Cryptography Hash Functions

-

Cryptographic hash functions are mathematical algorithms that transform input data into a fixed-length sequence of characters, referred to as a hash value. Cryptographic hash functions are intended to be fast, deterministic, and one-way, meaning that even a minor change in input yields a very different hash. These functions are important for securing information digitally by allowing data verification and authentication.

- Cryptographic hash functions protect data integrity by creating identifying hash values, which enable systems to identify any unauthorized changes to messages or files in real time.
- Within cybersecurity, they are the foundation for digital signatures and certificate validation, giving a secure means to verify the authenticity of software and communications.
- They provide safe password storage through hashing passwords prior to saving, stopping direct exposure of sensitive credentials even if there happens to be a breach of data.
- Hash functions facilitate blockchain and other distributed ledgers by connecting blocks based on hash values to allow transparency and tamper-proof record-keeping.

Working of Cryptography Hash Function

- **Input Processing:** Cryptographic hash functions process an input of any length—whether text, file, or data stream—and subject it to a sequence of mathematical operations. The input can range from several bytes to gigabytes of information.
- **Fixed-Size Output Generation:** No matter what the length of the input, the function generates a fixed-size hash value, normally in the form of a hexadecimal string. This uniform output size provides equality regardless of the inputs.
- **Deterministic Operation:** The hash function consistently computes the same hash for the same input. Such a property enables uncompromising data authentication, as any alteration in the input leads to a totally unique hash.
- **Avalanche Effect:** A minor alteration in the input, even the flipping of one bit, significantly alters the resultant hash. The sensitivity ensures that collisions among hashes (two inputs having the same hash) are highly unlikely.
- **One-Way Computation:** The algorithm is made irreversible in the sense that it is computationally impossible to recover the original input from its hash value. This one-way feature protects sensitive information such as passwords and digital signatures.
- **Collision Resistance:** Hash functions used in cryptography are designed to minimize the probability of two distinct inputs generating the same hash value, upholding the integrity and trustworthiness of verification processes.

Properties of Cryptographic Hash Functions

- **Deterministic:** The same input always generates the exact same hash output, ensuring consistent and reliable verification of data.
- **Fast Computation:** Cryptographic hash functions are designed to process inputs quickly and efficiently, making them practical for handling large datasets and real-time applications.
- **Pre-image Resistance:** It is computationally infeasible to reverse-engineer or retrieve the original input data from its hash value, protecting sensitive information from exposure.
- **Second Pre-image Resistance:** Given an input and its hash, it is extremely difficult to find a different input that produces the same hash, preventing impersonation or forgery.
- **Collision Resistance:** The function minimizes the chance that two distinct inputs will produce identical hash values, ensuring unique data fingerprints for security and integrity.

- **Avalanche Effect:** Even a tiny change in the input, such as flipping a single bit, causes a significant and unpredictable change in the hash output, enhancing the function’s sensitivity to data modifications.

Applications of Cryptographic Hash Functions

Below are some applications of cryptography hash functions

Message Authentication

- Message authentication is a system or service that verifies the integrity of a communication.
- It ensures data is received precisely as transmitted, with no modifications, insertions, or deletions, a hash function is used for message authentication, and the value is sometimes referred to as a message digest.
- Message authentication often involves employing a message authentication code (MAC).
- MACs are widely used between two parties that share a secret key for authentication purposes. A MAC function uses a secret key and data block to generate a hash value, that identifies the protected communication.

Data Integrity Check

- Hash functions are most commonly used to create checksums for data files.
- This program offers the user with assurance that the data is correct.
- The integrity check allows the user to detect any modifications to the original file.
- It does not assure uniqueness. Instead of altering file data, the attacker can update the entire file, compute a new hash, and deliver it to the recipient.

Digital Signatures

- The digital signature application is comparable to message authentication.
- Digital signatures operate similarly to MACs.
- Digital signatures encrypt message hash values using a user's private key.
- The digital signature may be verified by anybody who knows the user's public key.

Popular Cryptographic Hash Algorithms

MD5 (Message Digest Algorithm 5)

Once widely used for data integrity and digital signatures, MD5 is now considered insecure due to vulnerabilities that allow attackers to generate hash collisions easily. Its speed and simplicity made it popular historically, but it is no longer recommended for security-critical applications.

