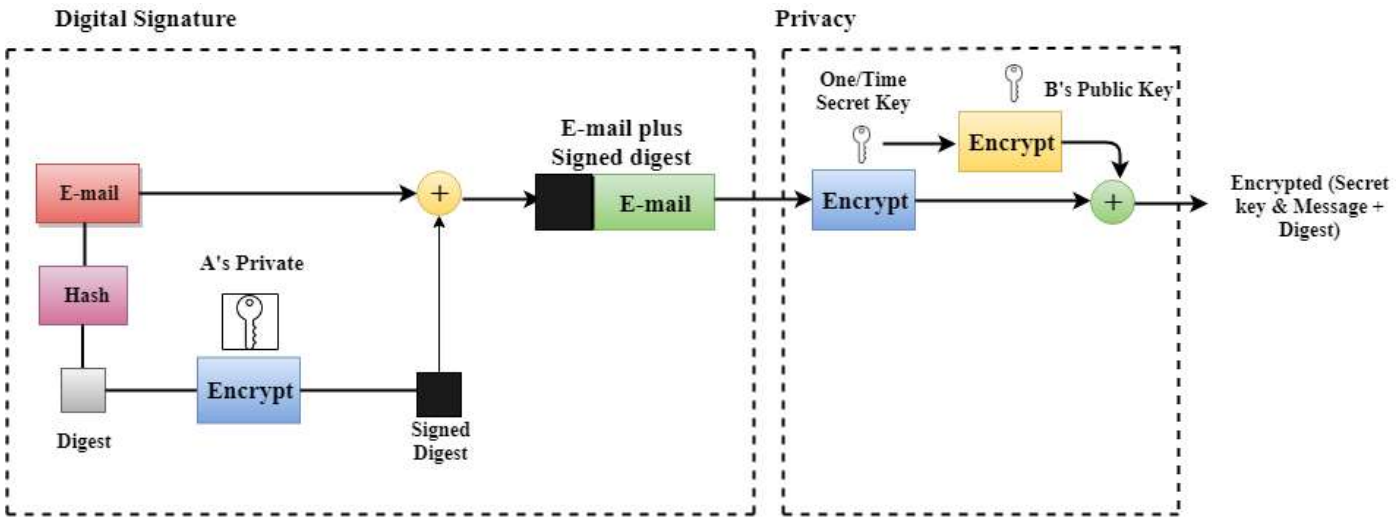
PGP

- PGP stands for Pretty Good Privacy (PGP) which is invented by Phil Zimmermann.
- PGP was designed to provide all four aspects of security, i.e., privacy, integrity, authentication, and non-repudiation in the sending of email.
- PGP uses a digital signature (a combination of hashing and public key encryption) to provide integrity, authentication, and non-repudiation. PGP uses a combination of secret key encryption and public key encryption to provide privacy. Therefore, we can say that the digital signature uses one hash function, one secret key, and two private-public key pairs.
- PGP is an open source and freely available software package for email security.
- PGP provides authentication through the use of Digital Signature.
- It provides confidentiality through the use of symmetric block encryption.
- It provides compression by using the ZIP algorithm, and EMAIL compatibility using the radix-64 encoding scheme.

Following are the steps taken by PGP to create secure e-mail at the sender site:

- The e-mail message is hashed by using a hashing function to create a digest.
- The digest is then encrypted to form a signed digest by using the sender's private key, and then signed digest is added to the original email message.
- The original message and signed digest are encrypted by using a one-time secret key created by the sender.
- The secret key is encrypted by using a receiver's public key.
- Both the encrypted secret key and the encrypted combination of message and digest are sent together.

