Kalinga University

Course- BCSAIMLCS/BCAAIML **Sem-IV**
Subject- Cloud Computing and Its Security
Subject Code: BCSAIMLCS403 & BCAAIML402

UNIT-III

Confidentiality, Integrity, Availability, Authenticity, and Non-repudiation (often abbreviated as **"CIA"** or "**CIAAN**") are the five core security properties that are used to ensure the security and reliability of information systems. Together, they form the foundation of information security and are the key elements that must be protected in order to ensure the safe and secure handling of sensitive information.

1. **Confidentiality** is important to protect sensitive information from being disclosed to unauthorized parties. This includes protecting data at rest, in transit, and in use. Common techniques used to maintain confidentiality include encryption, access controls, and data masking.

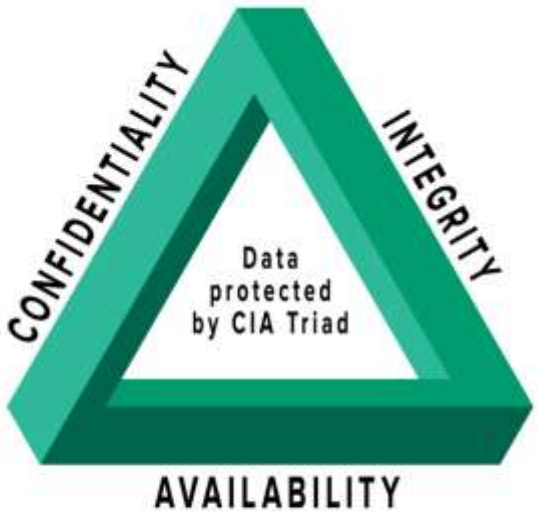
2. **Integrity** is important to ensure that information has not been tampered with or modified in an unauthorized way. This includes protecting data from unauthorized modification, deletion or addition. Common techniques used to maintain integrity include digital signatures, message authentication codes, and data hashing.

3. **Availability** is important to ensure that information and systems are accessible to authorized users when they need them. This includes protecting against denial of service attacks and ensuring that systems are highly available and can withstand failures. Common techniques used to maintain availability include load balancing, redundancy, and disaster recovery planning.

4. **Authenticity** is important to ensure that information and communication come from a trusted source. This includes protecting against impersonation, spoofing and other types of identity fraud. Common techniques used to establish authenticity include authentication, digital certificates, and biometric identification.

5. **Non-repudiation** is important to ensure that a party cannot deny having sent or received a message or transaction. This includes protecting against message tampering and replay attacks. Common techniques used to establish non-repudiation include digital signatures, message authentication codes and timestamps.

Together, these five properties form the foundation of information security and are critical to protecting the confidentiality, integrity, and availability of sensitive information.

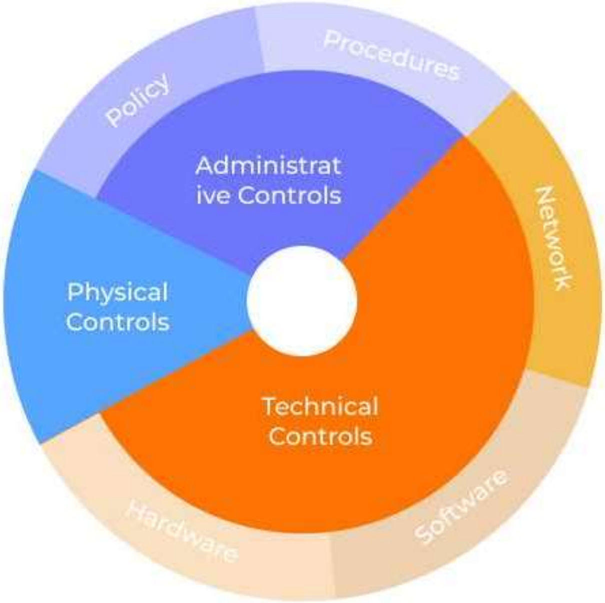


Why is Defense in Depth Important?

As the outcome of a successful cyber threat can be too detrimental, experts consider having various approaches in place. A single product is not enough to encounter every attack and danger around us.

With DiD concept, it’s easy and possible to deploy more than one cybersecurity measure for your systems, eliminating all the odds of security failures and threat penetrations.

If one security measure fails, by any chance, another comes into effect and protects the resources. So, network security is better and follows redundancy when DiD is at work.



Defense in Depth Security Architecture

Before talking about DiD's underlying security architecture, it's crucial to know that there is no standard format to follow as each organization will have varied needs. However, each defense in depth architecture is likely to feature one or many below-mentioned aspects.

- **Technical controls**

This set involves the usage of software and hardware capabilities to keep threats like DDoS attacks, data breaches, and other notorious threats at bay. Products like firewall, WAF, secure web gateway, IDS/IPS, EDR software, anti-malware software, and many others are used in this aspect.

- **Physical controls**

They aim to protect the IT systems, data centers, and physical assets from dangers like data theft, tampering, and non-permitted access. Practices like access control, alarm systems, ID scanners, and surveillance procedures are part of this section.

- **Administrative controls**

Administrative controls entail the security policies defined and governed by security teams and administrators so that internal systems/resources are protected.

Those who seek improved security can add a leveraged security layer with solutions like access measures, perimeter defenses, threat intelligence, monitoring/prevention, perimeter defenses, and workstation defenses.

How does defense-in-depth help?

Using the approach, it's doable to bring multiple security practices into action and reduce the possibilities of a data breach. In most cases, only one security layer is deployed at one point, which is not enough to provide overall protection.

For example, if a hacker succeeds in manipulating the network via bypassing the Cloud WAF or other security means, DiD will ensure that immediate and relevant countermeasures are in place.

This timely action keeps intermediate or futuristic dangers at bay. Additionally, it's also possible to have an around-the-clock security system activated to protect the resources.

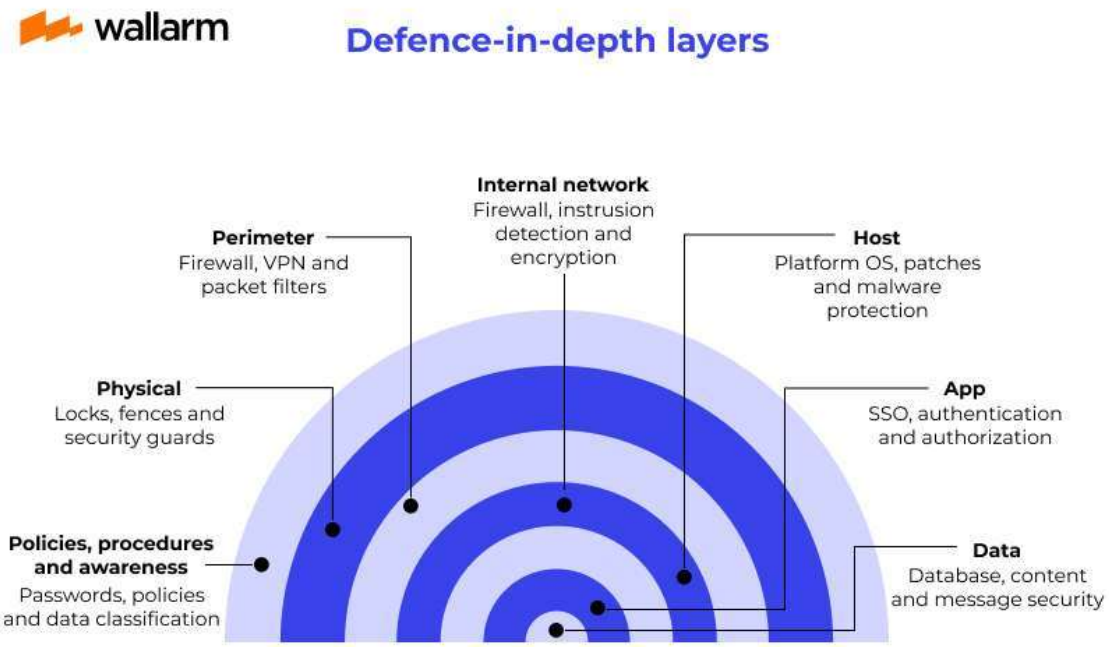
As many security layers are active, it increases the attackers' efforts and time. They have to work hard to bypass so many security measures before they lay a hand on your data/resources/network. Such extensiveness, achieved through defense in depth layers, is a huge demotivated factor for them. They will be fatigued and even stop carrying out the attacks on their own.

Defense in Depth vs Layered Security

As DiD and Layered Security have some similarities at a base level, it's obvious to consider both as the same terms/things. However, there are certain markers that set these two apart. We present you the basic layered security vs Defense in Depth review.

