UNIT III

**Message Authentication:**

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.

**SHA- 512 ALGORITHM:**

SHA-512 is a hashing algorithm that performs a hashing function on some data given to it.

Hashing algorithms are used in many things such as internet security, digital certificates and even blockchains. Since hashing algorithms play such a vital role in digital security and cryptography, this is an easy-to-understand walkthrough, with some basic and simple maths along with some diagrams, for a hashing algorithm called SHA-512. It's part of a group of hashing algorithms called SHA-2 which includes SHA-256 as well which is used in the bitcoin blockchain for hashing.

Before starting with an explanation of SHA-512, I think it would be useful to have a basic idea of what a hashing function's features are.

Hashing Algorithm — SHA-512

So, SHA-512 does its work in a few stages. These stages go as follows:

1. Input formatting
2. Hash buffer initialization
3. Message Processing
4. Output

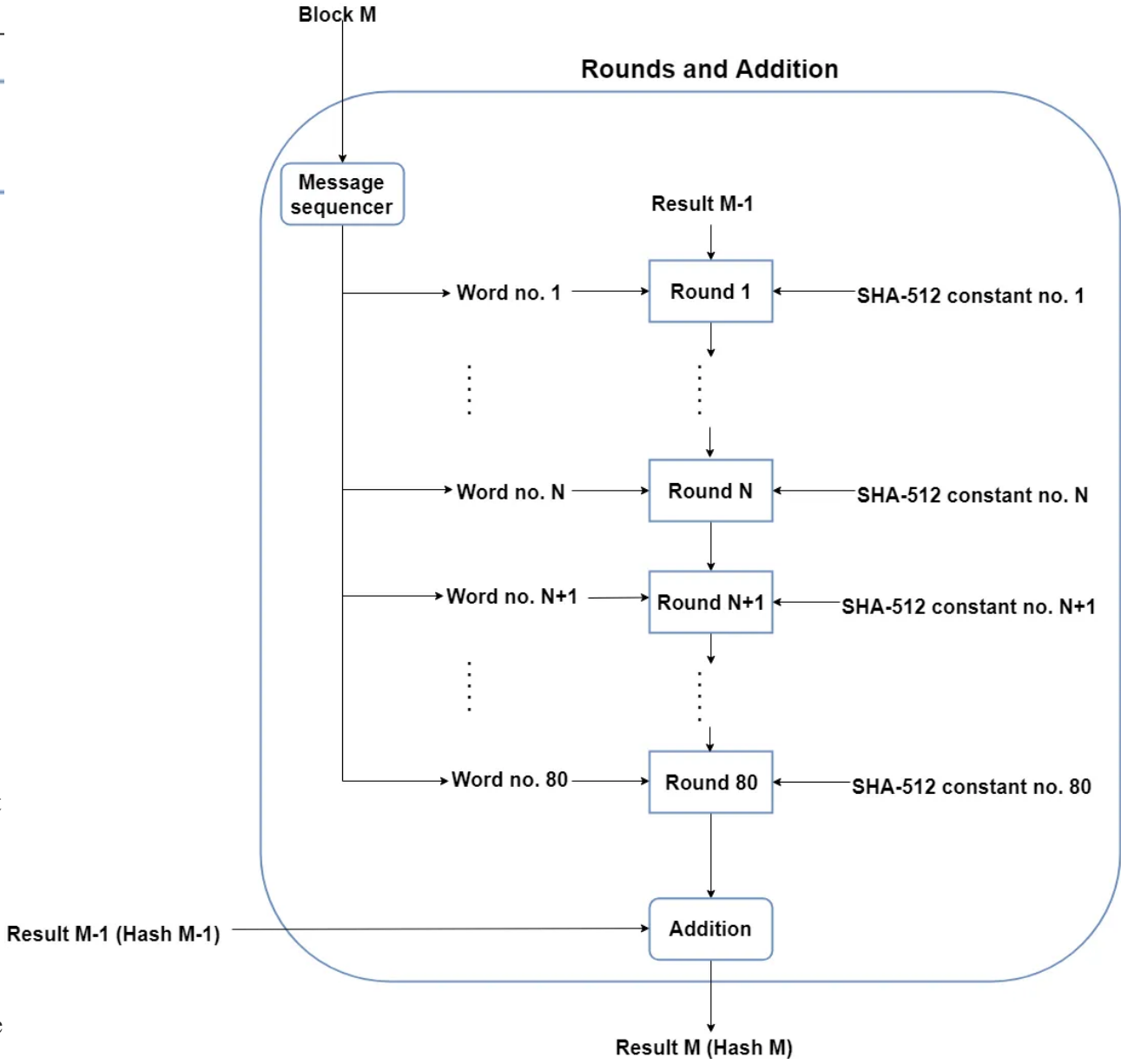
Let's look at these one-by-one.