SHA-1 (Secure Hash Algorithm 1)

SHA-1 improved upon MD5 with a longer hash length and better resistance to collisions. However, advances in computational power and cryptanalysis exposed weaknesses, leading to practical collision attacks. Consequently, SHA-1 is deprecated for most security uses, including SSL/TLS certificates and digital signatures.

SHA-2 Family (SHA-256, SHA-512)

The SHA-2 family is currently the industry standard for cryptographic hashing, offering robust security with longer hash outputs of 256 and 512 bits. These algorithms provide strong collision and pre-image resistance, making them the preferred choice for secure communication protocols, blockchain technologies, and password hashing.

SHA-3 (Keccak)

Adopted as the latest NIST standard, SHA-3 uses a unique sponge construction different from SHA-2, enhancing security and flexibility. It offers comparable hash lengths with improved resistance to certain types of attacks, making it suitable for applications demanding long-term security.

BLAKE2 & BLAKE3

Designed as high-speed, secure alternatives to SHA-2 and SHA-3, BLAKE2 and BLAKE3 deliver faster hashing without compromising security. BLAKE3, in particular, supports parallel processing and incremental updates, making it ideal for modern systems requiring both speed and strong cryptographic guarantees.

PROPERTIES OF CRYPTOGRAPHIC HASH FUNCTION

The news of NIST and their SHA-3 algorithm competition and a recent lunch and learn at Denim Group reminded me of the Cryptographic lectures I gave at UTSA. One of the hardest concepts my students had grasping was secure cryptographic hash functions, partially because of the number theory, but also in differentiating between the three properties of a secure hash function: collision resistance, preimage resistance, and second preimage resistance.

Preimage resistance: Given a hash value *h*, it should be hard to find any message *m* such that $h = hash(k, m)$.

This is the most usual property developers think of when they think of a cryptographic hash function. Unlike an encryption, there should be no "dehashing" function. A good preimage resistant function should be "hard" to invert. An example of a hash function that is not preimage resistant is $h = hash(k, m) = m \bmod 2^k$. For cryptographic hash functions, "hard" is, for a hash function with a range of size 2^k, it should take the adversary $2^k/2$ or 2^k-1 attempts before he or she finds the solution. Since this is very easy to invert, for any value *h*, a message of *m* can be found (basically, every message of the form $h + x * 2^k$, where x is an integer.

Collision resistance: Given two messages m_1 and m_2, it should be "hard" to find a hash such that $hash(k, m_1) = hash(k, m_2)$, where *k* is the hash key.

What this says is that given complete control over picking any messages you want, it should be "hard" to find two of them such that have the same hash value the same hash. An example of a hash function that is ***not*** collision resistant is $hash(k, m) = 4$, since all hashes result in 4, making it 100% likely that two messages will have the same hash. This is the hardest standard for a hash function to attain; collision resistant hash functions are in fact a superset of second preimage resistant ones (if you've found a second preimage of a hash function, you have necessarily found a collision).

Second preimage resistance: Given a message *m1*, it should be hard to find a different message m_2 such that $hash(k, m_1) = hash(k, m_2)$.

This is the toughest of the three for students in cryptography to get their head around. On the surface, this seems to be an easier version of collision resistance, because it seems that we have extra information to use. We only need one more message, rather than two. In actuality, this is the birthday paradox in action—second preimage resistance is actually a much stronger property than collision resistance.

Of course, it's important to reiterate that hashes are "digests", and not "encryption" algorithms. Encryption is a two-way operation. That is, given a message m, an encryption algorithm enc and its corresponding decryption algorithm *dec*, $dec(enc(m)) = m$. Hash functions are different—there is (ideally) no way to "decrypt" a hashed message, per preimage resistance. Because of this, it is common to say that hash functions are one-way operations.

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.