For instance, layered security practice combines various products to fix one security concern while the Defense In-Depth approach addresses a wide range of threats at a time.

The products used in layered security are likely to belong to the same security and tend to perform the same task. But, the Defense in Depth approach brings multiple products and practices together.


wallarm
Defence-in-depth layers
Internal network
Firewall, instrusion detection and encryption
Perimeter
Firewall, VPN and packet filters
Host
Platform OS, patches and malware protection
Physical
Locks, fences and security guards
App
SSO, authentication and authorization
Policies, procedures and awareness
Passwords, policies and data classification
Data
Database, content and message security

Elements of Defense in Depth (DiD)

When you wish to go to any length to protect your systems against vulnerabilities, DiD strategy presents various suggestions for your organization. Physical, technical and administration level controls are required. See the top components that could be part of your IT environment to safeguard it in depth:

1. **The Education about Cybersecurity Issues**

No matter how well you protect your network, if there exist unsecured endpoints, it can still be compromised. So, if you are leaving the human endpoints of your digital ecosystem unsafe and unaware of security threats and best practices, all efforts will be in vain.

As a part of DiD, we suggest you consider ISAT for your employees. Doing so will ensure that you have a smart and well-aware workforce, encouraging the business's cyber-resilience.

This training is to improve Internet Security awareness among enterprise users. However, the course syllabus is equally useful for all sizes of ventures, as it focuses on how to detect common threats, how to safeguard your critical organizational data, and how an attack can affect your operations/business adversely.

2. **An Efficient Access Management Mechanism**

To strengthen your organization's security posture, you must ensure that only the people requiring access to a resource/asset should have access to it – for the duration when it is essential. POLP states the same.

For this, you can opt for role-based or attribution-based privilege assignment. They are a good way to prevent misuse of admin or network user rights. Enabling requests' escalation and de-escalation by top-level or relevant users will ensure that no unwanted users/elements barge into your network.

3. **Antivirus & Firewall**

Securing against cyberthreats cannot be done until you have a virus-filtering and firewall method activated. These 2 elements will ensure that your devices remain untouched by troubles.

For example, a good anti-virus can save you against malware, adware, eavesdropping, trojans, DDoS, worms, and other sorts of intrusions. A Firewall, on the other hand, can secure you against the threats entering your network, pretending to be a part of good traffic.

4. **Password Protection and Management**

In the case of passwords, you need to take care of various things.

From maintaining the strength of all your passwords as 'strong' to saving them with a reliable password manager, deploying a trustworthy authentication method (like 2FA), and encrypting/hashing them using an efficient algorithm - you cannot miss out on any front. Additionally, if you are using biometric data, it should be kept in a secure database too.

5. **Intrusion Prevention System (IPS)**

Monitoring the traffic, preventing intrusions, and detecting malicious actors at the earliest in your network are the top ways of minimizing the impact of cyberthreats on your organization. Also, finding security loopholes or API vulnerabilities is very important too. So, make sure that you have an IPS to ensure the same.

A reliable IPS tool is capable of blocking network-traffic, alerting you about threats, resetting connections, discarding fishy data packets, and taking other similar actions to safeguard your network. An ML-enabled solution can learn about new classes of threats and become more effective progressively.

6. **Network Segmentation**

A vast network with everything controlled at a centralized level is at a bigger risk of exposure. One unsecured endpoint may result in the whole network's compromise. To prevent the same, it is essential that you design multiple subnets and implement different levels of security for each of them. It will also be good from the cost management perspective.

7. **Patch Management**

Most of the cybercriminals make use of outdated and unpatched applications. It is easy to take advantage of a software application with loopholes and security issues. In fact, more than 20% compromises, in 2020 alone, happened due to outdated systems and hardware in enterprise networks Hence, an effective patch management process needs to be placed.

8. **API Security**

Often overlooked by enterprises, API Protection is super-essential for any business considering the 100% wellbeing of its network. Using a DevSecOps platform like Wallarm can be your help. Keeping DDoS attacks and bots away from your SOAP or REST, graphQL, and gRPC APIs, its real-time threat detection feature is very useful.

How to implement least privilege access in the cloud

As cloud becomes the norm rather than the exception, identity -- specifically, privilege allocation -- is still the elephant in the room. Organizations are creating more complex cloud infrastructures even as they employ a wider variety of services. But they are also finding themselves saddled with overly permissive privilege models.

The principle of least privilege -- a cornerstone in on-site identity and access management (IAM) -- should be extended to the cloud to maintain security and ensure users and devices only have access to those resources necessary to complete their jobs.

Fortunately, there are a variety of cloud least privilege practices organizations can implement and manage to address the elephant in the room.

Control and manage cloud policies

The biggest issue organizations face is how easy it is to exploit cloud policies to allocate privileges. For many years, security professionals have railed against the overallocation of privileges with servers and applications on site. It's also always true that job functions are easier to fulfill when you're the root or admin user. But that isn't secure.

This is a common problem in the cloud too. DevOps and cloud engineering teams often deploy infrastructure that "just works" with identity policies and is entirely too permissive.

To counteract privilege creep in the cloud and other cloud access security pitfalls, security teams should do the following:

- **Monitor cloud services that monitor.** Enable and monitor all cloud-native security services that monitor cloud IAM policies and provide alerting and guidance on privilege reduction. These include AWS IAM Access Analyzer, Google Cloud Policy Analyzer for IAM and Azure role-based access control policy analytics in Azure Policy. Such services provide insight and reporting into what privileges are defined, where they're allocated and how current privilege levels could be reduced to improve security.
- **Use cloud security posture management (CSPM).** Consider a CSPM service that integrates with IaaS and PaaS clouds used by the organization and continually scans for configuration issues and vulnerabilities -- which can easily include IAM policies.
- **Secure SaaS with a CSPM, SaaS security posture management (SSPM) or cloud access security broker (CASB).** For SaaS clouds, use a CSPM, SSPM or CASB that can help identify any vulnerabilities in how privileges are allocated.
- **Evaluate cloud security analysis platforms.** Consider a dedicated cloud security analysis platform that focuses on identity. These may not technically fall into the CSPM or SSPM categories. But they can analyze the entire set of interrelated policies defined and implemented within a cloud environment.

Symmetric Key Cryptography

Symmetrical Key Cryptography also known as conventional or single-key encryption was the primary method of encryption before the introduction of public key cryptography in the 1970s. In symmetric-key algorithms, the same keys are used for data encryption and decryption. This type of cryptography plays a crucial role in securing data because the same key is used for both encryption and decryption.

In this article, we will cover the techniques used in symmetric key cryptography, its applications, principles on which it works, its types and limitations as well as what type of attacks in the digital world it gets to face.

Techniques Used in Symmetric Key Cryptography

Substitution and **Transposition** are two principal techniques used in symmetric-key cryptography.

Substitution Techniques

The symmetric key cryptographic method employs one secret key for the operations of encryption and decryption. Substitution techniques provide two significant approaches, wherein elements (letters, characters) from the plaintext message are replaced with new elements according to the rules based on the secret key.

- **Caesar Cipher:** Caesar cipher has since their predictability is so complete and no complexity is invested.
- **Monoalphabetic Ciphers:** This is where the ciphers use one rule of substitution throughout the message. This may involve replacing letters with numbers, symbols, or another set of letters in another order.
- **Playfair Cipher:** Implementation of repeated letters or letter pairs can expose patterns, and cryptanalysis techniques exist to exploit them.
- **Hill Cipher:** This cipher operates on blocks of letters (typically bigrams or trigrams) using a matrix multiplication approach. The Hill ciphers have a limitation on key size and susceptibility towards cryptanalysis for larger key sizes.
- **Polyalphabetic Ciphers:** This is the type of cipher where any one of the letters in the plaintext is substituted by a different letter to keep frequency analysis challenging. For example, the Vigenère cipher operates with a keyword that would determine the shift value for each letter in the plaintext.
- **One-Time Pad (OTP):** It is a theoretically impossible cipher where the key is a random string of characters that is exactly as long as the message itself. The key is used for a single encryption and then discarded.