1. **Input Formatting:**

SHA-512 can't actually hash a message input of any size, i.e. it has an input size limit. This limit is imposed by its very structure as you may see further on. The entire formatted mesage has basically three parts: the original message, padding bits, size of original message. And this should all have a

combined size of a whole multiple of 1024 bits. This is because the formatted message will be processed as blocks of 1024 bits each, so each bock should have 1024 bits to work with.

<pic: original message>

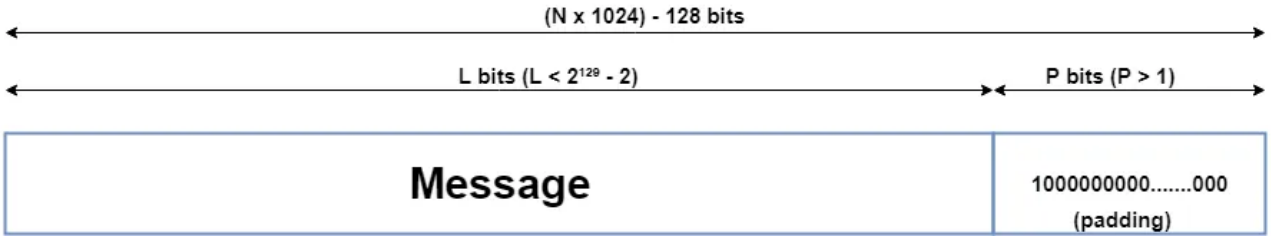



**Padding bits**

The input message is taken and some padding bits are appended to it in order to get it to the desired length. The bits that are used for padding are simply ‘0’ bits with a leading ‘1’ (100000…000). Also, according to the algorithm, padding *needs* to be done, even if it is by one bit. So a single padding bit would only be a ‘1’.

The total size should be equal to 128 bits short of a multiple of 1024 since the goal is to have the formatted message size as a multiple of 1024 bits (N x 1024).

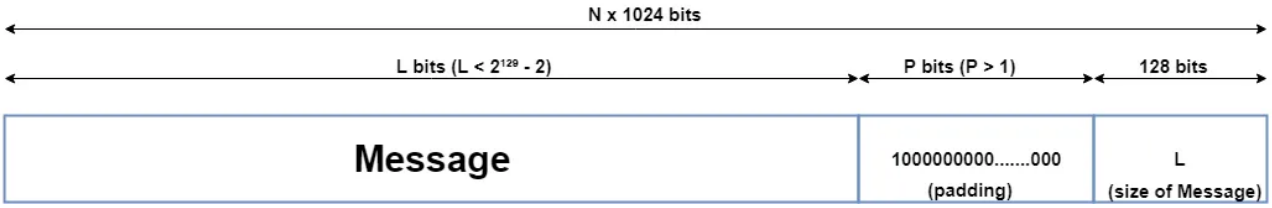
<pic: msg + pad>




**Padding size**

After this, the size of the original message given to the algorithm is appended. This size value needs to be represented in 128 bits and is the only reason that the SHA-512 has a limitation for its input message.

<pic: msg + pad +size>




Now that the padding bits and the size of the message have been appended, we are left with the completely formatted input for the SHA-512 algorithm.


**2. Hash buffer initialization:**

The algorithm works in a way where it processes each block of 1024 bits from the message using the result from the previous block. Now, this poses a problem for the first 1024 bit block which can't use the result from any previous processing. This problem can be solved by using a default value to be used for the first block in order to start off the process. (Have a look at the second-last diagram).

