Public-Key Cryptography

Table of Contents

1. Public Key Cryptography/ Asymmetric Cryptography

- Also known as two-key cryptography
- It can be used for confidentiality, authentication or both
- It has two keys:
    1. Public key: it's known by anyone. It's used for **message encryption and signature verifications.**
    2. Private key: it's known only by the owner. It's is used for **message decryption** and **signature creation(signing)**.
- *It's asymmetric because those who encrypt message or verify signature cannot decrypt messages or create signatures*.

1.1. Digital Signature

- A **digital signature** is a mathematical scheme for verifying the authenticity of digital messages or documents. A valid digital signature on a message gives a recipient that the message came from a sender known to the recipient.
- A digital signature gives the receiver a reason to believe the message was sent by the claimed sender.
- It employs asymmetric cryptography.

- They are equivalent to traditional handwritten signatures, but properly implemented digital signature are more difficult to forge than the handwritten type.
- Digital signature can also provide **non-repudiation**, meaning that the signer can not successfully claim they didn't sign a message, while also claiming their private key remains secret. *repudiation- rejection, refusal to acknowledge*
- A digital signature typically consists of three algorithms:
    1. **A key generation algorithm** that selects a *private key* uniformly at random from a set of possible private keys. The algorithm outputs the private key an corresponding *public key.*
    2. **A signing algorithm** given a message and private key produces *a signature*
    3. **Signature verifying algorithm** that given, the message,public key, and signature either accepts or rejects the message's claim to authenticity.
- Two main properties are required:
    1. The authenticity of the message can be verified using the corresponding public key
    2. It should be computationally infeasible to generate a valid signature for a party without knowing that party's private key.
- In 1976, Whitfield Diffie and Martin Hellman first described the notion of digital signature. Soon afterwards RSA algorithm was invented which could be used to produce primitive digital signatures.
- when used in digital signature, the private key is used to sign and public key to verify. This means that any one can verify the signature, but only the owner of the corresponding private key could have generated.
- When used in encryption, the public key is used to encrypt and private key to decrypt. Any one could encrypt a message, but only the owner of the private key can decrypt. However the size of the message they can encrypt is proportional to the size of the key, so they don't scale well for long messages.

1.2. Application of Public key Cryptography

- Are categorized in two 3:
    1. Encryption and decryption - to provide confidentiality.
    2. Digital Signature - to provide authentication.
    3. Key exchanges - to exchange session keys.
- Some algorithms are suitable for all uses, while others are specific to one of them.
- It is much secure than secret key cryptography though slow in processing. Thus, recommended to use for the last two applications. Why is it slower than secret key?

1.3. Key Distribution

- Users' public keys can be distributed using one of the following:
    1. Public announcements
        - Broadcast using emails, or related announcements
        - Vulnerable for forgery
    2. Public available directory
        - Better trusted with entry details, which contains {name, public-key}
        - With secure, periodically updating directory.
    3. Public-key authority

        - users know the public key of the directory.
    4. Certificate authority
        - In public key authority each time the user must appeal to the authority for the public key

1.4. RSA Cryptosystem

- By Rivest, Shamir, and Adleman of MIT 1977.
- It is best known and widely used public-key scheme.
- Uses large integers(e.g 1024 bits).
- Security due to cost of factoring large numbers.
- Plain text is encrypted in blocks.

1.5. Diffie-Hellman Cryptosystem

- It predates RSA and is in fact the oldest public key system still in use.
- It does neither encryption nor signature. It's used for shared secret key exchanges.
- Once secret key is exchanged using Diffie-Hellman, communicating parties can use secret key cryptography for communications.
- The limitation with this algorithm is that it doesn't have an *authenticating*.
- Security confidence of Diffie-Hellman is on the properties of the prime number.
    - The prime number shall be large number
    - It's also advisable to use a prime number, p with property of (p-1)/2 be another prime. e.g: a prime number 5 has (5-1)/2= 2 which is another prime number.

1.6. ElGamal Cryptosystem

- A variant of the Diffie-Hellman key distribution scheme, allowing secure exchange of messages.
- It's is based on discrete algorithm
- It's published in 1985 by ElGamal
- Like Diffie-Hellman its security depends on the difficulty of *factoring logarithms*
- ElGamal cryptosystem, called **Elliptic Curve Variant**, is based on the discrete logarithm problem.
- It doesn't use numbers modulo n.

2. Hashing Functions

- Hashing in cryptography is a concept that transforms data into fixed-length string of characters, known as a **hash value**.
- A hash function H accepts *a variable length block of data* as input and produces a *fixed-size hash value*. e.g: a hash function that generates 32-character hash value will consistently produce a unique 32 character code for any input.
- It's one way function

    - easy to compute
    - Hard to invert

  e.g: Phone book

- **Properties of hash function:**
    - Preimage resistance, given y, it's hard to find an x, such that h(x)=y.
    - Second preimage resistance
        - given x and y = h(x), it's hard to find x'!=x such that h(x)=h(x')
    - Collision resistance
        - It's hard to find any x, x' such that h(x)=h(x')
    - Avalanche effect
        - When an input changes slightly, the output changes significantly.
- **Use cases of hash functions:**
    - Hash table
    - Hash chains
    - Hash trees
    - Checksums
- **Famous Hash functions**
    - MD5
    - SHA