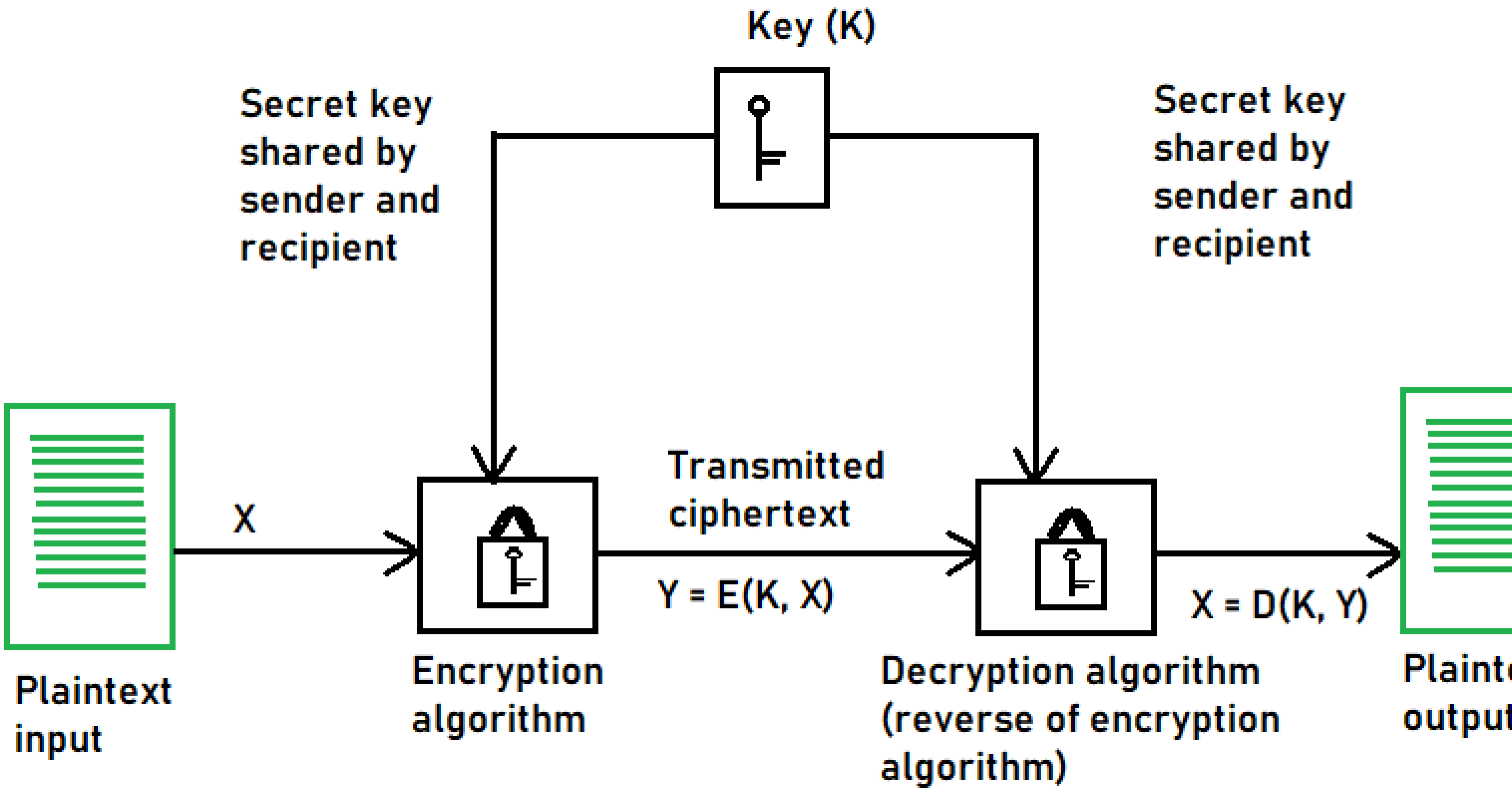

Key (K)
Secret key shared by sender and recipient
Secret key shared by sender and recipient
X
Transmitted ciphertext
Y = E(K, X)
X = D(K, Y)
Plaintext input
Encryption algorithm
Decryption algorithm (reverse of encryption algorithm)
Plaint
outpu

Diagram of Symmetric Encryption

Transposition Techniques

Transposition techniques rearrange the order of elements in the plaintext message without changing the elements themselves.

- **Rail Fence Cipher:** This is a simple cipher that rearranges the elements by writing the plaintext message in a zigzag pattern, with the different components written in rows (rails) of an imaginary fence and then reading through the columns in a standard order. The key to this is the number of rails used.

- **Columnar Transposition:** In the case of a plaintext message written in columns and then the columns rearranged according to a permutation determined by the key, this cipher is known as columnar transposition. Although it is still vulnerable to cryptanalysis techniques that exploit the statistical properties of the language.

Types of Symmetric Key Cryptography

1. Stream Ciphers
2. Block Ciphers

Stream Ciphers

The encryption process begins with the stream cipher's algorithm generating a pseudo-random keystream made up of the encryption key and the unique randomly generated number known as the nonce. The result is a random stream of bits corresponding to the length of the ordinary plaintext. Then, the ordinary plaintext is also deciphered into single bits.

These bits are then joined one by one to the keystream bits, gradually converting the ordinary plaintext into the ciphertext using the XOR bitwise operations. When the recipient wants to decrypt the encrypted plaintext, they must generate a new keystream made during the encryption. The encrypted plaintext is then deciphered one by one to derive the encrypted plaintext at the recipient's end.

Block Cipher

The result of a block cipher is a sequence of blocks that are then encrypted with the key. The output is a sequence of blocks of encrypted data in a specific order. When the ciphertext travels to its endpoint, the receiver uses the same cryptographic key to decrypt the ciphertext blockchain to the plaintext message.

The most common block cipher algorithms are

Advanced Encryption Standard (AES)

- It has support for three-length keys: 128 bits, 192 bits, or 256 bits, the most commonly used one is a 128-bit key.
- It includes secure communication, data encryption in storage devices, digital rights management (DRM), and so on.

Data Encryption Standard (DES)

- In DES, the 64-bit blocks of plaintext are encrypted using a 56-bit key.
- This weakness caused by the small key size led to the development of a more secure algorithm, called AES.

Triple Data Encryption Algorithm (Triple DES)

- The development of the Triple DES, also called Triple-DES or TDEA, was triggered by the weak security resulting from the small key size in the DES.
- Triple DES denotes a method of three times applying the DES algorithm sequentially (encrypt-decrypt-encrypt) on every plaintext block.

Principles of Symmetric Key Encryption

Basic principles which underpin the security of symmetric key encryption algorithms.

- **Resistance to brute force attack:** The most basic requirement for the security of an encryption cipher is that the keyspace size—in other words, the number of possible distinct keys from which someone using the algorithm could have chosen—is very large.
- **Cryptographic attack resistance:** The second fundamental requirement for symmetric or non-symmetrical encryption is the ability to generate information-influenced (i.e., non-random) encrypted messages. For this to happen, a critical but not sufficient requirement in an informational sense is that the encrypted message has high entropy.

Advantages of Symmetric Key Cryptography

- **Speed and efficiency**: Symmetric key+ algorithms are better suited for encrypting large volumes of data or for use in real-time communication scenarios as they are faster and less resource-intensive than asymmetric cryptography. SKC algorithms do not involve algebraically mathematical operations.
- **Scalability**: Because symmetric key algorithms have relatively low computational overhead, they scale well with the number of users and the amount of data being encrypted.
- **Simplicity**: Symmetric encryption protocols are often more straightforward to implement and understand than asymmetric key methods, and this would go a long way in attracting developers and users.

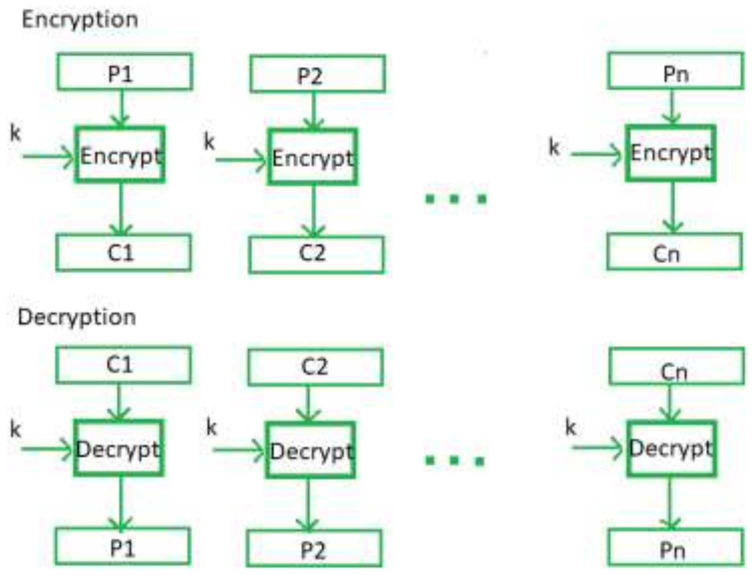
Block Cipher modes of Operation

Encryption algorithms are divided into two categories based on the input type, as a block cipher and stream cipher. **Block cipher** is an encryption algorithm that takes a fixed size of input say *b* bits and produces a ciphertext of *b* bits again. If the input is larger than *b* bits it can be divided further. For different applications and uses, there are several modes of operations for a block cipher.