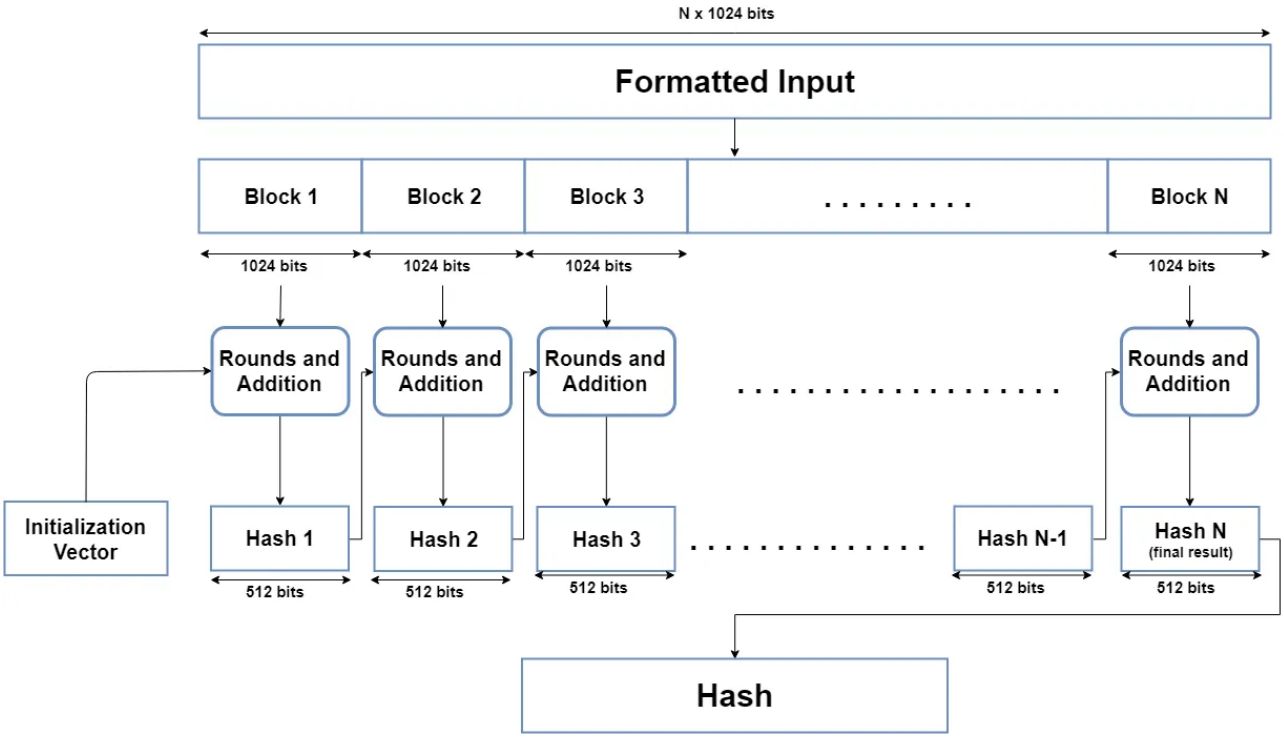
Since each intermediate result needs to be used in processing the next block, it needs to be stored somewhere for later use. This would be done by the *hash buffer*, this would also then hold the final hash digest of the entire processing phase of SHA-512 as the last of these 'intermediate' results.

So, the default values used for starting off the chain processing of each 1024 bit block are also stored into the hash buffer at the start of processing. The actual value used is of little consequence, but for those interested, the values used are obtained by taking the first 64 bits of the fractional parts of the square roots of the first 8 prime numbers (2,3,5,7,11,13,17,19). These values are called the Initial Vectors (IV).

### 3. Message Processing:

Message processing is done upon the formatted input by taking one block of 1024 bits at a time. The actual processing takes place by using two things: The 1024 bit block, and the result from the previous processing.