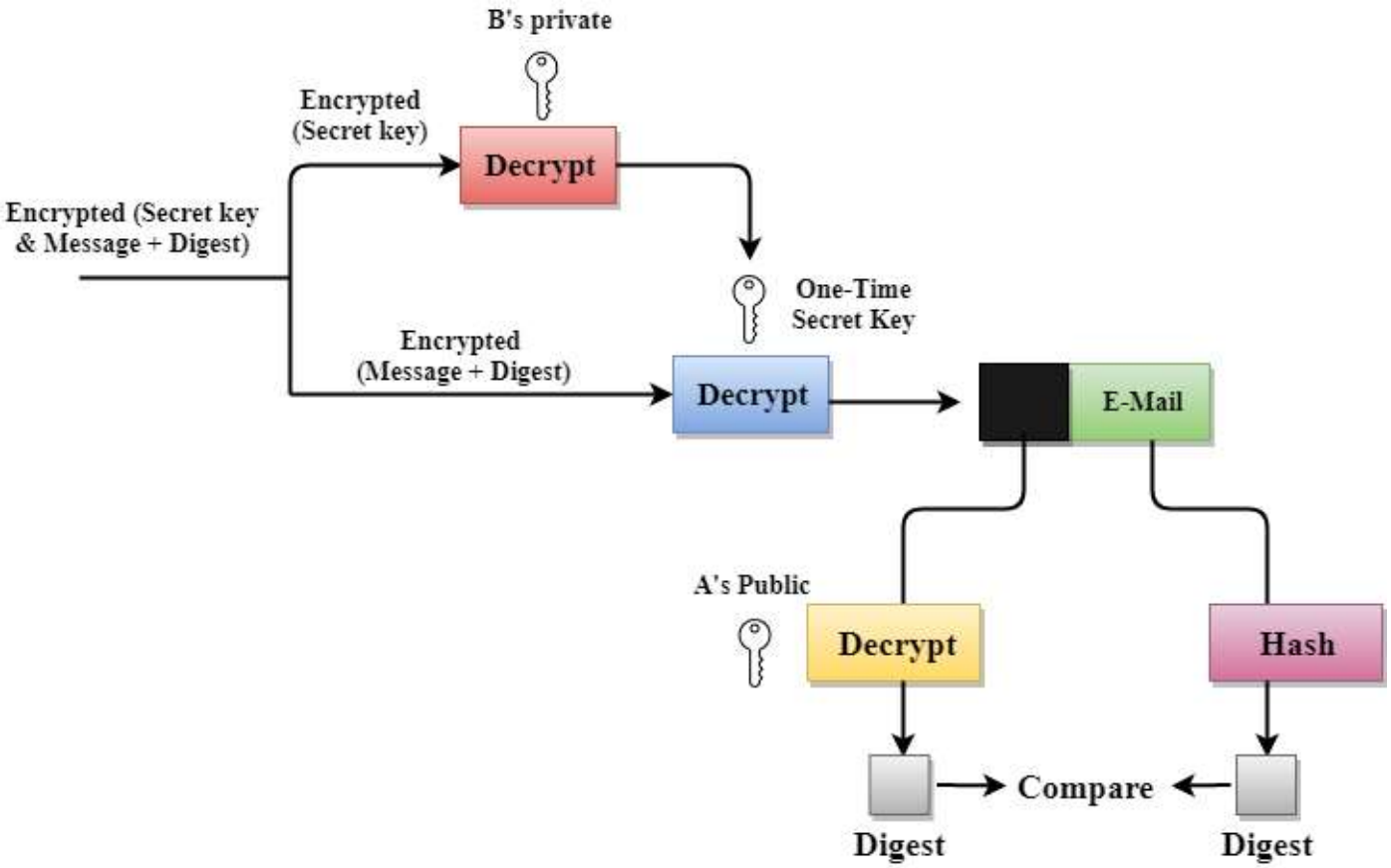
PGP at the Sender site (A)



Following are the steps taken to show how PGP uses hashing and a combination of three keys to generate the original message:

- The receiver receives the combination of encrypted secret key and message digest is received.
- The encrypted secret key is decrypted by using the receiver's private key to get the one-time secret key.
- The secret key is then used to decrypt the combination of message and digest.
- The digest is decrypted by using the sender's public key, and the original message is hashed by using a hash function to create a digest.
- Both the digests are compared if both of them are equal means that all the aspects of security are preserved.

PGP at the Receiver site (B)



Disadvantages of PGP Encryption

- **The Administration is difficult:** The different versions of PGP complicate the administration.
- **Compatibility issues:** Both the sender and the receiver must have compatible versions of PGP. For example, if you encrypt an email by using PGP with one of the encryption technique, the receiver has a different version of PGP which cannot read the data.
- **Complexity:** PGP is a complex technique. Other security schemes use symmetric encryption that uses one key or asymmetric encryption that uses two different keys. PGP uses a hybrid approach that implements symmetric encryption with two keys. PGP is more complex, and it is less familiar than the traditional symmetric or asymmetric methods.
- **No Recovery:** Computer administrators face the problems of losing their passwords. In such situations, an administrator should use a special program to retrieve passwords. For example, a technician has physical access to a PC which can be used to retrieve a password. However, PGP does not offer such a special program for recovery; encryption methods are

What is RC4 Encryption?