Electronic Code Book (ECB) –

Electronic code book is the easiest block cipher mode of functioning. It is easier because of direct encryption of each block of input plaintext and output is in form of blocks of encrypted ciphertext. Generally, if a message is larger than *b* bits in size, it can be broken down into a bunch of blocks and the procedure is repeated.

Procedure of ECB is illustrated below:



Advantages of using ECB –

- Parallel encryption of blocks of bits is possible, thus it is a faster way of encryption.
- Simple way of the block cipher.

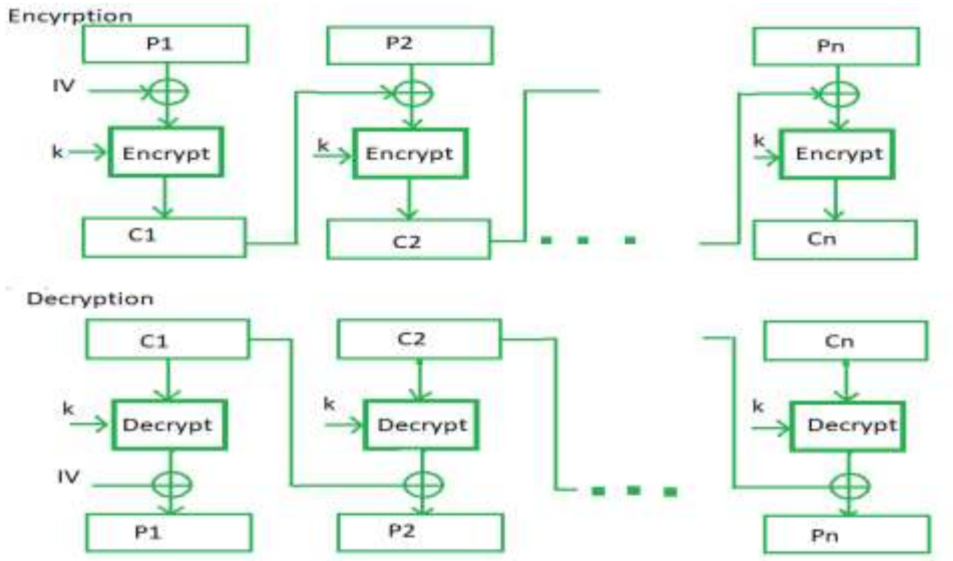
Disadvantages of using ECB –

- Prone to cryptanalysis since there is a direct relationship between plaintext and cipher text.
-

Cipher Block Chaining –

Cipher block chaining or CBC is an advancement made on ECB since ECB compromises some security requirements. In CBC, the previous cipher block is given as input to the next encryption algorithm after XOR with the original plaintext block. In a nutshell here, a cipher block is produced by encrypting an XOR output of the previous cipher block and present plaintext block.

The process is illustrated here:



Advantages of CBC –

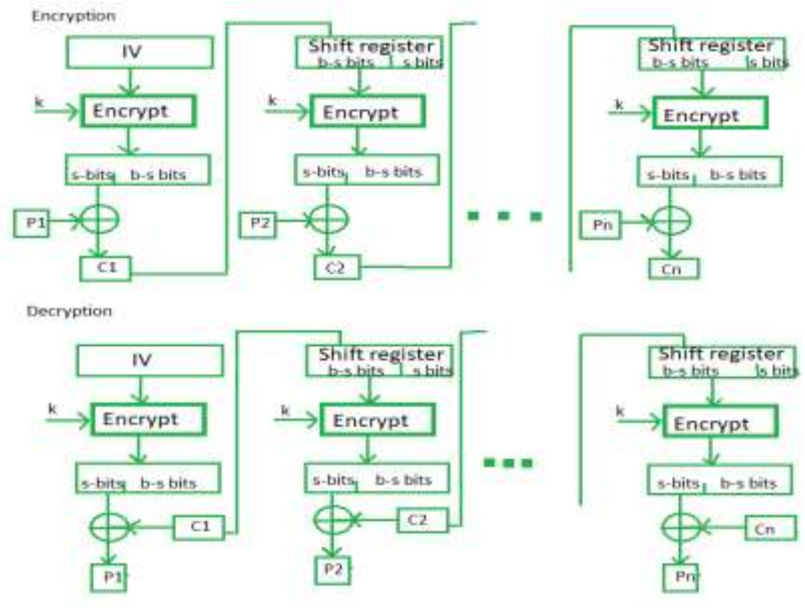
- CBC works well for input greater than *b* bits.
- CBC is a good authentication mechanism.
- Better resistive nature towards cryptanalysis than ECB.
-

Disadvantages of CBC –

- Parallel encryption is not possible since every encryption requires a previous cipher.
-

Cipher Feedback Mode (CFB) –

In this mode the cipher is given as feedback to the next block of encryption with some new specifications: first, an initial vector IV is used for first encryption and output bits are divided as a set of *s* and *b-s* bits.The left-hand side *s* bits are selected along with plaintext bits to which an XOR operation is applied. The result is given as input to a shift register having b-s bits to lhs,s bits to rhs and the process continues. The encryption and decryption process for the same is shown below, both of them use encryption algorithms.



Advantages of CFB –

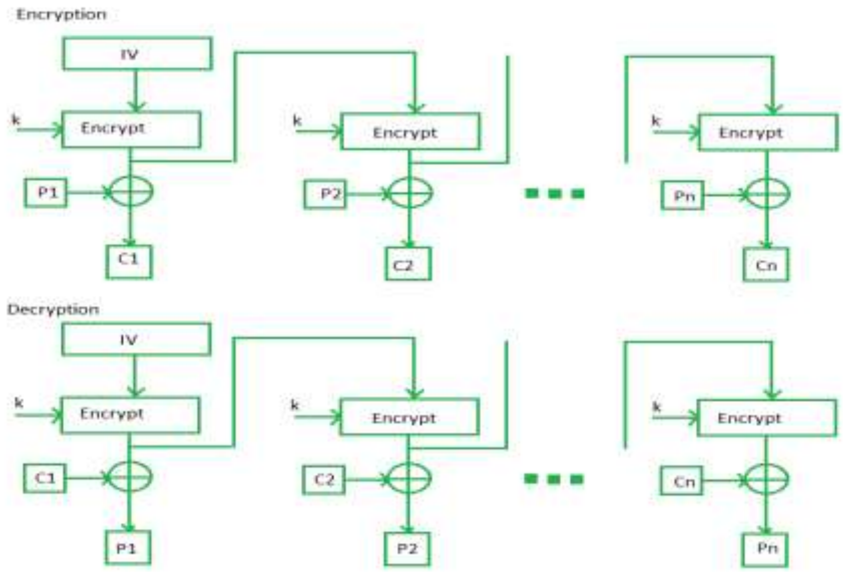
- Since, there is some data loss due to the use of shift register, thus it is difficult for applying cryptanalysis.
-

Disadvantages of using CFB –

- The drawbacks of CFB are the same as those of CBC mode. Both block losses and concurrent encryption of several blocks are not supported by the encryption. Decryption, however, is parallelizable and loss-tolerant.

Output Feedback Mode –

The output feedback mode follows nearly the same process as the Cipher Feedback mode except that it sends the encrypted output as feedback instead of the actual cipher which is XOR output. In this output feedback mode, all bits of the block are sent instead of sending selected *s* bits. The Output Feedback mode of block cipher holds great resistance towards bit transmission errors. It also decreases the dependency or relationship of the cipher on the plaintext.



Advantages of OFB –

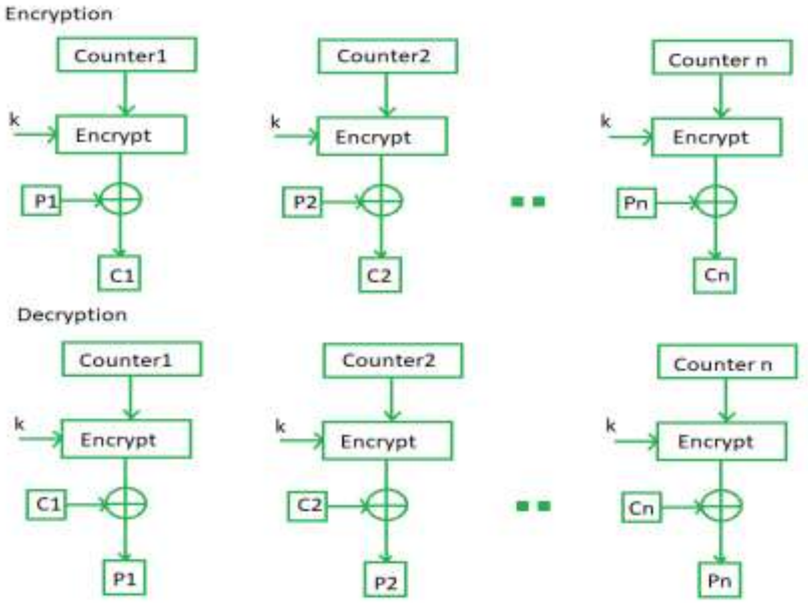
- In the case of CFB, a single bit error in a block is propagated to all subsequent blocks. This problem is solved by OFB as it is free from bit errors in the plaintext block.
-

Disadvantages of OFB-

- The drawback of OFB is that, because to its operational modes, it is more susceptible to a message stream modification attack than CFB.

