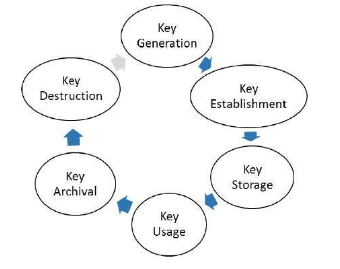
Key Life Cycle



1. Key Generation: This is the initial stage of the key life cycle, where the key is created using a secure random number generator or a key derivation function.
2. Key Establishment: Once the key is generated, it is securely shared or established between the parties who will use it for encryption, decryption, or digital signatures.
3. Key Storage: Keys must be securely stored to prevent unauthorized access, theft, or loss. The storage location must also be protected against physical and logical attacks.
4. Key Usage: The key is used for its intended purpose, such as encrypting and decrypting data or providing digital signatures.
5. Key Archival: Some systems may require that keys be archived for future use or to comply with legal or regulatory requirements.
6. Key Destruction: When a key is no longer needed or has reached the end of its useful life, it should be securely destroyed to prevent its potential reuse.

Requirements of Key Management

* Secrecy of private keys: Secret keys must remain secret from all parties except those who are owner and are authorized to use them.
* Assurance of public keys: The public keys are in open domain and seen as public pieces of data.

Public Key Infrastructure

Public Key Infrastructure (PKI) is a system used to securely manage the distribution and authentication of digital certificates, which are used to verify the identity of individuals or entities in online communications. PKI is based on the use of public key cryptography, where each individual or entity has a pair of keys - a public key and a private key.

PKI is commonly used in a variety of applications, including secure email, online banking, e-commerce, and secure remote access to networks. It helps to ensure the confidentiality, integrity, and authenticity of online transactions and communications, and is an important component of modern digital security.

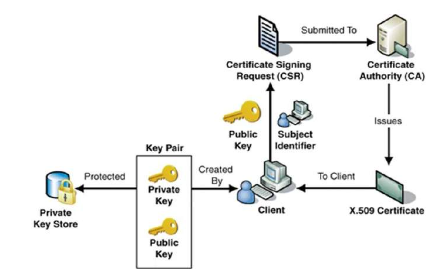
Digital Certificates

A digital certificate, also known as a public key certificate, is an electronic document that verifies the identity of a person, organization, or device in online communications. Digital certificates are issued by a trusted third-party organization, called a Certificate Authority (CA), which verifies the identity of the certificate holder and their ownership of a specific public key.

Digital certificates are based on the principles of public key cryptography, where each user has a public key and a private key. The public key is widely available, while the private key is kept secret. When a user requests a digital certificate, the CA verifies their identity and generates a certificate that includes the user's public key, along with other identifying information. The certificate is then digitally signed by the CA using its private key, creating a digital signature that can be used to verify the authenticity of the certificate.

Digital certificates are commonly used in a variety of applications, including secure email, online banking, e-commerce, and secure remote access to networks. When a user connects to a website secured by a digital certificate, their browser verifies the authenticity of the certificate by checking the digital signature and the identity of the CA that issued the certificate. If the certificate is deemed authentic, the browser establishes a secure, encrypted connection with the website.

Process of obtaining digital certificate



Key features of CA:  
Generating key pairs: − The CA may generate a key pair independently or jointly with the client.

Issuing digital certificates : − The CA issues a certificate after client provides the credentials to confirm his identity.

Publishing Certificates : CA need to publish certificates so that users can find them.

Verifying Certificates : The CA makes its public key available in environment to assist verification of his signature on clients’ digital certificate.

Revocation of Certificates : CA revokes the certificate issued due to some reason such as compromise of private key by user or loss of trust in the client.

After revocation, CA maintains the list of all revoked certificate that is available to the environment.

Classes of certificates

Class 1 − These certificates can be easily acquired by supplying an email address.

Class 2 − These certificates require additional personal information to be supplied.

Class 3 − These certificates can only be purchased after checks have been made about the requestor’s identity.

Class 4 − They may be used by governments and financial organizations needing very high levels of trust.

Public Key Cryptosystem

Each user generates a pair of keys to be used for the encryption and decryption of messages.

Each user places one of the two keys in a public register or other accessible file. This is the public key.

The companion key is kept private. This is Private Key.

Each user maintains a collection of public keys obtained from others.

With this approach, all participants have access to public keys

Private keys are generated locally by each participant and therefore need never be distributed.