RC4 means Rivest Cipher 4 invented by Ron Rivest in 1987 for RSA Security. It is a Stream Ciphers. Stream Ciphers operate on a stream of data byte by byte. RC4 stream cipher is one of the most widely used stream ciphers because of its simplicity and speed of operation. It is a variable key-size stream cipher with byte-oriented operations. It uses either 64 bit or 128-bit key sizes. It is generally used in applications such as Secure Socket Layer (SSL), Transport Layer Security (TLS), and also used in IEEE 802.11 wireless LAN std.

Why Encryption Is Important?

Unauthorized data access can be prevented by encryption. If we perform encryption then third parties can not have access to data which we share or receive. The encryption is done by using a secret key, or we can say that by using a public key and private key. Both sender and receiver are having their public key and private key through which encryption of plain text and decryption of ciphertext is performed.

History of RC4 Encryption

RC4 was designed by Ron Rivest in 1987. He was working under RSA Security. Rivest Cipher 4 is an official name while it is also known as Ron's Code. Initially, RC4 was trade secret but once it's code spread in the public domain it was no more a trade secret. While Ron did not

reveal the RC4 algorithm until 2014 when he described the history of RC4 in English Wikipedia.

Applications of RC4

RC4 is used in various applications such as WEP from 1997 and WPA from 2003. We also find applications of RC4 in SSL from 1995 and it is a successor of TLS from 1999. RC4 is used in varied applications because of its simplicity, speed, and simplified implementation in both software and hardware.

Types of RC4

There are various types of RC4 such as Spritz, RC4A, VMPC, and RC4A.

1. **SPRITZ:** Spritz can be used to build a cryptographic hash function, a deterministic random bit generator (DRBG), n an encryption algorithm that supports authenticated encryption with associated data (AEAD).
2. **RC4A:** Souraduyti Paul and Bart Preneel have proposed an RC4 variant, which they call RC4A, which is stronger than RC4.
3. **VMPC:** VMPC is another variant of RC4 which stands for Variably Modified Permutation Composition.

4. **RC4A+:** RC4A+ is a modified version of RC4 with a more complex three-phase key schedule which takes about three times as long as RC4 and a more complex output function which performs four additional lookups in the S array for each byte output, taking approximately 1.7 times as long as basic RC4.

Algorithm

The algorithm operates on a user-selected variable-length key(K) of 1 to 256 bytes (8 to 2048 bits), typically between 5 and 16 bytes. To generate a 256-byte state vector S, the master key is used.
The first step is the array initialization. It is a character array of size 256 i.e. S[256]. After that, for every element of the array, we initialize S[i] to i.

Code for array initialization:

```
Char S[256];
int i;
for(i=0;i<256;i++)
S[i] = i
The array will look like -
S[] = {0, 1, 2, 3, ------, 254, 255}
```

After this, we will run the **KSA algorithm-**

KSA is going to use the secret key to scramble this array. KSA is a simple loop, in which we are having two variable i and j. We are using these variables to rearrange the array. Rearranging the array is done by using a secret key.

Code for KSA (Key Scheduling Algorithm) :

```
int i, j=0;
for(i=0;i<256;i++)
{
j=( j + S[i] + T[i]) mod 256;
```

```
Swap(S[i], S[j]);
}
```

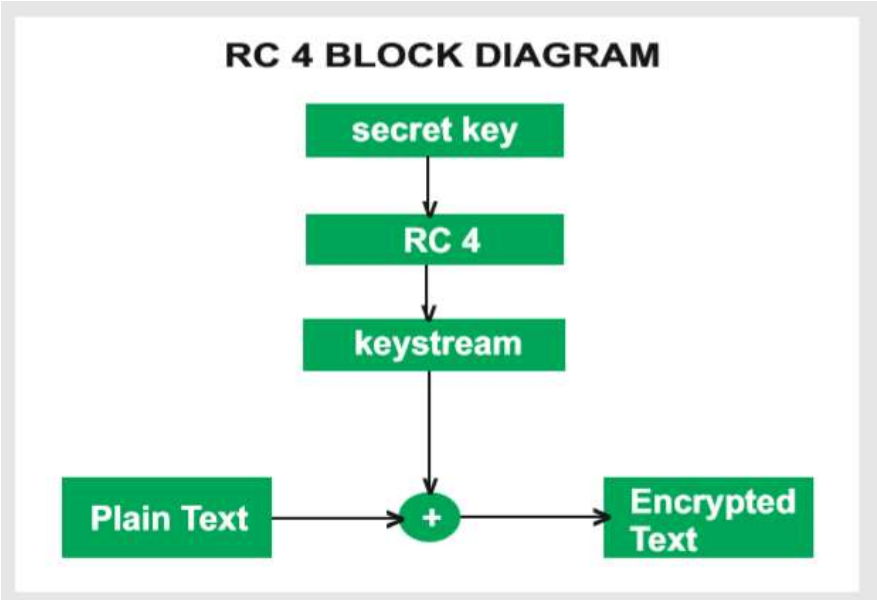
KSA has been scrambled, S[256] array is used to generate the PRGA(Pseudo Random Generation Algorithm). This is the actual Keystream.