Counter Mode –

The Counter Mode or CTR is a simple counter-based block cipher implementation. Every time a counter-initiated value is encrypted and given as input to XOR with plaintext which results in cipher text block. The CTR mode is independent of feedback use and thus can be implemented in parallel.
Its simple implementation is shown below:



Advantages of Counter –

- Since there is a different counter value for each block, the direct plaintext and ciphertext relationship is avoided. This means that the same plain text can map to different ciphertext.
- Parallel execution of encryption is possible as outputs from previous stages are not chained as in the case of CBC.

Disadvantages of Counter-

- The fact that CTR mode requires a synchronous counter at both the transmitter and the receiver is a severe drawback. The recovery of plaintext is erroneous when synchronisation is lost.

Applications of Block Ciphers

1. **Data Encryption:** Block Ciphers are widely used for the encryption of private and sensitive data such as passwords, credit card details and other information that is transmitted or stored for a communication. This encryption process converts a plain data into non-readable and complex form. Encrypted data can be decrypted only by the authorised person with the private keys.
2. **File and Disk Encryption:** Block Ciphers are used for encryption of entire files and disks in order to protect their contents and restrict from unauthorised users. The disk encryption softwares such as BitLocker, TrueCrypt aslo uses block cipher to encrypt data and make it secure.
3. **Virtual Private Networks (VPN):** Virtual Private Networks (VPN) use block cipher for the encryption of data that is being transmitted between the two communicating devices over the internet. This process makes sure that data is not accessed by unauthorised person when it is being transmitted to another user.
4. **Secure Sockets Layer (SSL) and Transport Layer Security (TLS):** SSL and TLS protocols use block ciphers for encryption of data that is transmitted between web browsers and servers over the internet. This encryption process provides security to confidential data such as login credentials, card information etc.
5. **Digital Signatures:** Block ciphers are used in the digital signature algorithms, to provide authenticity and integrity to the digital documents. This encryption process generates the unique signature for each document that is used for verifying the authenticity and detecting if any malicious activity is detected.

Public Key Encryption

Public key cryptography provides a secure way to exchange information and authenticate users by using pairs of keys. The public key is used for encryption and signature verification, while the private key is used for decryption and signing. When the two parties communicate with each other to transfer the intelligible or sensible message, referred to as plaintext, is converted into apparently random unreadable for security purposes referred to as **ciphertext**.

What is Public Key Cryptography?

Public key cryptography is a method of secure communication that uses a pair of keys, a public key, which anyone can use to encrypt messages or verify signatures, and a private key, which is kept secret and used to decrypt messages or sign documents. This system ensures that only the intended recipient can read an encrypted message and that a signed message truly comes from the claimed sender. Public key cryptography is essential for secure internet communications, allowing for confidential messaging, authentication of identities, and verification of data integrity.

What is a Cryptographic Key?

A cryptographic key is a piece of information used by cryptographic algorithms to encrypt or decrypt data, authenticate identities, or generate digital signatures. It serves as a parameter to control cryptographic operations, ensuring the security and privacy of digital communications and transactions.

How Does TLS/SSL Use Public Key Cryptography?

TLS/SSL uses public key cryptography to keep our internet connections secure. It does this in two main ways:

1. **Encryption**: When you visit a secure website (HTTPS), TLS/SSL helps encrypt data exchanged between your browser and the website's server. It uses a combination of public and private keys to create a secure connection. Your browser and the server agree on a secret key for this session, which keeps your data safe from eavesdroppers.
2. **Authentication**: TLS/SSL verifies the identity of websites. When you connect to a site, it presents a digital certificate signed by a trusted authority. Your browser checks this certificate to ensure you're really connecting to the right site and not a fake one trying to steal your information.

By using public key cryptography, TLS/SSL protects our privacy online and ensures that the websites we visit are genuine and trustworthy.

Encryption

The process of changing the plaintext into the ciphertext is referred to as **encryption.** The encryption process consists of an algorithm and a key. The key is a value independent of the plaintext.

The security of conventional encryption depends on the major two factors

1. The Encryption algorithm
2. Secrecy of the key

Once the ciphertext is produced, it may be transmitted. The Encryption algorithm will produce a different output depending on the specific key being used at the time. Changing the key changes the output of the algorithm.
Once the ciphertext is produced, it may be transmitted. Upon reception, the ciphertext can be transformed back to the original plaintext by using a decryption algorithm and the same key that was used for encryption.

Decryption

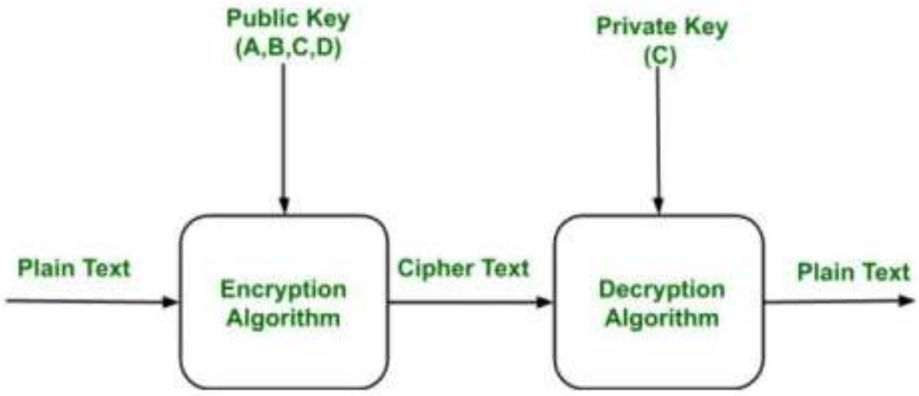
The process of changing the ciphertext to the plaintext that process is known as **decryption**.

Public Key Encryption : Asymmetric is a form of Cryptosystem in which encryption and decryption are performed using different keys-Public key (known to everyone) and Private key (Secret key). This is known as **Public Key Encryption.**

Characteristics of Public Encryption key

- Public key Encryption is important because it is infeasible to determine the decryption key given only the knowledge of the cryptographic algorithm and encryption key.
- Either of the two keys (Public and Private key) can be used for encryption with other key used for decryption.
- Due to Public key cryptosystem, public keys can be freely shared, allowing users an easy and convenient method for encrypting content and verifying digital signatures, and private keys can be kept secret, ensuring only the owners of the private keys can decrypt content and create digital signatures.
- The most widely used public-key cryptosystem is RSA (Rivest–Shamir–Adleman). The difficulty of finding the prime factors of a composite number is the backbone of RSA.

Example:
Public keys of every user are present in the Public key Register. If B wants to send a confidential message to C, then B encrypt the message using C Public key. When C receives the message from B then C can decrypt it using its own Private key. No other recipient other than C can decrypt the message because only C know C’s private key.



Public Key Encryption

Components of Public Key Encryption

- **Plain Text:** This is the message which is readable or understandable. This message is given to the Encryption algorithm as an input.
- **Cipher Text:** The cipher text is produced as an output of Encryption algorithm. We cannot simply understand this message.
- **Encryption Algorithm:** The encryption algorithm is used to convert plain text into cipher text.
- **Decryption Algorithm:** It accepts the cipher text as input and the matching key (Private Key or Public key) and produces the original plain text
- **Public and Private Key:** One key either Private key (Secret key) or Public Key (known to everyone) is used for encryption and other is used for decryption

Weakness of the Public Key Encryption

- Public key Encryption is vulnerable to Brute-force attack.
- This algorithm also fails when the user lost his private key, then the Public key Encryption becomes the most vulnerable algorithm.
- Public Key Encryption also is weak towards man in the middle attack. In this attack a third party can disrupt the public key communication and then modify the public keys.
- If user private key used for certificate creation higher in the PKI (Public Key Infrastructure) server hierarchy is compromised, or accidentally disclosed, then a “man-in-the-middle attack” is also possible, making any subordinate certificate wholly insecure. This is also the weakness of public key Encryption.