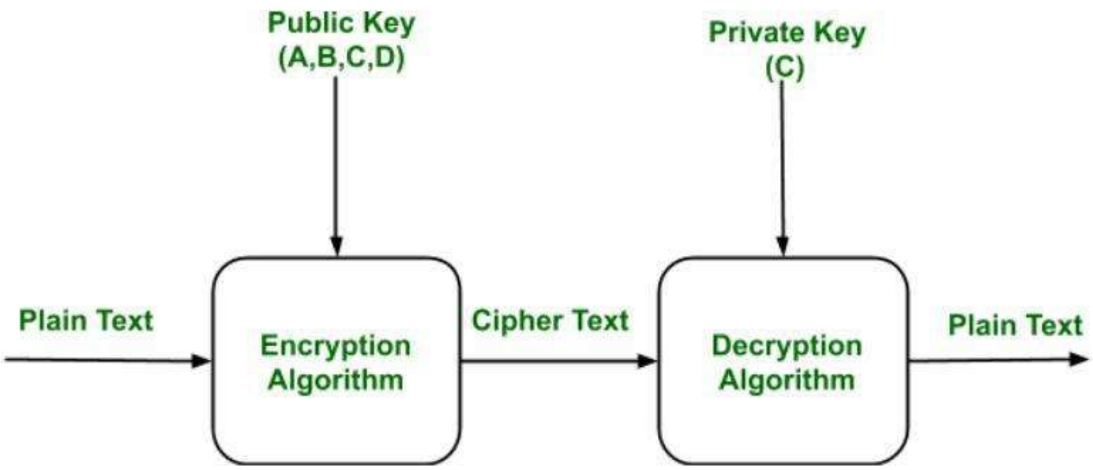
Public Key Cryptography

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.

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:

- **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.
- **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.

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.

Public Key Encryption Working

Key Pair Generation : A user generates a pair of keys :

- **Public Key**: Shared openly. Anyone can use it to send an encrypted message.
- **Private Key**: Kept secret. Only the key owner can decrypt messages encrypted with the public key.

Encryption : If someone wants to send a private message:

- They obtain the recipient’s public key.
- They encrypt the message using that public key.
- The encrypted message is sent over a network.

Decryption : Upon receiving the message:

- The recipient uses their private key to decrypt the message and recover the original plaintext

Public Key Encryption

Public Key Encryption Practical Example: Secure Website (HTTPS)

When you visit a secure website like `https://www.bank.com`, public key encryption is used behind the scenes to encrypt data between your browser and the bank's server.

Bank's Server Has a Key Pair

- **Private Key**: Secret, stored securely on the server.
- **Public Key**: Shared with anyone via an SSL certificate.

You Connect to the Website

- Your browser gets the bank’s public key from its SSL certificate.
- It verifies the certificate is valid (issued by a trusted certificate authority).

Encrypting the Session Key

- Your browser creates a random symmetric key (used for actual data encryption).
- It encrypts this key using the bank's public key.
- Only the bank can decrypt it using its private key.

Secure Communication Begins

- Now both your browser and the bank share a secret symmetric key.
- All further communication (login info, account data, etc.) is encrypted using this key.

Why Public Key Encryption is Used

- It ensures that only the server (with the private key) can read the symmetric key.
- Even if someone intercepts the traffic, they can’t decrypt the session key or data.

Characteristics of Public Encryption key

- **Security Assurance**: It is computationally infeasible to determine the private (decryption) key from the public (encryption) key and algorithm alone.
- **Key Pair Flexibility**: Either key (public or private) can be used for encryption, with the other used for decryption supporting both confidentiality and authentication.
- **Easy Public Key Distribution**: Public keys can be shared freely, enabling convenient encryption and digital signature verification.
- **Private Key Confidentiality**: Private keys are kept secret, ensuring that only the key owner can decrypt content or create valid digital signatures.
- **Foundation of RSA**: The most widely used public-key cryptosystem, **RSA**, is based on the difficulty of factoring large composite numbers into primes.

Limitations of the Public Key Encryption

- **Susceptible to Brute-Force Attacks**: Although computationally hard, public key encryption algorithms can be theoretically brute forced if key lengths are too short or computational power advances (e.g., quantum computing).
- **Private Key Loss**: If a user loses their private key, they can no longer decrypt data or prove their identity, making the system non-recoverable and highly vulnerable.
- **Man-in-the-Middle (MitM) Attack Risk**: A third party could intercept and alter public keys during transmission, leading to unauthorized decryption or spoofed signatures if keys aren't verified through a trusted channel.
- **PKI Chain of Trust Vulnerability**: If a private key higher in the PKI hierarchy (e.g., a root certificate authority) is compromised, it can invalidate all subordinate certificates, enabling widespread MitM attacks.

Applications of the Public Key Encryption

- **SSL/TLS protocols** : They use public key encryption to securely exchange symmetric session keys between a web browser and a server.
- **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.
- **SSH keys** : For secure login to remote servers use public/private key pairs for authentication.
- **Blockchain and Cryptocurrencies**: Users control wallets with private keys , public keys serve as wallet addresses.