Code for PRGA (Pseudo Random Generation Algorithm):

```
i=j=0;
while(true)
{
i = ( i + 1 ) mod 256;
j = ( j + S[i] ) mod 256;
Swap( S[i], S[j] );
t = ( S[i] + S[j] ) mod 256 ;
k = S[t];
}
```

This is the next step of scrambling.

RC4 Block Diagram



Working of RC4

Encryption Procedure

1. The user inputs a plain text file and a secret key.
2. The encryption engine then generates the keystream by using KSA and PRGA Algorithm.
3. This keystream is now XOR with the plain text, this XORing is done byte by byte to produce the encrypted text.
4. The encrypted text is then sent to the intended receiver, the intended receiver will then decrypted the text and after decryption, the receiver will get the original plain text.

Decryption Procedure

Decryption is achieved by doing the same byte-wise X-OR operation on the Ciphertext.

Example: Let A be the plain text and B be the keystream (A xor B) xor B = A

Advantages

1. RC4 stream ciphers are simple to use.
2. The speed of operation in RC4 is fast as compared to other ciphers.
3. RC4 stream ciphers are strong in coding and easy to implement.
4. RC4 stream ciphers do not require more memory.
5. RC4 stream ciphers are implemented on large streams of data.

Disadvantages

- If RC4 is not used with strong MAC then encryption is vulnerable to a bit-flipping attack.
- RC4 stream ciphers do not provide authentication.
- RC4 algorithm requires additional analysis before including new systems.
- RC4 stream ciphers cannot be implemented on small streams of data.

SHA-1 Hash

SHA-1 or Secure Hash Algorithm 1 is a cryptographic algorithm which takes an input and produces a 160-bit (20-byte) hash value. This hash value is known as a message digest. This message digest is usually then rendered as a hexadecimal number which is 40 digits long. It is a U.S. Federal Information Processing Standard and was designed by the United States National Security Agency. SHA-1 is now considered insecure since 2005. Major tech giants browsers like Microsoft, Google, Apple and Mozilla have stopped accepting SHA-1 SSL certificates by 2017. To calculate cryptographic hashing value in Java, **MessageDigest Class** is used, under the package **java.security**. MessageDigest Class provides following cryptographic hash function to find hash value of a text as follows:

- MD2
- MD5
- SHA-1
- SHA-224
- SHA-256
- SHA-384
- SHA-512

These algorithms are initialized in static method called **getInstance()**. After selecting the algorithm the message digest value is calculated and the results are returned as a byte array. BigInteger class is used, to convert the resultant byte array into its signum representation. This

representation is then converted into a hexadecimal format to get the expected MessageDigest. **Examples:**

> ***Input*** *: hello world* ***Output*** *:*
> *2aae6c35c94fcfb415dbe95f408b9ce91ee846ed* ***Input*** *:*
> *GeeksForGeeks* ***Output*** *:*
> *addf120b430021c36c232c99ef8d926aea2acd6b*

Below program shows the implementation of SHA-1 hash in Java.

Java

```
// Java program to calculate SHA-1 hash value

import java.math.BigInteger;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;

public class GFG {
    public static String encryptThisString(String input)
    {
        try {
            // getInstance() method is called with algorithm SHA-1
            MessageDigest md = MessageDigest.getInstance("SHA-1");
```

```
HashCode Generated by SHA-1 for:

GeeksForGeeks : addf120b430021c36c232c99ef8d926aea2acd6b

hello world : 2aae6c35c94fcfb415dbe95f408b9ce91ee846ed
```

Applications:

- **Cryptography**: The main application of SHA1 is to protect communications from being intercepted by outside parties. From a given data input, SHA1 generates a fixed-size, singular, and irreversible hash value. The integrity of the data can then be confirmed by comparing this hash value to the original hash value. This makes it possible to confirm that the data was not changed or tampered with in any manner during transmission.
- **Data Integrity**: In many industries, such as finance, healthcare, and government, data integrity is a major concern. Data integrity in a system is checked using the SHA1 algorithm. A fingerprint of the original data is created using a hash value produced by the SHA1 algorithm. If the data changes in any way, the hash value will also change, indicating that the data has been tampered with.
- **Digital Signatures**: Digital signatures are used to confirm the legitimacy of digital documents and messages. The digital document or communication is hashed using the SHA1 technique, and its hash value is subsequently encrypted with the sender's private key. Using the sender's public key to decode the message, the recipient can then compare the hash value to the original value.
- **Digital Forensics**: In digital forensics, a hash of a file containing digital evidence can be produced using the SHA1 algorithm. To ensure that the evidence hasn't been altered with during the investigation, utilize this hash value as proof. It gives proof that the file has not been altered if the hash values of the original file and the evidence file match.

- **Password Storage**: SHA1 can be used to save passwords. A hash of the password is generated using SHA1 when a user creates a password. The password itself is then substituted in a database for the hash value. The user's password is hashed with SHA1 when they attempt to log in, and the resulting hash is compared to a previously generated hash.
- **Software Updates**: The integrity of software updates can be guaranteed using SHA1. The SHA1 hash of the update file can be made public on the software vendor's website when an update is made available. By comparing the hash of the downloaded file with the published hash, users can download the update and ensure its integrity.

Last Updated : 31 May, 2023

25

Similar Reads

Bug in SHA-512 Hash Generation Java code

SHA-384 Hash In Java

SHA-224 Hash In Java

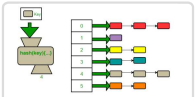
SHA-256 Hash in Java

Implement Secure Hashing Algorithm - 512 (SHA-512) as Functional…

What are Hash Functions and How to choose a good Hash Function?

Hash Functions and list/types of Hash functions

Advantages of BST over Hash Table



Implementing our Own Hash Table with Separate Chaining in Java

Data Structures | Hash | Question 1

Related Tutorials

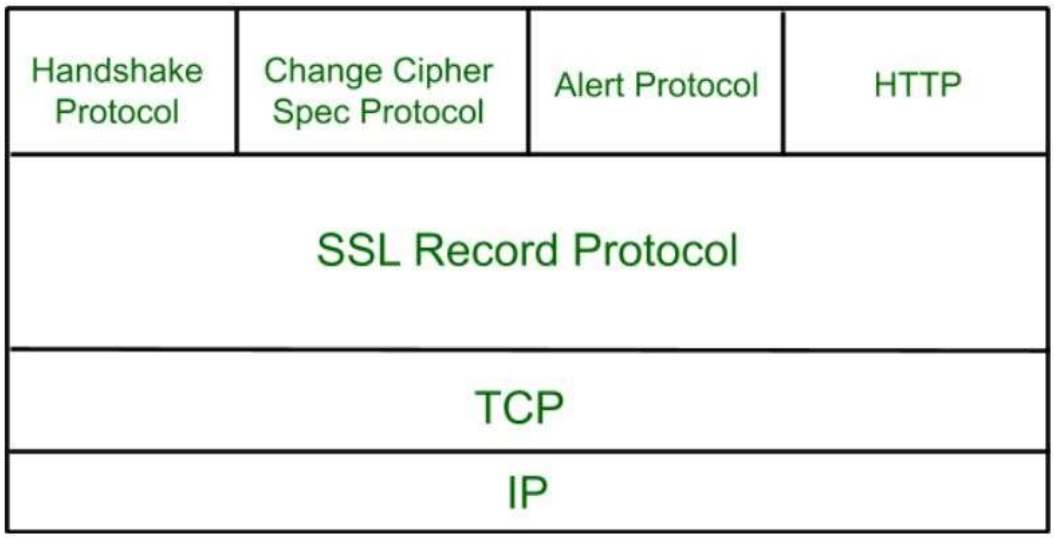
Secure Socket Layer (SSL)

Secure Socket Layer (SSL) provides security to the data that is transferred between web browser and server. SSL encrypts the link between a web server and a browser which ensures that all data passed between them remain private and free from attack.