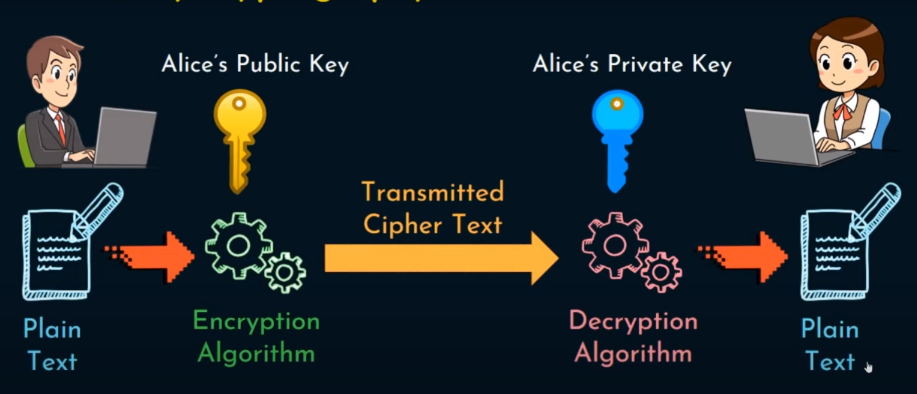
As long as a user’s private key remains protected and secret, incoming communication is secure.

At any time, a system can change its private key and publish the companion public key to replace its old public key.

In this case, the entire encrypted message serves as a digital signature.

It is impossible to alter the message without access to A’s private key, so the message is authenticated both in terms of source and in terms of data integrity.

This approach does not provide Confidentiality (because any observer can decrypt the message by using the sender’s public key).



Difference between symmetric and asymmetric encryption



Approach to attack RSA Algorithm

1. Brute force attack

Involves trying all possible private keys.

Defense against the brute-force approach is to use a large key space. Thus, the larger the number of bits in d.

key generation and encryption/decryption, are complex.

1. Mathematical attacks

Effort to factoring the product of two primes.

1. Timing attacks

These depend on the running time of the decryption algorithm.

1. Chosen ciphertext attacks

This type of attack exploits properties of the RSA algorithm. The larger the size of the key, the slower the system will run.

Possible Attacks during communication

1. Disclosure: Release of message contents to any person or process not possessing the appropriate cryptographic key.
2. Traffic analysis: Discovery of the pattern of traffic between parties. In a connection oriented application, the frequency and duration of connections could be determined. In either a connection-oriented or connectionless environment, the number and length of messages between parties could be determined.
3. Masquerade: Insertion of messages into the network from a fraudulent source. This includes the creation of messages by an opponent that are purported to come from an authorized entity. Also included are fraudulent acknowledgments of message receipt or nonreceipt by someone other than the message recipient.
4. Content Modification: Changes to the contents of a message, including insertion, deletion, transposition, or modification.
5. Sequence modification: Any modification to a sequence of messages between parties, including insertion, deletion, and reordering.
6. Timing modification: Delay or replay of messages. In a connection-orientated application, an entire session or sequence of messages could be a replay of some previous valid session, or individual messages in the sequence could be delayed or replayed.
7. Repudiation: Denial of receipt of message by destination or denial of transmission of message by source.

Message Authentication

Message authentication is a procedure to verify that received messages come from the alleged source and have not been altered. Message authentication may also verify sequencing and timeliness.

A digital signature is an authentication technique that also includes measures to counter repudiation by either source or destination.

Authentication Functions

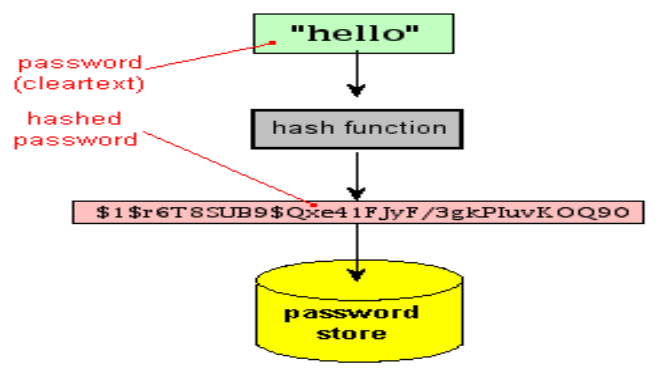
1. Message Encryption: The ciphertext of the entire message serves as its authenticator.
2. Message Authentication Code1 (MAC): A public function of the message and a secret key that produces a fixed length value that serves as the authenticator.
3. Hash Functions: A public function that maps a message of any length into a fixed length hash value, which serves as the authenticator.

Hash Function

A hash function H accepts a variable-length block of data M as input and produces a fixed-size hash value h = H(M) A change to any bit or bits results a change to the hash code.