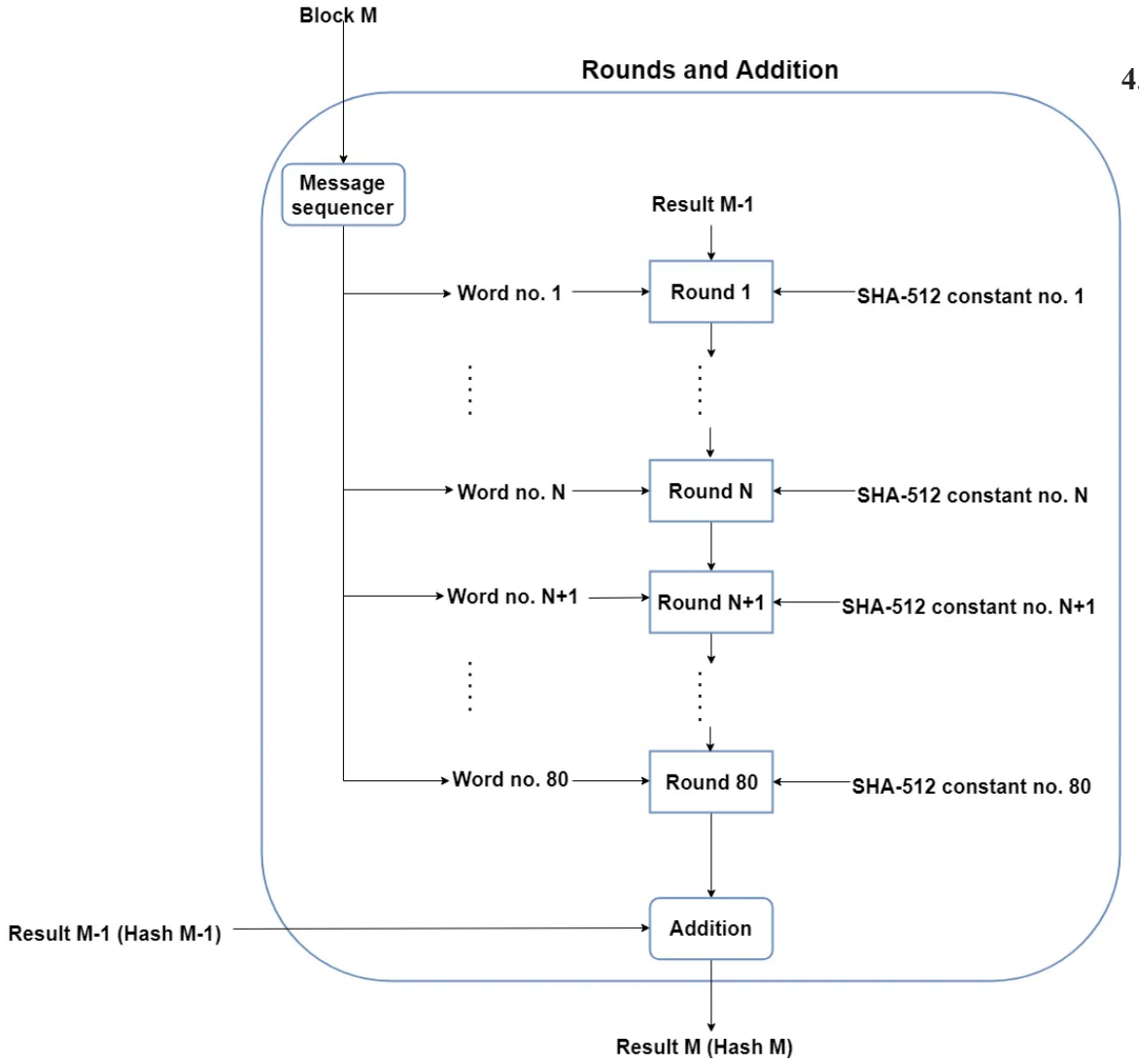
This part of the SHA-512 algorithm consists of several



So, the Message block (1024 bit) is expanded out into ‘Words’ using a ‘message sequencer’. Eighty Words to be precise, each of them having a size of 64 bits.

**Rounds**

The main part of the message processing phase may be considered to be the Rounds. Each round takes 3 things: one Word, the output of the previous Round, and a SHA-512 constant. The first Round doesn’t have a previous Round whose output it can use, so it uses the final output from the previous message processing phase for the previous block of 1024 bits. For the first Round of the first block (1024 bits) of the formatted input, the Initial Vector (IV) is used.