Applications of the Public Key Encryption

- **Encryption/Decryption:** Confidentiality can be achieved using Public Key Encryption. In this the Plain text is encrypted using receiver public key. This will ensure that no one other than receiver private key can decrypt the cipher text.
- **Digital signature:** Digital signature is for senders authentication purpose. In this sender encrypt the plain text using his own private key. This step will make sure the authentication of the sender because receiver can decrypt the cipher text using senders public key only.
- **Key exchange:** This algorithm can use in both Key-management and securely transmission of data.

Public Key Infrastructure

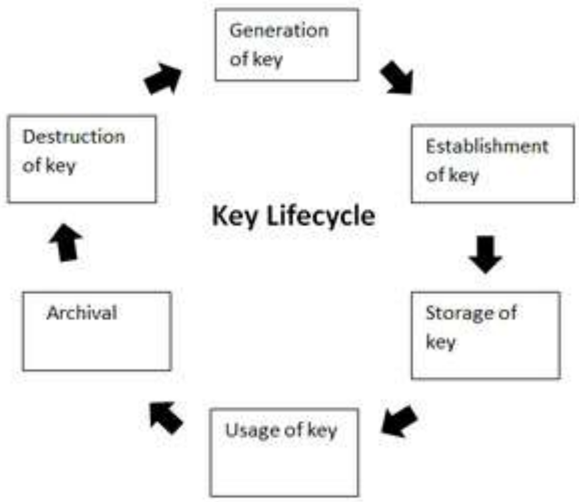
Public key infrastructure or PKI is the governing body behind issuing digital certificates. It helps to protect confidential data and gives unique identities to users and systems. Thus, it ensures security in communications.

The public key infrastructure uses a pair of keys: the public key and the private key to achieve security. The public keys are prone to attacks and thus an intact infrastructure is needed to maintain them.

Managing Keys in the Cryptosystem:

The security of a cryptosystem relies on its keys. Thus, it is important that we have a solid key management system in place. The 3 main areas of key management are as follows:

- A cryptographic key is a piece of data that must be managed by secure administration.
- It involves managing the key life cycle which is as follows:



- Public key management further requires:
  - **Keeping the private key secret:** Only the owner of a private key is authorized to use a private key. It should thus remain out of reach of any other person.
  - **Assuring the public key:** Public keys are in the open domain and can be publicly accessed. When this extent of public accessibility, it becomes hard to know if a key is correct and what it will be used for. The purpose of a public key must be explicitly defined.

PKI or public key infrastructure aims at achieving the assurance of public key.

Public Key Infrastructure:

Public key infrastructure affirms the usage of a public key. PKI identifies a public key along with its purpose. It usually consists of the following components:

- A digital certificate also called a public key certificate
- Private Key tokens
- Registration authority
- Certification authority
- CMS or Certification management system

Working on a PKI:

Let us understand the working of PKI in steps.

- **PKI and Encryption:** The root of PKI involves the use of cryptography and encryption techniques. Both symmetric and asymmetric encryption uses a public key. The challenge here is – “how do you know that the public key belongs to the right person or to the person you think it belongs to?”. There is always a risk of MITM(Man in the middle). This issue is resolved by a PKI using digital certificates. It gives identities to keys in order to make the verification of owners easy and accurate.
- **Public Key Certificate or Digital Certificate:** Digital certificates are issued to people and electronic systems to uniquely identify them in the digital world. Here are a few noteworthy things about a digital certificate. Digital certificates are also called X.509 certificates. This is because they are based on the ITU standard X.509.

  - The Certification Authority (CA) stores the public key of a user along with other information about the client in the digital certificate. The information is signed and a digital signature is also included in the certificate.
  - The affirmation for the public key then thus be retrieved by validating the signature using the public key of the Certification Authority.
- **Certifying Authorities:** A CA issues and verifies certificates. This authority makes sure that the information in a certificate is real and correct and it also digitally signs the certificate. A CA or ***Certifying Authority performs these basic roles***:
  - Generates the key pairs – This key pair generated by the CA can be either independent or in collaboration with the client.
  - Issuing of the digital certificates – When the client successfully provides the right details about his identity, the CA issues a certificate to the client. Then CA further signs this certificate digitally so that no changes can be made to the information.
  - Publishing of certificates – The CA publishes the certificates so that the users can find them. They can do this by either publishing them in an electronic telephone directory or by sending them out to other people.
  - Verification of certificate – CA gives a public key that helps in verifying if the access attempt is authorized or not.
  - Revocation – In case of suspicious behavior of a client or loss of trust in them, the CA has the power to revoke the digital certificate.

How Hashing Algorithm Used in Cryptography?

A Hash Function (H) takes a variable-length block of data and returns a hash value of a fixed size. A good hash function has a property that when it is applied to a large number of inputs, the outputs will be evenly distributed and appear random. Generally, the primary purpose of a hash function is to maintain data integrity. Any change to any bits or bits in the results will result in a change in the hash code, with a high probability.

The type of hash function that is needed for security purposes is called a **cryptographic hash function.**

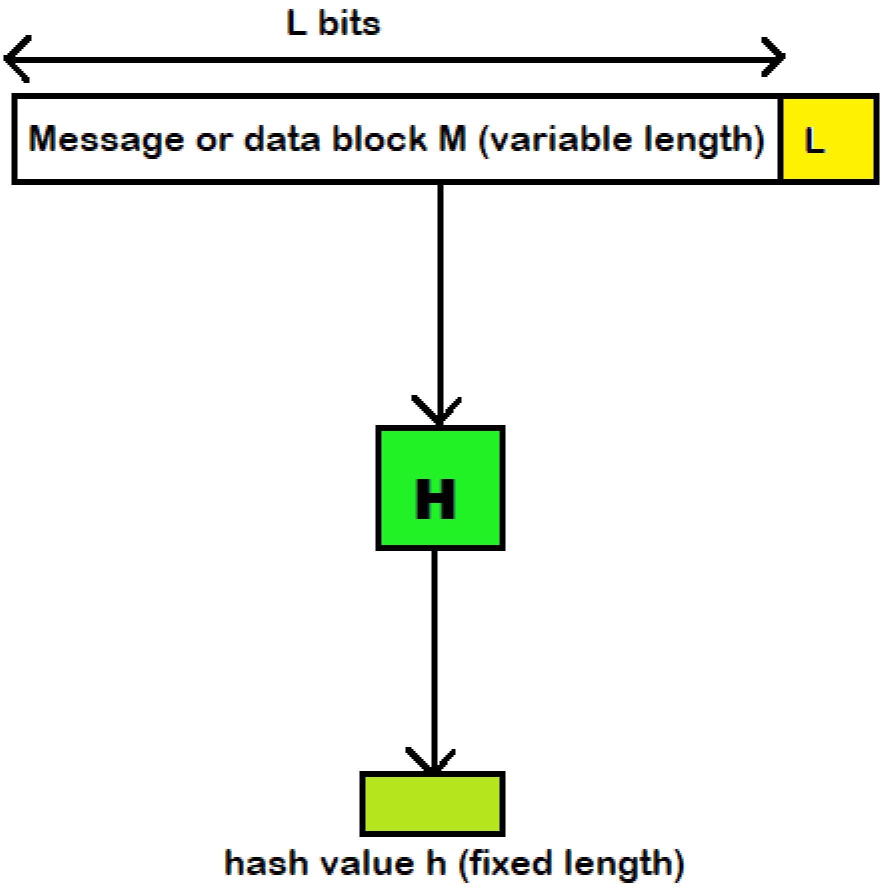
A Hash Function (H) takes a variable-length block of data and returns a hash value of a fixed size. A good hash function has a property that when it is applied to a large number of inputs, the outputs will be evenly distributed and appear random. Generally, the primary purpose of a hash function is to maintain data integrity. Any change to any bits or bits in the results will result in a change in the hash code, with a high probability.

The type of hash function that is needed for security purposes is called a **cryptographic hash function.**

A cryptographic hash function (or cryptographic hash algorithm) is an algorithm that is not computationally efficient (no attack is more efficient than brute force) when it is used to find either:

- A data object which maps to a predefined hash result
- Two data objects that map to the hash result in collision-free property.

Because of these properties, a hash function is often used to check whether data has changed.


L bits
Message or data block M (variable length)
L
H
hash value h (fixed length)

Block Diagram of Cryptographic: Hash Function; h = H(M)

Working on Hashing Algorithms in Cryptography

Now that we have a basic idea of what a hash function is in cryptography, let's break down the internal mechanics.

The first act of the hashing algorithm is to divide the large input data into blocks of equal size. Further, the algorithm applies the hashing process to the data blocks one by one.

Though one block is hashed separately, all the blocks are related to each other. The output hash value for the first data block is taken as an input value and is summed up with the second data block. Similarly, the hashed output of the second block is summed up with the third block, and the summed-up input value is again hashed. And this process goes on and on until you get the final hash output, which is the summed-up value of all the blocks that were involved.