Uses of hash function:

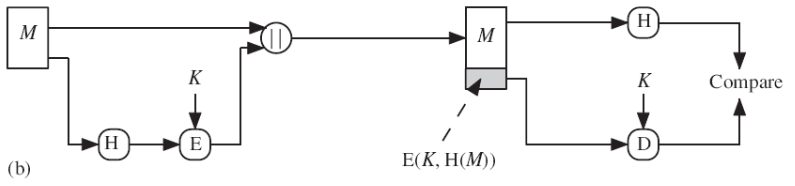
1. Verify the integrity
2. Hashing Password



1. Digitally signing documents

Weakness of hash – same hash generated for different inputs that may result in the suspect for digital signatures

Working of hash function



Digital Signature

A digital signature is a cryptographic technique used to ensure the authenticity, integrity, and non-repudiation of a digital document or message. It is similar to a handwritten signature in that it is used to verify the identity of the signer and to guarantee that the contents of the document have not been tampered with.

A digital signature is created using a combination of public-key cryptography and hashing. The sender of a document generates a unique digital signature by applying a mathematical algorithm to the document's contents using their private key. The receiver of the document can then verify the signature using the sender's public key and a matching algorithm, which confirms that the document has not been altered and that the sender is who they claim to be.

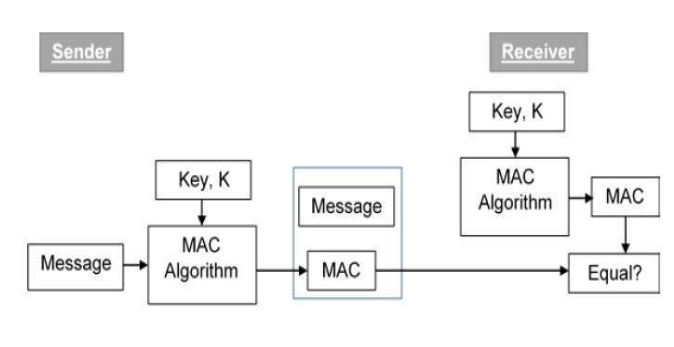
Digital signatures are commonly used in electronic transactions, such as e-commerce and online banking, to ensure the authenticity of electronic documents and to provide a secure way to exchange information over the Internet.

Message Authentication Code

A message authentication code (MAC) is a cryptographic technique used to ensure the authenticity and integrity of a message. It is similar to a digital signature, but it is used for verifying the authenticity of a message between two parties that share a secret key.

To create a MAC, a cryptographic algorithm takes the secret key and the message as input, and generates a fixed-length code that represents the message. This code is then appended to the message, and the result is transmitted to the recipient. The recipient can then use the same cryptographic algorithm, along with the shared secret key, to generate a new MAC for the received message. If the newly generated MAC matches the MAC that was transmitted with the message, then the recipient can be assured that the message has not been tampered with and that it came from the expected sender.

MACs are commonly used in situations where two parties share a secret key and need to exchange messages securely, such as in secure communication protocols like SSL/TLS, SSH, and IPsec. MACs provide a way to verify the integrity and authenticity of the message without the need for digital signatures or public key cryptography.