**4.**

**Output:**

After every block of 1024 bits goes through the message processing phase, i.e. the last iteration of the phase, we get the final 512 bit Hash value of our original message. So, the intermediate results are all used from each block for processing the next block. And when the final 1024 bit block has finished being processed, we have with us the final result of the SHA-512 algorithm for our original message.

---

**HMAC ALGORITHM:**

HMAC (Hash-based Message Authentication Code) is a type of 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 MACs, it is used for both data integrity and authentication.

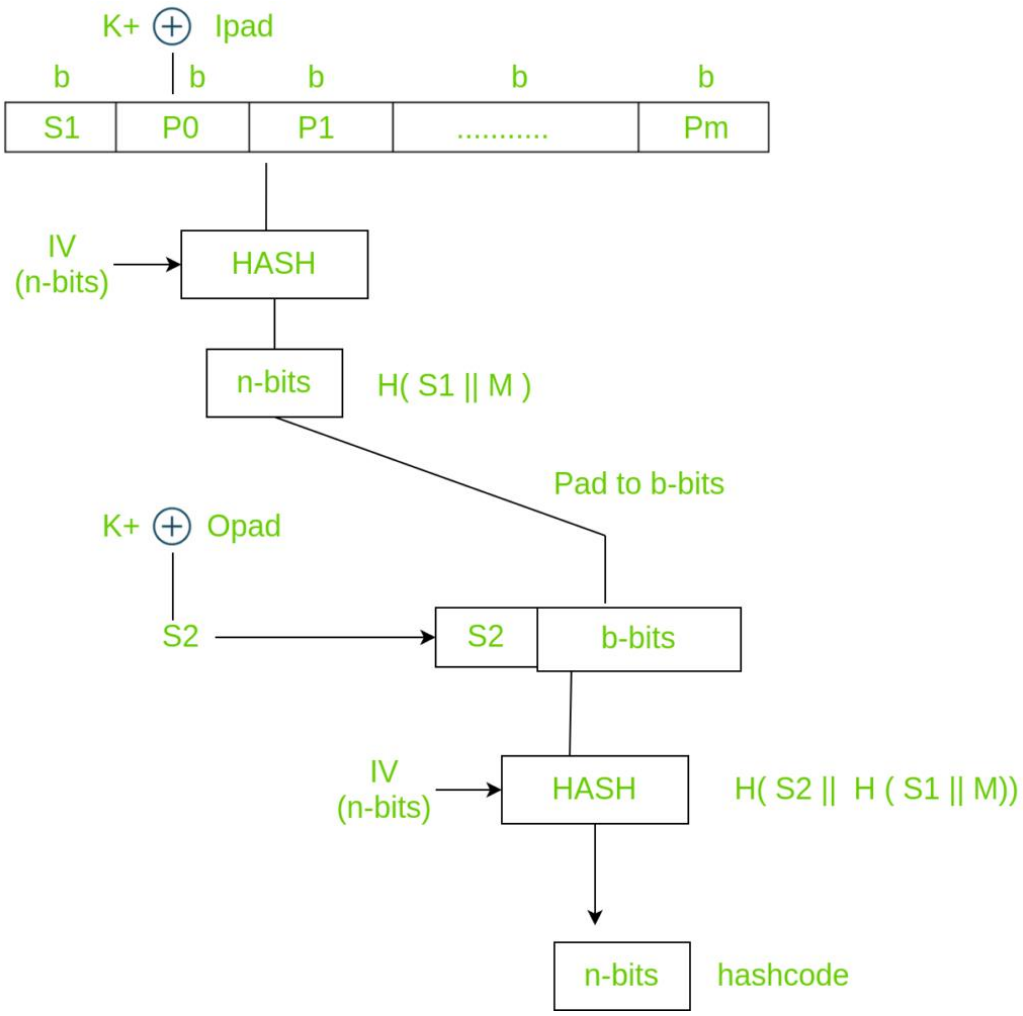
What is HMAC?

**HMAC (Hash-Based Message Authentication Code)** is a cryptographic technique that ensures data integrity and authenticity using a hash function and a secret key. Unlike approaches based on signatures and asymmetric cryptography. 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. HMAC uses a **symmetric key**(same copy) while Signatures uses an **asymmetric** (two different keys).