. However, since the hash values did not match, Bob was aware of the change. Now, he contacts Alice by phone and shares with her the information in the document he received. Alice confirms that her bank account is different than what is written in the document.

That's how a hashing function saves Alice and Bob from financial fraud. Now imagine this scenario with your own business and how it could.

Primary Terminologies

Preimage: Let’s say we have a hash value (hash value h = h(x)). We say that x is the first image of h. Let’s call x a data block, whose hash function (using the function H) is h. Because H is a multiple-fold mapping, there will always be some number of preimages for any hash value h.

Collision: If x≠y*x*□=*y* and H(x) = H(y), we’ll have a collision. Since we’re dealing with hash functions, it’s obvious that collisions are not desirable.

Popular Hash Functions

Hash functions play an important role in computing, providing versatile capabilities like: Quick retrieval of data, Secure protection of information (cryptography), Ensuring data remains unaltered (integrity verification). Some commonly used hash functions are

Message Digest 5 (MD5)

MD5 is a specific message digest algorithm, a type of cryptographic hash function. It takes an input of any length (a message) and produces a fixed-length (128-bit) hash value, which acts like a unique fingerprint for the message.

MD5 was widely used from the early 1990s onwards for various purposes, including:

- File Check: Making sure a file got from the web was not changed while transferring. MD5 was used to make a code for the first file and compare it to the code of the received file.
- Password Storage: MD5 was sometimes used to store passwords on servers. However, it was never recommended to store passwords directly in plain text. Instead, the password was hashed using MD5, and the hash value was stored. This meant that even if a security breach occurred, the actual passwords wouldn't be compromised.

Secure Hash Function (SHA)

SHA stands for Safe Hash Algorithm. It's a group of codes for keeping data safe made by NIST. These codes convert any size input into a fixed code, called a hash value or message digest.

There are different SHA types, each with varied lengths and security features:

- SHA-1: The first SHA code, making a 160-bit hash. It's now unsafe because of flaws and is no longer used.
- SHA-2: A family of improved SHA algorithms with various output lengths:
    - SHA-224 (224 bits)
    - SHA-256 (256 bits - most common)
    - SHA-384 (384 bits)
    - SHA-512 (512 bits)
- SHA-3: A completely redesigned hash function introduced after weaknesses were found in SHA-2. It offers improved security but isn't as widely used yet.

SHAs have a number of applications in digital security:

- Data Integrity: Checking if data is changed. Even small change means different hash value.
- E-Signatures: Verify documents. It uses private key, hash to sign data. Receiver checks signature using sender's public key, re-computed hash.
- Password Protection: Passwords are encrypted before saved. If there's a breach, only hash is compromised, not passwords.
- Software Check: Verify downloaded file is unchanged. Often, hash is given by distributor to check file's authenticity.

In the case of a digital signature, the hash value of a message is encrypted using the private key owned by the person sending it out so that no one else can change what they have said without being detected easily by those looking for it also within seconds by those scanning across different networks particularly corporate or government intranets. With this information at hand, hackers always attempt hacking passwords and other security codes so that torrent downloaders from this work freely without facing any restrictions they may not be able to avoid under lawful circumstances.

A hash code is used to provide a digital signature as:

a. The hash code is encrypted with public-key encryption using the sender's private key. This provides authentication, but it also provides a digital signature because only the sender can have produced the encrypted hash code. In fact, this is what the digital signature technique is all about.

b. If you want both confidentiality and a digital signature, then you can encrypt the message plus the private-key encrypted hash code using a symmetrical secret key. This is a common technique.

Create a single-pass password

One of the most common uses of hash functions is the creation of a single pass password file. A single pass password file is a scheme where the operating system holds the hash value of a user's password rather than the actual password itself.In other words, it's the hash value that the operating system stores. That's why a hacker can't get the real password from that file. When you type a password into your computer, the system uses the hash value to check if you typed the right password. Many systems use this method to protect passwords.

Intrusion detection and virus

Hash functions can also be used to detect intrusions and viruses. For each file on your system, store H(F) and keep the hash values safe (for example, on a protected CD-R). You can later check if a file is changed by recreating H(F). For example, an intruder would have to modify F without altering H(F).

Pseudorandom number generator (PRNG)

You can use a cryptographic hash function to create a PRF or a PRNG. One of the most common uses for a hash based PRF is to generate symmetric keys.

Security Requirements for Cryptographic Hash Functions

Requirement	**Description**
Variable input size	H can be applied to a block of data of any size.
Fixed output size	H produces a fixed-length output.
Efficiency	H(x) is relatively easy to compute for any given x, making both hardware and software implementations practical
Preimage resistant (one-way property)	For any given hash value , it is computationally infeasible to find u such that H(y) = h
Second preimage resistant (weak collision resistant)	For any given block x, it is computationally infeasible to find y≠xy□=*x* with H(y)=H(x) .
Collision resistant (strong collision resistant)	It is computationally infeasible to find any pair (x , y) such that H(x) = H(y).
Pseudorandomness	Output of H meets standard tests for pseudorandomness.

Key Management in Cryptography

In cryptography, it is a very monotonous 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. In this article, we will learn about key management, how Cryptographic Keys Work**,** Types of Key Management, and Key Management Lifecycle.

What is Key Management?

Key management refers to the processes and procedures involved in generating, storing, distributing, and managing cryptographic keys used in cryptographic algorithms to protect sensitive data. It ensures that keys used to protect sensitive data are kept safe from unauthorized access or loss. Good key management helps maintain the security of encrypted information and is important for protecting digital assets from cyber threats. Effective key management is crucial for ensuring the confidentiality, integrity, and availability of encrypted information by securing cryptographic keys from unauthorized access, loss, or compromise.

How Cryptographic Keys Works?

Cryptographic keys are special codes that protect information by locking (encrypting) and unlocking (decrypting) it. In **symmetric key cryptography**, a single shared key does both jobs, so the same key must be kept secret between users. In **asymmetric key cryptography**, there are two keys: a public key that anyone can use to encrypt messages or verify signatures, and a private key that only the owner uses to decrypt messages or create signatures. This makes it easier to share the public key openly while keeping the private key secret. These keys are crucial for secure communication, like when you visit a secure website (HTTPS), where they help encrypt your data and keep it safe from eavesdroppers and criminals. So, to manage these keys properly is vital to keep digital information secure and dependable.

Types of Key Management

There are two aspects of Key Management:

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

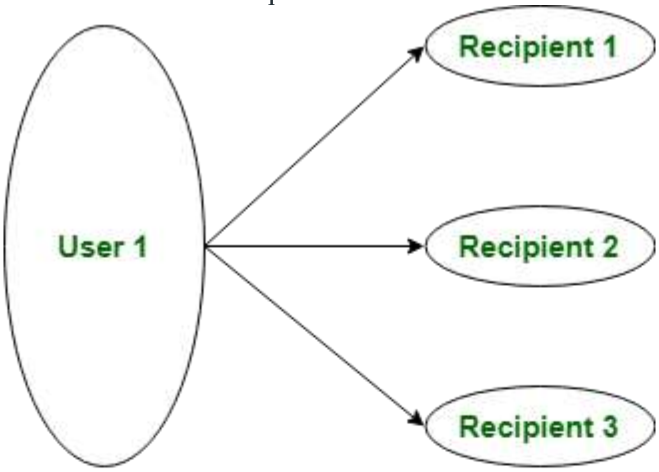
Distribution of Public Key

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 broadcast 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.

Key Management Lifecycle

The **key management lifecycle** outlines the stages through which cryptographic keys are generated, used, and eventually retired or destroyed. Proper management of these keys is critical to ensuring the security of cryptographic systems. Here's an overview of each stage:

1. Key Generation:

- **Creation**: Keys are created using secure algorithms to ensure randomness and strength.
- **Initialization**: Keys are initialized with specific parameters required for their intended use (e.g., length, algorithm).

2. Key Distribution:

- **Sharing**: For symmetric keys, secure methods must be used to share the key between parties.
- **Publication**: For asymmetric keys, the public key is shared openly, while the private key remains confidential.

3. Key Storage:

- **Protection**: Keys must be stored securely, typically in hardware security modules (HSMs) or encrypted key stores, to prevent unauthorized access.
- **Access Control**: Only authorized users or systems should be able to access keys.

4. Key Usage:

- **Application**: Keys are used for their intended cryptographic functions, such as encrypting/decrypting data or signing/verifying messages.
- **Monitoring**: Usage is monitored to detect any unusual or unauthorized activities.

. Key Rotation:

- **Updating**: Keys are periodically updated to reduce the risk of exposure or compromise.
- **Re-Keying**: New keys are generated and distributed, replacing old ones while ensuring continuity of service.