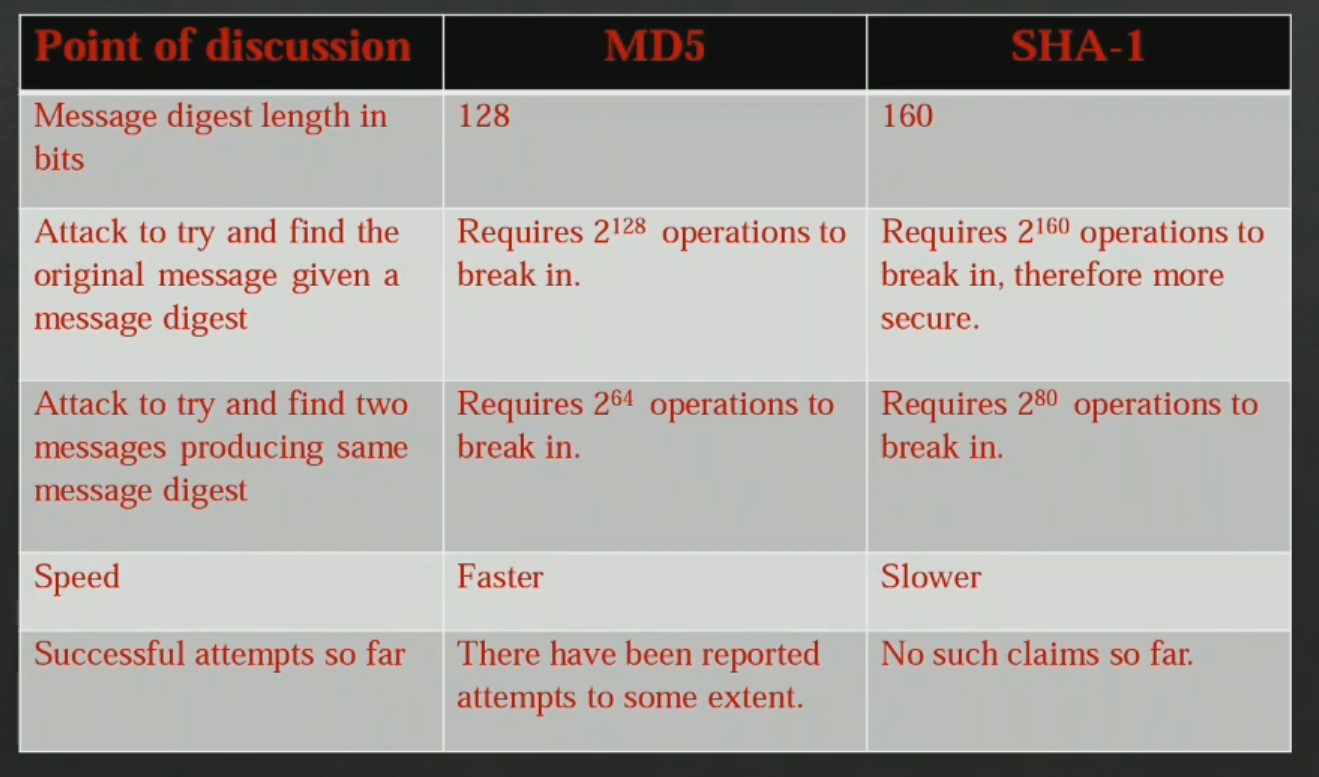
Limitations of MAC

1. Limited to Shared Key Cryptography: MACs are limited to situations where both parties have access to the same secret key. This means that they are not suitable for use in situations where the sender and receiver do not have a pre-existing shared secret.
2. Vulnerable to Key Compromise: If the secret key used to generate a MAC is compromised, then an attacker can generate valid MACs for any message, which would allow them to impersonate the sender and tamper with the message.
3. Limited to a Single Message: A MAC is only valid for a single message, and it cannot be used to verify the authenticity of multiple messages.
4. Not Suitable for Non-Repudiation: MACs are not suitable for providing non-repudiation, which is the ability to prove that a message was sent by a particular party and cannot be denied by that party.
5. Potential for Replay Attacks: If an attacker intercepts a valid message and its MAC, they can replay the message with the same MAC, which can result in the message being processed twice or cause other issues.

Features of Hash Function

1. Deterministic: A given input should always produce the same hash output. This property allows the hash function to be used for data integrity and digital signatures.
2. Quick Computation: The hash function should be computationally efficient and able to produce a hash quickly, even for large inputs.
3. Pre-image Resistance: Given a hash output, it should be difficult to find an input that produces that hash. This property ensures that the hash function is a one-way function and helps prevent brute force attacks.
4. Collision Resistance: It should be difficult to find two different inputs that produce the same hash output. This property helps prevent intentional or unintentional manipulation of data.
5. Fixed Output Length: The hash function should produce a fixed-length output, which enables the comparison of hash values and makes it possible to use them in digital signatures and message authentication codes.
6. Avalanche Effect: A small change to the input should produce a significant change in the hash output. This property ensures that a small change in the input will result in a significantly different hash output, making it difficult for an attacker to manipulate the input to produce a desired output.

Difference between SHA and MD5



Why SHA 512

More secure

Larger output size

Schnorr Digital Signature

It is based on the discrete logarithm problem and is considered to be a more efficient and secure alternative to other digital signature schemes such as the Digital Signature Algorithm (DSA) and the Elliptic Curve Digital Signature Algorithm (ECDSA).

In the Schnorr signature scheme, a signer generates a random secret value, computes a public key from that value, and then signs the message using a combination of the secret value and the public key. The resulting signature is a fixed-length value that can be verified by anyone who has access to the signer's public key.

One of the main advantages of the Schnorr signature scheme is that it supports batch verification, which means that multiple signatures can be verified simultaneously, resulting in improved efficiency and scalability. It also allows for more compact signatures, which can reduce storage and transmission costs.

Schnorr signatures have been adopted in several blockchain networks, including Bitcoin, as a way to enhance security and scalability. However, their use is still limited due to patent issues, which have only recently been resolved, and the need for widespread adoption of the scheme.