Working of Hash-based Message Authentication Code
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.




Digital Signature

A digital signature is a mathematical technique used to validate the authenticity and integrity of a message, software, or digital document.

1.**Key Generation Algorithms**: Digital signature is electronic signatures, which assure that the message was sent by a particular sender. While performing digital transactions authenticity and integrity should be assured, otherwise, the data can be altered or someone can also act as if he was the sender and expect a reply.

2.**Signing Algorithms**: To create a digital signature, signing algorithms like email programs create a one-way hash of the electronic data which is to be signed. The signing algorithm then encrypts the hash value using the private key (signature key). This encrypted hash along with other information like the hashing algorithm is the digital signature. This digital signature is appended with the data and sent to the verifier. The reason for encrypting the hash instead of the entire message or document is that a hash function converts any arbitrary input into a much shorter fixed-length value. This saves time as now instead of signing a long message a shorter hash value has to be signed and moreover hashing is much faster than signing.

3.**Signature Verification Algorithms** : Verifier receives Digital Signature along with the data. It then uses Verification algorithm to process on the digital signature and the public key (verification key) and generates some value. It also applies the same hash function on the received data and generates a hash value. If they both are equal, then the digital signature is valid else it is invalid.

The steps followed in creating digital signature are :

1.Message digest is computed by applying hash function on the message and then message digest is encrypted using private key of sender to form the digital signature. (digital signature = encryption (private key of sender, message digest) and message digest = message digest algorithm(message)).

2.Digital signature is then transmitted with the message.(message + digital signature is transmitted)

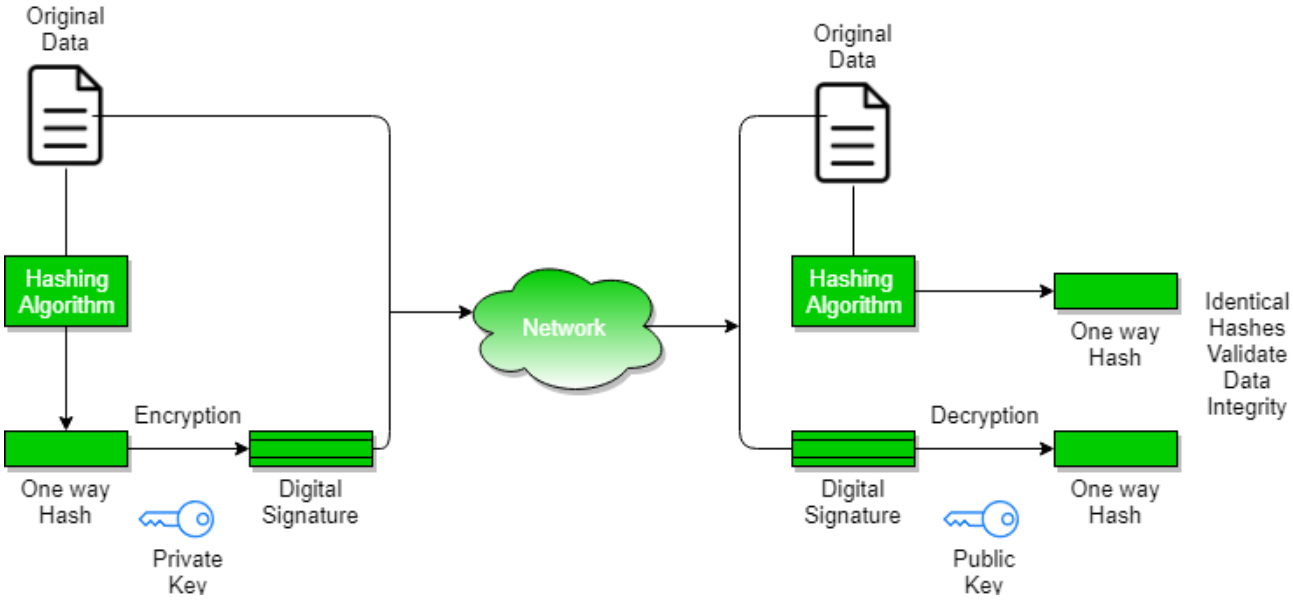
3.Receiver decrypts the digital signature using the public key of sender.(This assures authenticity, as only sender has his private key so only sender can encrypt using his private key which can thus be decrypted by sender's public key).