Secure Socket Layer Protocols:

- SSL record protocol
- Handshake protocol
- Change-cipher spec protocol
- Alert protocol

SSL Protocol Stack:



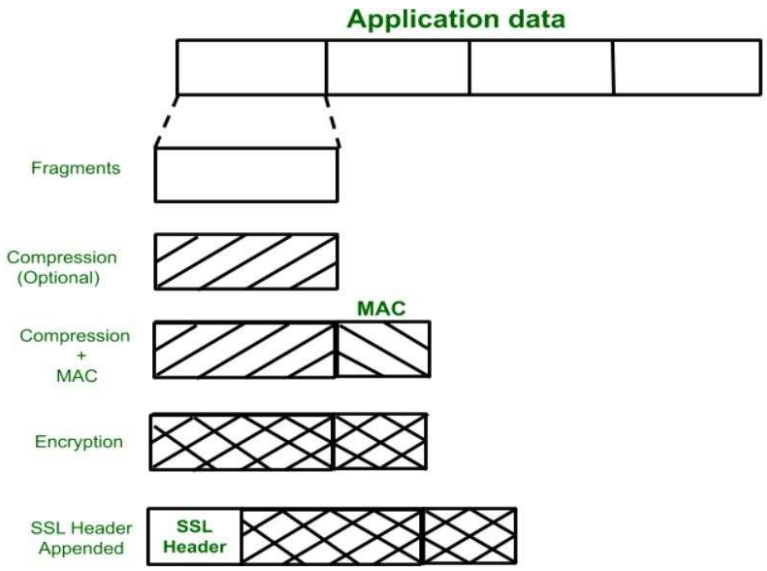
SSL Record Protocol:

SSL Record provides two services to SSL connection.

Spotlight

- Confidentiality
- Message Integrity

In the SSL Record Protocol application data is divided into fragments. The fragment is compressed and then encrypted MAC (Message Authentication Code) generated by algorithms like SHA (Secure Hash Protocol) and MD5 (Message Digest) is appended. After that encryption of the data is done and in last SSL header is appended to the data.