Digital Signatures and Certificates

-
-

Digital signatures and certificates are two key technologies that play an important role in ensuring the security and authenticity of online activities. They are essential for activities such as online banking, secure email communication, software distribution, and electronic document signing. By providing mechanisms for authentication, integrity, and non-repudiation, these technologies help protect against fraud, data breaches, and unauthorized access.

Experience the ease of obtaining legally binding signatures online, all while maintaining the highest standards of security and compliance with the leading e-signature platform, SignNow. It is a secure and efficient electronic signature solution designed to streamline your document signing process while ensuring top-tier security features.

Digital Signature

A digital signature is a mathematical technique used to validate the authenticity and integrity of a message, software, or digital document. These are some of the key features of it.

- **Key Generation Algorithms**: Digital signatures are 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 were the sender and expect a reply.
- **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 hashing is much faster than signing.
- **Signature Verification Algorithms**: The Verifier receives a Digital Signature along with the data. It then uses a Verification algorithm to process 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.

How Digital Signature Works

The steps followed in creating a digital signature are:

1. Message digest is computed by applying the hash function on the message and then message digest is encrypted using the private key of the sender to form the digital signature. (digital signature = encryption (private key of sender, message digest) and message digest = message digest algorithm (message)).
2. A digital signature is then transmitted with the message. (message + digital signature is transmitted)
3. The receiver decrypts the digital signature using the public key of the sender. (This assures authenticity, as only the sender has his private key so only the sender can encrypt using his private key which can thus be decrypted by the 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.

Digital Signature vs. Electronic Signature

A digital signature is a specific type of electronic signature that uses cryptographic techniques such as public and private key pairs to verify the authenticity and integrity of a message or document. On the other hand, an electronic signature is a broader term including any electronic method that signifies agreement such as typing a name, clicking a button or scanning a handwritten signature. While it may not offer the same level of security or authentication as a digital signature, it is commonly used for non-sensitive transactions and agreements.

Assurances About Digital Signatures

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

- **Authenticity**: The identity of the signer is verified.
- **Integration:** Since the content was digitally signed, it hasn't been altered or interfered with.
- **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.
- **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 attacker 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.

Drawbacks of Digital Signature

- **Dependency on technology:** Because digital signatures rely on technology, they are susceptible to crimes, including hacking. As a result, businesses that use digital signatures must make sure their systems are safe and have the most recent security patches and upgrades installed.
- **Complexity:** Setting up and using digital signatures can be challenging, especially for those who are unfamiliar with the technology. This may result in blunders and errors that reduce the system's efficacy. The process of issuing digital signatures to senior citizens can occasionally be challenging.
- **Limited acceptance:** Digital signatures take time to replace manual ones since technology is not widely available in India, a developing nation.

Digital Certificate

Digital certificate is issued by a trusted third party which proves sender's identity to the receiver and receiver's identity to the sender. A digital certificate is a certificate issued by a Certificate Authority (CA) to verify the identity of the certificate holder. Digital certificate is used to attach public key with a particular individual or an entity.

Digital Certificate Contains

- Name of certificate holder.
- Serial number which is used to uniquely identify a certificate, the individual or the entity identified by the certificate
- Expiration dates.
- Copy of certificate holder's public key. (used for decrypting messages and digital signatures)
- Digital Signature of the certificate issuing authority.

Digital certificate is also sent with the digital signature and the message.

Advantages of Digital Certificate

- **Network Security:** A complete layered strategy is required by modern cybersecurity methods, wherein many solutions cooperate to offer the highest level of protection against attackers. An essential component of this puzzle is digital certificates, which offer strong defense against manipulation and man-in-the-middle attacks.
- **Verification:** Digital certificates facilitate cybersecurity by restricting access to sensitive data, which makes authentication a crucial component of cybersecurity. Thus, there is a decreased chance that attackers will cause disturbance. At many different endpoints, certificate-based authentication provides a dependable method of identity verification. Compared to other popular authentication methods like biometrics or one-time passwords, certificates are more flexible.
- **Buyer Success:** Consumers demand complete assurance that the websites they visit are reliable. Because digital certificates are supported by certificate authority that users' browsers trust, they offer a readily identifiable indicator of reliability.

