**An Overview of Cloud Cryptography**

Cloud cryptography is a set of techniques used to secure data stored and processed in cloud computing environments. It provides data privacy, data integrity, and data confidentiality by using encryption and secure key management systems. Common methods used in cloud cryptography include:

1. Symmetric encryption: encrypts and decrypts data using the same key.
2. Asymmetric encryption: uses two different keys, a public key for encryption and a private key for decryption.
3. Hash functions: create a unique digest of a message to ensure its integrity.
4. Key management: securely stores and manages encryption keys to ensure the security of encrypted data.

The use of cryptography in the cloud is essential for protecting sensitive information and ensuring compliance with regulations such as GDPR and HIPAA.

**Cloud Cryptography** is encryption that safeguards data stored within the cloud. Several measures are being placed within cloud cryptography which adds a strong layer of protection to secure data to avoid being breached, hacked or affected by malware. Any data hosted by cloud providers are secured with encryption, permitting users to access shared cloud services securely and conveniently. Cloud Cryptography secures sensitive data without delaying the delivery of information.

**How does cryptography in the cloud work?**

Cloud cryptography is based on encryption, in which computers and algorithms are utilized to scramble text into ciphertext. This ciphertext can then be converted into plaintext through an encryption key, by decoding it with a series of bits. The encryption of data can take place in one of the following ways:

1. **Pre-encrypted data which is synced with the cloud:**There is software accessible to pre-encrypt it before information gets to the cloud, making it impossible to read for anyone who tries to hack it.
2. **End-to-end encryption:**Senders and receivers send messages, whereby they are the only ones who can read them.
3. **File encryption:**File encryption occurs when at rest, data is encrypted so that if an unauthorized person tries to intercept a file, they will not be able to access the data it holds.
4. **Full disk encryption:**When any files are saved on an external drive, they will be automatically encrypted. This is the key method to secure hard drives on computers.

**How the data on the cloud be secured by Cryptography?**

Cloud cryptography brings the same level of security to cloud services by securing data stored with encryption. It can protect sensitive cloud data without delaying data transmission. Many organizations define various cryptographic protocols for their cloud computing to keep a balance between security and efficiency. The cryptography algorithms used for Cloud Security are:

1. **Symmetric Key Cryptographic Algorithm:**This algorithm gives authentication and authorization to the data because data encrypted with a single unique key cannot be decrypted with any other key. [Data Encryption Standard (DES)](https://www.geeksforgeeks.org/data-encryption-standard-des-set-1/), Triple Data Encryption Standard (3DES), [Advanced Encryption Standard (AES)](https://www.geeksforgeeks.org/aes-full-form/) are the most popular Symmetric-key Algorithms which are usedin cloud computing for cryptography.
2. **Asymmetric Key Cryptographic Algorithm:**This algorithm is using two separate different keys for the encryption and decryption process in order to protect the data on the cloud. The algorithms used for cloud computing are [Digital Signature Algorithm (DSA)](https://www.geeksforgeeks.org/difference-between-rsa-algorithm-and-dsa/), [RSA](https://www.geeksforgeeks.org/rsa-algorithm-cryptography/) and [Diffie-Helman Algorithm](https://www.geeksforgeeks.org/implementation-diffie-hellman-algorithm/).
3. **Hashing:**Unlike other methods, hashing is one way process. A key is hashed using algorithms like sha256 or sha512. A hash of a key mark it’s uniqueness. Hashing is extensively used to store password, digital signatures and any sensitive information which must not be shared with others

**Advantages of Cloud Cryptography**

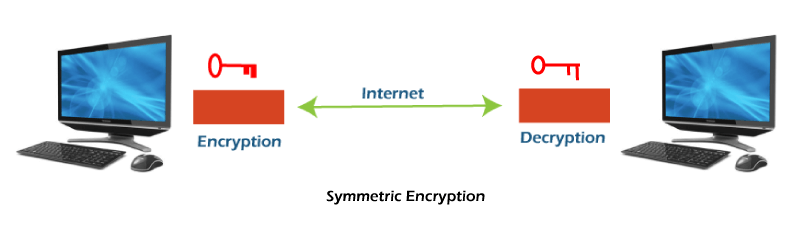
* The data remains private for the users. This reduces cybercrime from hackers.
* Organization receive notifications immediately if an unauthorized person tries to make modifications. The users who have cryptographic keys are granted access.
* The encryption prevents the data from being vulnerable when the data is being brought over from one computer to another,
* Cloud encryption permits organizations to be proactive in their defence against data breaches and cyberattacks and have become a necessity in today’s data-driven world.
* Receivers of the data have the ability to identify if the data received is corrupted, permitting an immediate response and solution to the attack.
* Encryption is one of the safest methods to store and transfer the data as it complies with the restrictions imposed by organizations such as FIPS, FISMA, HIPAA or PCI/DSS.

**Disadvantages of Cloud Cryptography**

* Cloud cryptography only grants limited security to the data which is already in transit.
* It needs highly advanced systems to maintain encrypted data.
* The systems must be scalable enough to upgrade which adds to the involved expenses.
* Overprotective measures can create difficulties for organizations when recovering data.