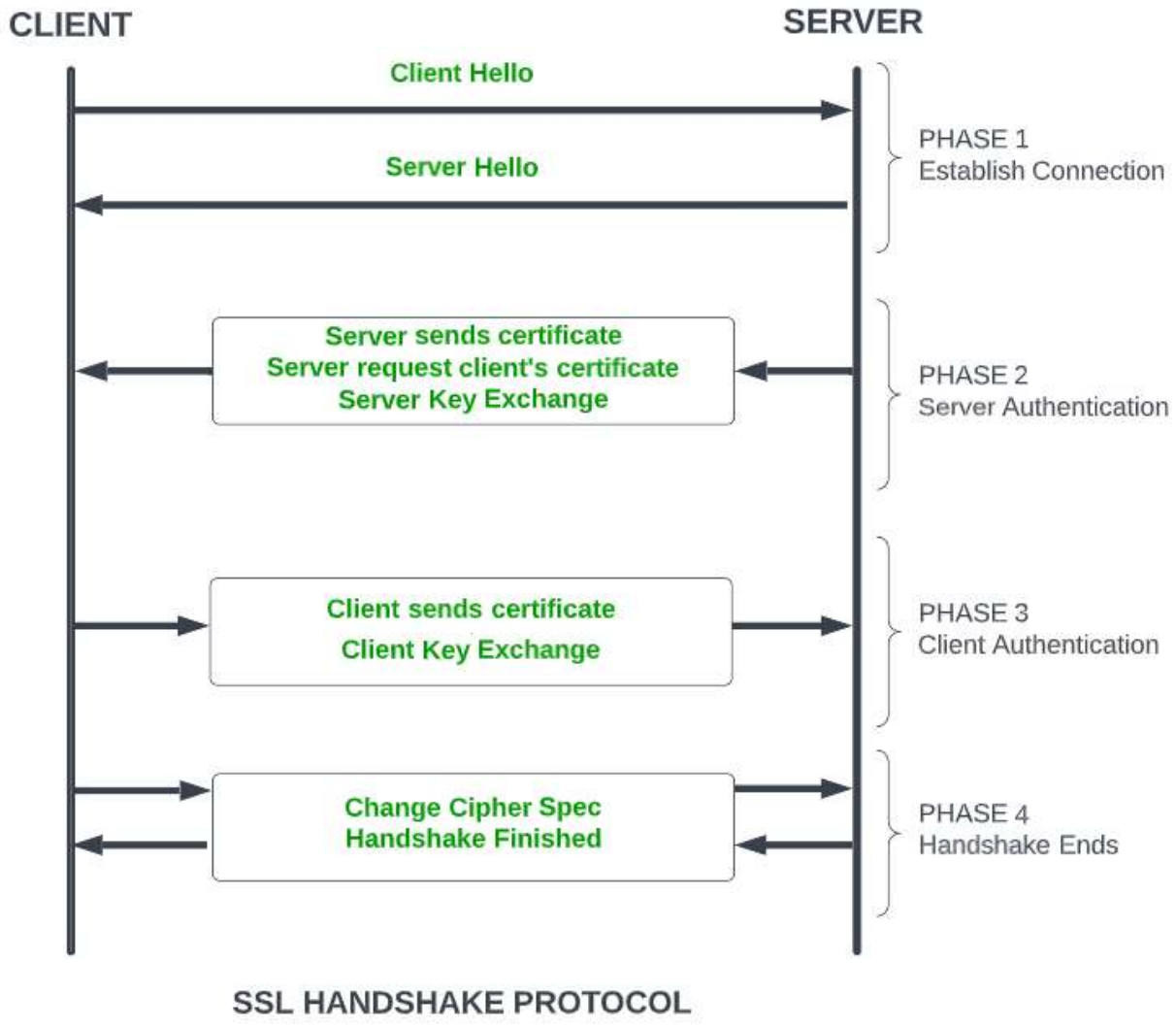


Handshake Protocol:

Handshake Protocol is used to establish sessions. This protocol allows the client and server to authenticate each other by sending a series of messages to

each other. Handshake protocol uses four phases to complete its cycle.

- **Phase-1:** In Phase-1 both Client and Server send hello-packets to each other. In this IP session, cipher suite and protocol version are exchanged for security purposes.
- **Phase-2:** Server sends his certificate and Server-key-exchange. The server end phase-2 by sending the Server-hello-end packet.
- **Phase-3:** In this phase, Client replies to the server by sending his certificate and Client-exchange-key.
- **Phase-4:** In Phase-4 Change-cipher suite occurs and after this the Handshake Protocol ends.



SSL Handshake Protocol Phases diagrammatic representation

Change-cipher Protocol:

This protocol uses the SSL record protocol. Unless Handshake Protocol is completed, the SSL record Output will be in a pending state. After the handshake protocol, the Pending state is converted into the current state. Change-cipher protocol consists of a single message which is 1 byte in length and can have only one value. This protocol's purpose is to cause the pending state to be copied into the current state.

Alert Protocol:

This protocol is used to convey SSL-related alerts to the peer entity. Each message in this protocol contains 2 bytes.

The level is further classified into two parts:

What is VPN? How It Works

VPN stands for the **Virtual Private Network**. A virtual private network (VPN) is a technology that creates a safe and encrypted connection over a less secure network, such as the Internet. A Virtual Private Network is a way to extend a private network using a public network such as the Internet. The name only suggests that it is a Virtual "private network,**i.e., a**" i.e. user can be part of a local network sitting at a remote location. It makes use of tunneling protocols to establish a secure connection.

How does a VPN work?

Let us understand VPN with Let's an example

Think of a situation where the corporate office of a bank is situated in Washington, USA. This office has a local network consisting of say 100 computers. Suppose other branches of the bank are in Mumbai, India, and Tokyo, Japan. The traditional method of establishing a secure connection between the head office and the **the** branch was to have a leased line between the branches and head office which was a very costly as well as troublesome job. VPN lets us overcome this issue in an effective manner.

The situation is described below

- All 100 hundred computers of the corporate office at Washington are connected to the VPN server(which is a well-configured server containing a public IP address and a switch to connect all computers present in the local network i.e. in US head office).
- The person sitting in the Mumbai office connects to The VPN server using a dial-up window and the VPN server returns an IP address that belongs to the series of IP addresses belonging to a local network of the corporate office.
- Thus person from the Mumbai branch becomes local to the head office and information can be shared securely over the public internet.
- So this is the intuitive way of extending the local network even across the geographical borders of the country.

VPN is well exploited all across the globe

We will explain to you with an example. Suppose we are using smartphones regularly. Spotify-a Swedish music app which is not active in India But we are making full use of it sitting in India. So how ?? VPN can be used to camouflage our geolocation.

- Suppose the Ip address is 101.22.23.3 which belongs to India. That's why our device is not able to access the Spotify music app.
- But the magic begins when we used the Psiphon app which is an android app and is used to change the device IP address to the IP address of the location we want(say US where Spotify works in a seamless manner).

IP address changed to an IP address belonging to USA

1. VPN also ensures security by providing an encrypted tunnel between client and VPN server.
2. VPN is used to bypass many blocked sites.
3. VPN facilitates Anonymous browsing by hiding your ip address.
4. Also, most appropriate Search engine optimization(SEO) is done by analyzing the data from VPN providers which provide country-wise stats of browsing a particular product. This method of SEO is used widely my many internet marketing managers to form new strategies.
5. VPNs encrypt your internet traffic, safeguarding your online activities from potential eavesdropping and cyber threats, thereby enhancing your privacy and data protection.