6. Key Revocation:

- **Invalidation**: Keys that are no longer secure or needed are invalidated.
- **Revocation Notices**: For public keys, revocation certificates or notices are distributed to inform others that the key should no longer be trusted.

7. Key Archival:

- **Storage**: Old keys are securely archived for future reference or compliance purposes.
- **Access Restrictions**: Archived keys are kept in a secure location with restricted access.

8. Key Destruction:

- **Erasure**: When keys are no longer needed, they are securely destroyed to prevent any possibility of recovery.
- **Verification**: The destruction process is verified to ensure that no copies remain.

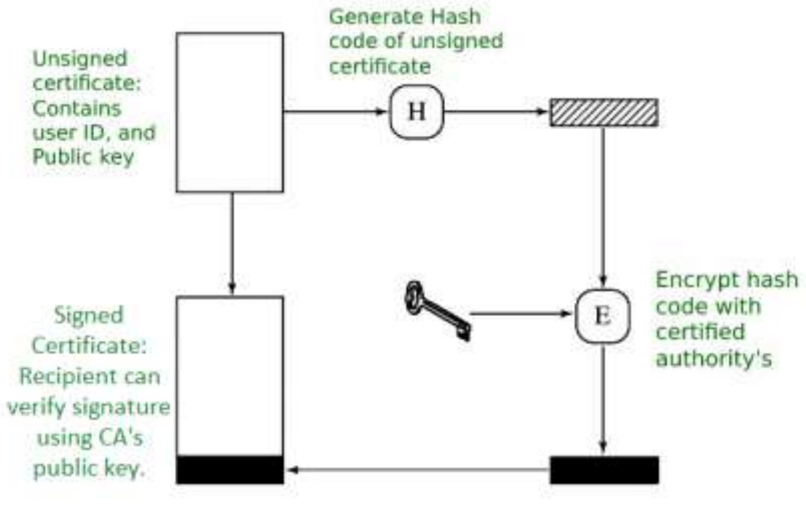
X.509 Authentication Service

X.509 is a digital certificate that is built on top of a widely trusted standard known as ITU or International Telecommunication Union X.509 standard, in which the format of PKI certificates is defined. X.509 digital certificate is a certificate-based authentication security framework that can be used for providing secure transaction processing and private information. These are primarily used for handling the security and identity in computer networking and internet-based communications.

Working of X.509 Authentication Service Certificate:

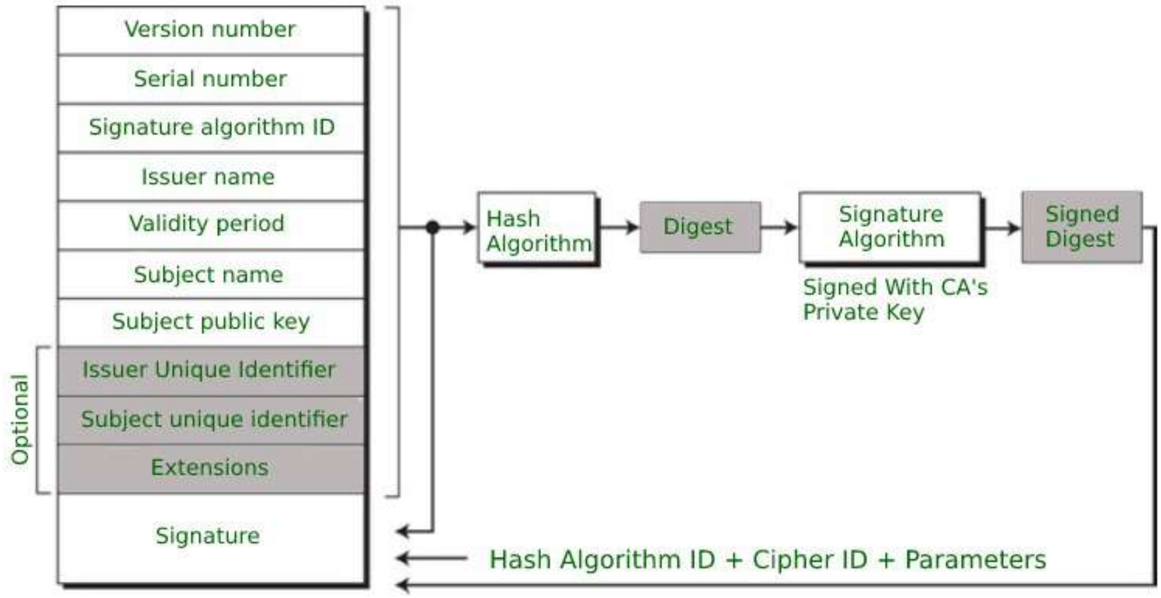
The core of the X.509 authentication service is the public key certificate connected to each user. These user certificates are assumed to be produced by some trusted certification authority and positioned in the directory by the user or the certified authority. These directory servers are only used for providing an effortless reachable location for all users so that they can acquire certificates. X.509 standard is built on an IDL known as ASN.1. With the help of Abstract Syntax Notation, the X.509 certificate format uses an associated public and private key pair for encrypting and decrypting a message.

Once an X.509 certificate is provided to a user by the certified authority, that certificate is attached to it like an identity card. The chances of someone stealing it or losing it are less, unlike other unsecured passwords. With the help of this analogy, it is easier to imagine how this authentication works: the certificate is basically presented like an identity at the resource that requires authentication.


Generate Hash code of unsigned certificate
Unsigned certificate: Contains user ID, and Public key
H
Encrypt hash code with certified authority's
E
Signed Certificate: Recipient can verify signature using CA's public key.

Public Key certificate use

Format of X.509 Authentication Service Certificate:



Generally, the certificate includes the elements given below:

- **Version number:** It defines the X.509 version that concerns the certificate.
- **Serial number:** It is the unique number that the certified authority issues.
- **Signature Algorithm Identifier:** This is the algorithm that is used for signing the certificate.
- **Issuer name:** Tells about the X.500 name of the certified authority which signed and created the certificate.
- **Period of Validity:** It defines the period for which the certificate is valid.
- **Subject Name:** Tells about the name of the user to whom this certificate has been issued.
- **Subject's public key information:** It defines the subject's public key along with an identifier of the algorithm for which this key is supposed to be used.
- **Extension block:** This field contains additional standard information.
- **Signature:** This field contains the hash code of all other fields which is encrypted by the certified authority private key.

Applications of X.509 Authentication Service Certificate:

Many protocols depend on X.509 and it has many applications, some of them are given below:

- Document signing and Digital signature
- Web server security with the help of Transport Layer Security (TLS)/Secure Sockets Layer (SSL) certificates
- Email certificates
- Code signing
- Secure Shell Protocol (SSH) keys
- Digital Identities

What Is OpenSSL?

OpenSSL is a cryptographic library that enables an open source implementation of Transport Layer Security (TLS) and Secure Sockets Layer (SSL) protocols. It provides functions to generate private keys, manage certificates, and equip client applications with encryption and decryption.

OpenSSL is widely used by software developers and system administrators to implement secure communication and encryption in various applications, such as web servers (like NGINX), email servers, VPNs, and more. It's available as a library that can be integrated into software applications or used as standalone command-line tools for various cryptographic operations.

OpenSSL Use Cases for Secure Application Delivery

OpenSSL can play a crucial role in a secure application delivery strategy, particularly when it comes to ensuring the confidentiality, integrity, and authenticity of data transmitted over networks.

It is commonly used for these use cases:

- Secure communication between clients and servers
- Data encryption
- Certificate management
- Data integrity and authentication
- Random number generation
- Secure protocols and algorithms
- Secure web server deployment

Advantages of OpenSSL

OpenSSL has been a widely used and established cryptographic library for many years. While there are alternatives available, OpenSSL has several advantages that have contributed to its popularity.

These advantages include:

- Maturity and wide adoption – OpenSSL has been in use for a long time and has been battle-tested in a wide range of applications. Its maturity and widespread adoption contribute to its reliability and stability. OpenSSL's implementation of SSL/TLS has been extensively used and reviewed, and it has been instrumental in securing internet communications.
- Community and documentation – OpenSSL has a large, active user and developer community, which results in better support, bug fixes, and security updates. Additionally, there are plenty of resources and documentation available for learning and troubleshooting.
- Feature rich – OpenSSL provides a comprehensive set of cryptographic functions, protocols, and tools, making it suitable for a wide variety of applications and use cases. In addition to SSL/TLS it also supports various encryption algorithms, key management, and certificate-related operations.
- Flexible – OpenSSL provides a wide range of options and configurations, allowing developers to tailor the library to their specific needs. It also provides bindings for various programming languages (including C, C++, Python, Java) and is available on many platforms and operating systems, making it versatile for cross-platform development.