4.The receiver now has the message digest.

5.The receiver can compute the message digest from the message (actual message is sent with the digital signature).

6.The message digest computed by receiver and the message digest (got by decryption on digital signature) need to be same for ensuring integrity.

Message digest is computed using one-way hash function, i.e. a hash function in which computation of hash value of a message is easy but computation of the message from hash value of the message is very difficult.




Assurances about digital signatures

The definitions and words that follow illustrate the kind of assurances that digital signatures offer.

1.**Authenticity**: The identity of the signer is verified.

2.**Integration:** Since the content was digitally signed, it hasn't been altered or interfered with.

3.**Non-repudiation:** demonstrates the source of the signed content to all parties. The act of a signer denying any affiliation with the signed material is known as repudiation.

4.**Notarization:** Under some conditions, a signature in a Microsoft Word, Microsoft Excel, or Microsoft PowerPoint document that has been time-stamped by a secure time-stamp server is equivalent to a notarization.

Benefits of Digital Signatures

- **Legal documents and contracts:** Digital signatures are legally binding. This makes them ideal for any legal document that requires a signature authenticated by one or more parties and guarantees that the record has not been altered.
- **Sales contracts:** Digital signing of contracts and sales contracts authenticates the identity of the seller and the buyer, and both parties can be sure that the signatures are legally binding and that the terms of the agreement have not been changed.
- **Financial Documents:** Finance departments digitally sign invoices so customers can trust that the payment request is from the right seller, not from a bad actor trying to trick the buyer into sending payments to a fraudulent account.
- **Health Data:** In the healthcare industry, privacy is paramount for both patient records and research data. Digital signatures ensure that this confidential information was not modified when it was transmitted between the consenting parties.

## Kerberos

Kerberos provides a centralized authentication server whose function is to authenticate users to servers and servers to users. In Kerberos Authentication server and database is used for client authentication. Kerberos runs as a third-party trusted server known as the Key Distribution Center (KDC). Each user and service on the network is a principal.
The main components of Kerberos are:

- Authentication Server (AS):

  The Authentication Server performs the initial authentication and ticket for Ticket Granting Service.

- Database:

  The Authentication Server verifies the access rights of users in the database.

- Ticket Granting Server (TGS):

The Ticket Granting Server issues the ticket for the Server.

Step-1:
User login and request services on the host. Thus user requests for ticket-granting service.

- Step-2:

  Authentication Server verifies user's access right using database and then gives ticket-granting-ticket and session key. Results are encrypted using the Password of the user.

- Step-3:

  The decryption of the message is done using the password then send the ticket to Ticket Granting Server. The Ticket contains authenticators like user names and network addresses.

- Step-4:

  Ticket Granting Server decrypts the ticket sent by User and authenticator verifies the request then creates the ticket for requesting services from the Server.

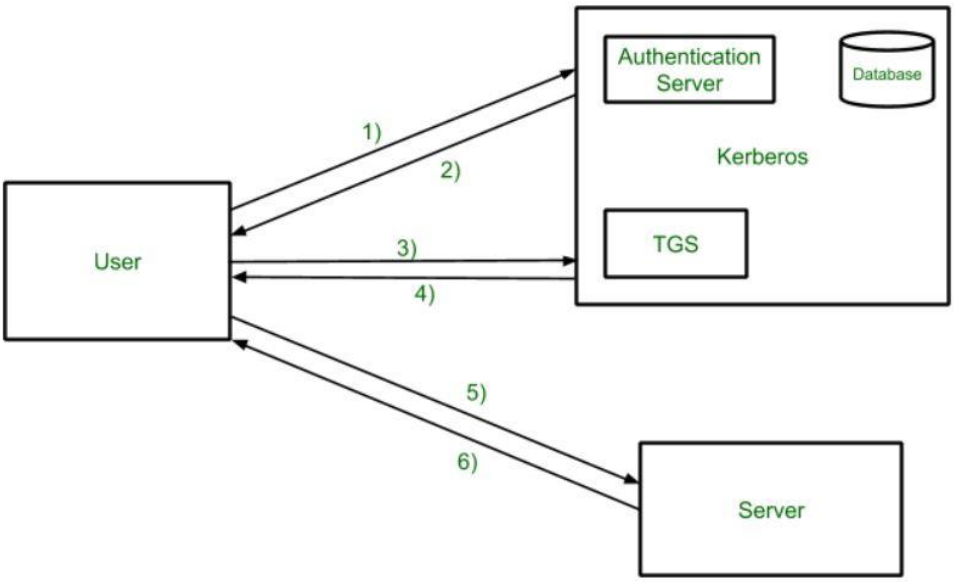
・Step-5:

The user sends the Ticket and Authenticator to the Server.

・Step-6:

The server verifies the Ticket and authenticators then generate access to the service. After this User can access the services.

X.509




Authentication Service

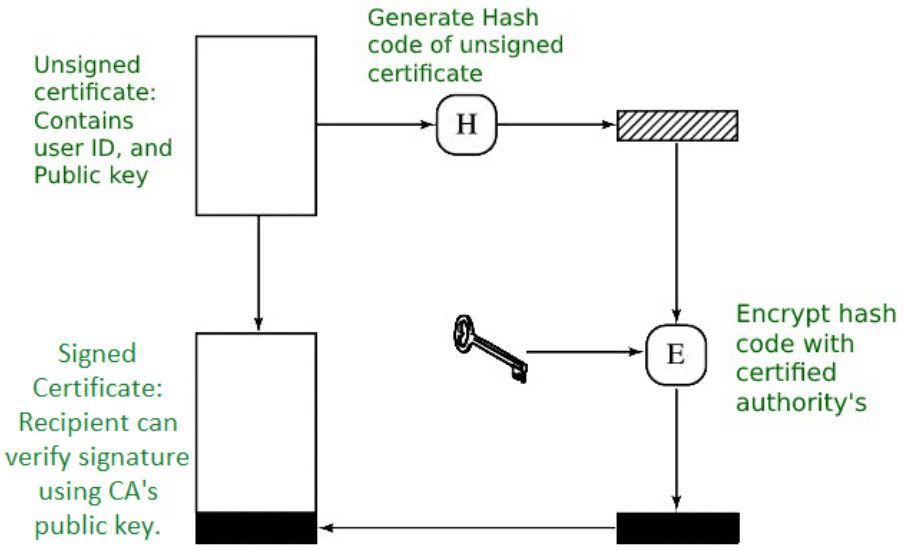
X.509 is a digital certificate that is built on top of a widely trusted standard known as ITU or International Telecommunication Union X.509 standard, in which the format of PKI certificates is defined. X.509 digital certificate is a certificate-based authentication security framework that can be used 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.




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

Public Key Infrastructure

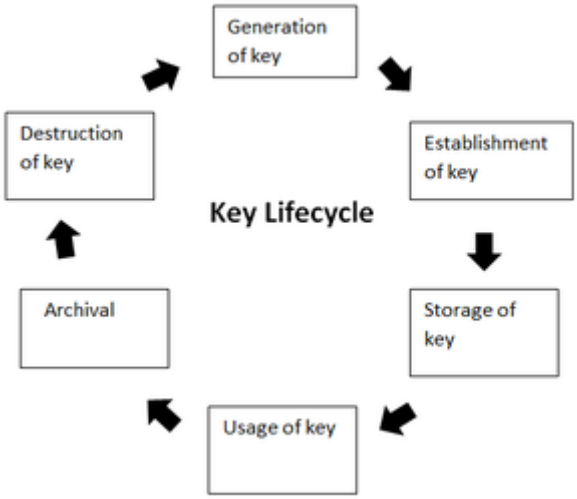
Public key infrastructure or PKI is the governing body behind issuing digital certificates. It helps to protect confidential data and gives unique identities to users and systems. Thus, it ensures security in communications.
The public key infrastructure uses a pair of keys: the public key and the private 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 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.




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 pWorking 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 AuthorWorking 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.ities: 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.

・Veification 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.erson 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.