Disadvantages of Digital Certificate

- **Phishing Attacks:** To make their websites look authentic, attackers can fabricate bogus websites and obtain certificates. Users may be fooled into providing sensitive information, such as their login credentials, which the attacker may then take advantage of.
- **Weak Encryption:** Older digital certificate systems may employ less secure encryption methods that are open to intrusions.
- **Misconfiguration:** In order for digital certificates to work, they need to be set up correctly. Websites and online interactions can be attacked due to incorrectly configured certificates.

Digital Certificate vs Digital Signature

Digital signature is used to verify authenticity, integrity, non-repudiation, i.e. it is assuring that the message is sent by the known user and not modified, while digital certificate is used to verify the identity of the user, maybe sender or receiver. Thus digital signature and certificate are different kind of things but both are used for security. Most websites use digital certificate to enhance trust of their users.

Feature	Digital Signature	Digital Certificate
Basics / Definition	A digital signature secures the integrity of a digital document in a similar way as a fingerprint or attachment.	Digital certificate is a file that ensures holder's identity and provides security.
Process / Steps	Hashed value of original data is encrypted using sender's private key to generate the digital signature.	It is generated by CA (Certifying Authority) that involves four steps: Key Generation, Registration, Verification, Creation.
Security Services	**Authenticity** of Sender, **integrity** of the document and **non-repudiation**.	It provides security and **authenticity** of certificate holder.
Standard	It follows Digital Signature Standard (DSS).	It follows X.509 Standard Format

Encryption: Process of converting electronic data into another form, called ciphertext, which cannot be easily understood by anyone except the authorized parties. This assures data security.
Decryption: Process of translating code to data.

- The message is encrypted at the sender's side using various encryption algorithms and decrypted at the receiver's end with the help of the decryption algorithms.
- When some message is to be kept secure like username, password, etc., encryption and decryption techniques are used to assure data security.

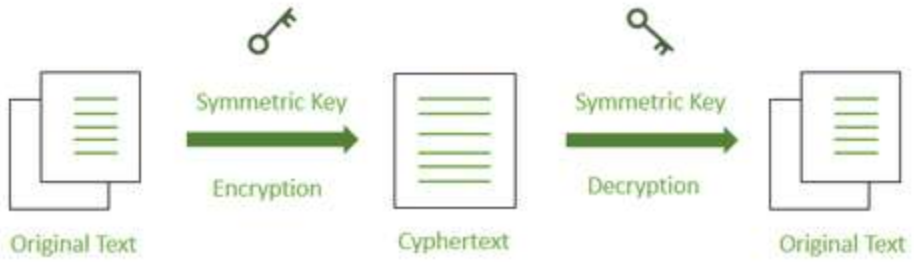
Types of Encryption

Data encryption transforms information into a code that is only accessible to those with a password or secret key, sometimes referred to as a decryption key. Data that has not been encrypted is referred to as plaintext, whereas data that has been encrypted is referred to as ciphertext. In today's business sector, encryption is one of the most popular and effective data protection solutions. By converting data into ciphertext, which can only be decoded with a special decryption key generated either before or at the time of the encryption, data encryption serves to protect the secrecy of data.

- **Symmetric Encryption**
  Data is encrypted using a key and the decryption is also done using the same key. There are a few strategies used in cryptography algorithms. For encryption and decryption processes, some algorithms employ a unique key. In such operations, the unique key must be secured

since the system or person who knows the key has complete authentication to decode the message for reading.

Symmetric Encryption



- **Asymmetric Encryption**
  Asymmetric Cryptography is also known as public-key cryptography. It uses public and private keys for the encryption and decryption of message. One key in the pair which can be shared with everyone is called the public key. The other key in the pair which is kept secret and is only known by the owner is called the private key.

Asymmetric Encryption

1. Public key: Key which is known to everyone. Ex-public key of A is 7, this information is known to everyone.
2. Private key: Key which is only known to the person who's private key it is.
3. Authentication: Authentication is any process by which a system verifies the identity of a user who wishes to access it.
4. Non-repudiation: Non-repudiation is a way to guarantee that the sender of a message cannot later deny having sent the message and that the recipient cannot deny having received the message.
5. Integrity: To ensure that the message was not altered during the transmission.
6. Message digest: The representation of text in the form of a single string of digits, created using a formula called a one way hash function. Encrypting a message digest with a private key creates a digital signature which is an electronic means of authentication.