SHA vs MD5

Keys for comparison	MD5	SHA
Security	Less secure than SHA	High secure than MD5
Message digest length	128bits	160 bits
Attack required to find out original message	2^128 bit operations	2^160 bit operations
Attacks to try two messages producing the same MD	2^64 bit operations	2^80 bit operations
Speed	Faster, only 64 iterations	Slower, require 80 iterations
Successful attack so far	reported to some extent	no such attack yet

- **Application of cryptographic hash function**
    1. Message Authentication
    2. Digital signatures
    3. Other applications
        - One-way password file
        - Intrusion detection
        - Virus detection
        - Psudorandom generator
- Message authentication is a mechanism or service used to verify the integrity of a message. It assures that the data received are exactly as sent.(i.e contain no modification, insertion, deletion or replay).
- In message authentication, the hash function value is referred to as a **message digest**.
- Message authentication is concerned with:

    1. Protecting the *integrity* of the message
    2. Validating *identity* of the originator
    3. *non-repudiation* of the origin(dispute resolution)
- Three alternative functions used:
    1. Message encryption
    2. Message authentication code(MAC)
    3. Hash functions
- The following attacks can be protected using message authentication:
    1. Disclosure: Release of message contents
    2. Traffic analysis: discover the pattern of traffic between parties
    3. Masquerade: insertion of message into the network from a fraudulent source
    4. content, sequence, and timing modification
    5. source and destination repudiation. (*repudiation- denying, or not acknowledging it's yours*)

2.1. Message Authentication Code(MAC)

- is also known as keyed hash function
- MACs are used between two parties that share a secret key to authenticate information exchange between those parties.
- Generated by an algorithm that creates a small fixed-size block.
    - Depending on both message and some key.
    - Like encryption though need not be reversible.
- Appended to message as signature. Not that MAC is not a digital signature.
- Receiver performs same computation on message and checks it matches the MAC.
- Provides an assurance that the message is unaltered and comes from sender.

3. Authentication Systems

- Authentication- reliably verifying the identity of someone(something).
- Proof of identity by providing evidence of:
    - What you know: Secret password, challenge response
    - What you have: smart card, token
    - What you are: Finger print, retinal scan, smile..
    - Where you are: Geolocalization, which terminal
- Thus, ways to authenticate someone or something are:
    - Password-based authentication
    - Address-based authentication
    - Cryptographic authentication

3.1. Password-Based Authentication

- It's not who you know. It's what you know.
- Biggest problem with this authentication is eavesdropping.
- Online attack: about guessing password on the system
- Offline attack: guessing password from captured content derived from the password.

- Dictionary attack: because of source of good password guess is a dictionary, an offline password guessing is called dictionary attack.
- An alternative to storing unencrypted password is to store hashes of password. But combination is good, encrypt the password hashes.

3.2. Address-Based Authentication

- It's not what you know. It's where you are.
- Identifying the source can be inferred based on the network address from which packets arrive.(using ip address).
- Sometimes users can be authenticated based on their addresses:
    - Using GPS
    - Using relative reference of locations. E.g Gate 1

3.3. Cryptographic-Based Authentication

- It's not what you know or what you are. It's %sAPlkjhui7t72/y72okl771!@##8&
- This method is much secure than the others.

3.4. Who is being authenticated

- A user from any workstation against her own account on server
- Machine to machine for updates autonomously - backing up database
- Combination of users and machines - a bank teller can only use from the terminal that s/he is employed.
- The difference is machines can use cryptographic keys whereas humans use passwords.

3.5. Trusted Intermediaries

- Assume that network security is a secret key technology. A network is fairly large, thus computer in the world can't register every other computer's password.

3.5.1. Key Distribution Centers(KDC)

- KDC is a trusted node and KDC knows keys for all the other nodes.
- A new node installed on the network shall be added to KDC, thus communication among nodes can happen via the KDC.
- Drawbacks of KDC:
    - If it's compromised all the network resource is vulnerable
    - The KDC is *a single point of failure*. If it goes down, no access.
    - The KDC might be a *performance bottleneck.* Having multiple KDCs can alleviate this problem.

3.5.2. Certification Authorities(CA)

- Key distribution is easier with public key cryptography.
- CA generates certificates and signs it.
- <u>Pros</u>
    - The CA doesn't need to be online
    - If the CA were to *crash* the network wouldn't be disabled

    - Certificates are not *security-sensitive*
    - A compromised CA can not decrypt conversations.

3.5.3. Digital Certificates

- X.059[ISO97] has a defined format for a certificate A Certificate includes:
    - The user's name
    - The user's public key
    - An expiry time, Serial number
    - The issuing CA's signature on the entire content of the certificate.

3.5.4. Certificate Revocation

- If a certificate lost, compromise, or expired, it will be revoked.
    - won't be able to communicate with others
    - It's very similar to bank credit cards
- X.059 has defined **Certificate Revocation List (CRL)**.
    - A CRL lists serial number of certificates that should not be honored.
    - A new CRL is posted periodically, and lists revoked and unexpired certificates.
    - A certificate is valid if it has *a valid CA signature, has not expired, and is not listed in the CA's most recent CRL*

Author: Ertale81

Created: 2024-05-27 Mon 04:02