What is VPN used for?

Do you need help determining when you should use a VPN? Let us shed some light on the subject and show you how the best VPNs can revolutionize your online activities.

- For Unlimited Streaming: Love streaming your favourite shows and sports games? A VPN is your ultimate companion for unlocking streaming services like Netflix or Hulu. Access all the content you desire and never miss a moment of your beloved NFL games.
- For elevating your Gaming Experience: Unleash your gaming potential with the added layer of security and convenience provided by a VPN. Defend yourself against vengeful competitors aiming to disrupt your gameplay while improving your ping for smoother, lag-free sessions. Additionally, gain access to exclusive games that may be restricted in your region, opening up a world of endless gaming possibilities.
- For Anonymous Torrenting: When it comes to downloading copyrighted content through torrenting, it’s essential to keep your IP address hidden. A VPN can mask your identity and avoid potential exposure, ensuring a safe and private torrenting experience.
- For supercharging your Internet Speed: Are you tired of your Internet speed slowing down when downloading large files? Your Internet Service Provider (ISP) might be intentionally throttling your bandwidth. Thankfully, a VPN

Web Security Considerations

Web Security is very important nowadays. Websites are always prone to security threats/risks. Web Security deals with the security of data over the internet/network or web or while it is being transferred to the internet. For e.g. when you are transferring data between client and server and you have to protect that data that security of data is your web security.

Hacking a Website may result in the theft of Important Customer Data, it may be the credit card information or the login details of a customer or it can be the destruction of one's business and propagation of illegal content to the users while somebody hacks your website they can either steal the important information of the customers or they can even propagate the illegal content to your users through your website so, therefore, security considerations are needed in the context of web security.

Security Threats:

A Threat is nothing but a possible event that can damage and harm an information system. Security Threat is defined as a risk that which, can potentially harm Computer systems & organizations. Whenever an Individual or an Organization creates a website, they are vulnerable to security attacks.

Security attacks are mainly aimed at stealing altering or destroying a piece of personal and confidential information, stealing the hard drive space, and illegally accessing passwords. So whenever the website you created is vulnerable to security attacks then the attacks are going to steal your data alter your data destroy your personal information see your confidential informat and also it accessing your password.

Top Web Security Threats :

Web security threats are constantly emerging and evolving, but many threats consistently appear at the top of the list of web security threats. These include:

- Cross-site scripting (XSS)
- SQL Injection
- Phishing
- Ransomware
- Code Injection
- Viruses and worms
- Spyware
- Denial of Service

Security Consideration:

- **Updated Software:** You need to always update your software. Hackers may be aware of vulnerabilities in certain software, which are sometimes caused by bugs and can be used to damage your computer system and steal personal data. Older versions of software can become a gateway for hackers to enter your network. Software makers soon become aware of these vulnerabilities and will fix vulnerable or exposed areas. That's why It is mandatory to keep your software updated, It plays an important role in keeping your personal data secure.
- **Beware of SQL Injection:** SQL Injection is an attempt to manipulate your data or your database by inserting a rough code into your query. For e.g.

somebody can send a query to your website and this query can be a rough code while it gets executed it can be used to manipulate your database such as change tables, modify or delete data or it can retrieve important information also so, one should be aware of the SQL injection attack.

- **Cross-Site Scripting (XSS):** XSS allows the attackers to insert client-side script into web pages. E.g. Submission of forms. It is a term used to describe a class of attacks that allow an attacker to inject client-side scripts into other users' browsers through a website. As the injected code enters the browser from the site, the code is reliable and can do things like sending the user's site authorization cookie to the attacker.
- **Error Messages:** You need to be very careful about error messages which are generated to give the information to the users while users access the website and some error messages are generated due to one or another reason and you should be very careful while providing the information to the users. For e.g. login attempt – If the user fails to login the error message should not let the user know which field is incorrect: Username or Password.
- **Data Validation:** Data validation is the proper testing of any input supplied by the user or application. It prevents improperly created data from entering the information system. Validation of data should be performed on both server-side and client-side. If we perform data validation on both sides that will give us the authentication. Data validation should occur when data is received from an outside party, especially if the data is from untrusted sources.
- **Password:** Password provides the first line of defense against unauthorized access to your device and personal information. It is necessary to use a strong password. Hackers in many cases use sophisticated software that uses brute force to crack passwords. Passwords must be complex to protect against brute force. It is good to enforce password requirements such as a minimum of eight characters long must including uppercase letters, lowercase letters, special characters, and numerals.