Symmetric encryption

Symmetric encryption encrypts and decrypts the information using a single password. In this encryption technique, the message is encrypted with a key, and the same key is used for decrypting the message. It is the simplest and commonly known encryption technique. It makes it easy to use but less secure.

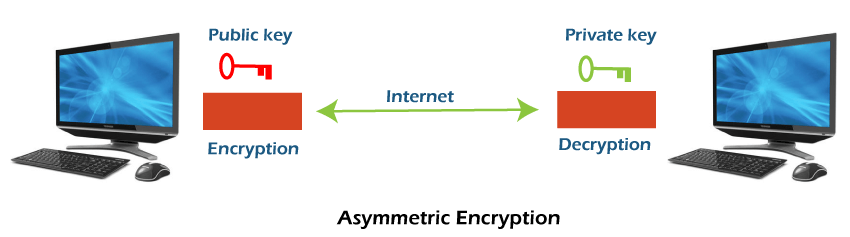


It is called symmetric encryption because the same key is responsible for encrypting or decrypting the data. The single key used in symmetric encryption is used to encrypt plain text into ciphertext, and that same key is used to decrypt that ciphertext back into plain text.

Symmetric encryption is also called secret key encryption. The algorithm behind the symmetric encryption executes faster and less complex, so it is the preferred technique to transmit the data in bulk.

Asymmetric encryption

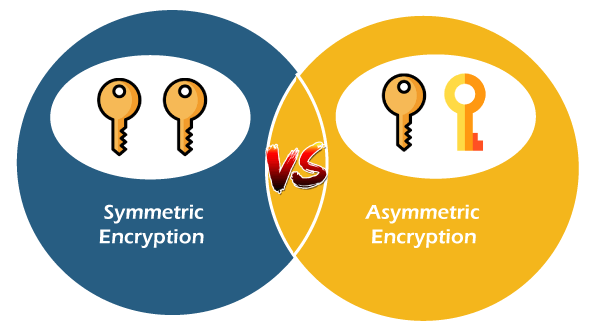
Asymmetric encryption uses two keys for encryption and decryption. It is based on the technique of public and private keys. A public key, which is interchanged between more than one user. Data is decrypted by a private key, which is not exchanged. It is slower but more secure. The public key used in this encryption technique is available to everyone, but the private key used in it is not disclosed.



The drawback of this encryption is that it takes more time than the symmetric encryption process. Asymmetric encryption is slower than secret-key encryption because, in secret key encryption, a single shared key is used to encrypt and decrypt the message, while in public-key encryption, two different keys are used, both related to each other by a complex mathematical process. Therefore, we can say that encryption and decryption take more time in public-key encryption.

In asymmetric encryption, a message that is encrypted using a public key can be decrypted by a private key, while if the message is encrypted by a private key can be decrypted by using the public key. Asymmetric encryption is widely used in day-to-day communication channels, especially on the internet.

Symmetric encryption v/s Asymmetric encryption



**Digital Signatures and Certificates**

**Digital signatures** and **certificates** are two key technologies that play a crucial 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*](https://signnow.sjv.io/baEbJm)*. It is a secure and efficient electronic signature solution designed to streamline your document signing process while ensuring top-tier security features.*

**Digital Signature**

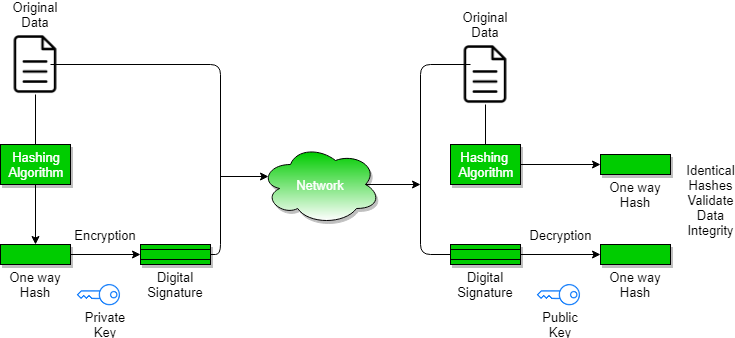
A digital signature is a mathematical technique used to validate the authenticity and integrity of a message, software, or digital document. These are some of the key features of it.

1. **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.
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 hashing is much faster than signing.
3. **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.

**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](https://www.geeksforgeeks.org/what-is-one-way-function-in-cryptography/#:~:text=A%20one%2Dway%20hash%20function,to%20as%20a%20hash%20value.), 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 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](https://www.geeksforgeeks.org/what-is-hacking-definition-types-identification-safety/). 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.

**Cryptography Hash Functions**

Hash functions in cryptography are extremely valuable and are found in practically every information security application. A hash function transforms one numerical input value into another compressed numerical value. It is also a process that turns plaintext data of any size into a unique ciphertext of a predetermined length.

**What is Cryptography Hash Function?**

A cryptographic hash function (CHF) is an equation that is widely used to verify the validity of data. It has many applications, particularly in information security (e.g. user authentication). A CHF translates data of various lengths of the message into a fixed-size numerical string the hash. A cryptographic hash function is a single-directional work, making it extremely difficult to reverse to recreate the information used to make it.

**How Does a Cryptography Hash Function Work?**

* The hash function accepts data of a fixed length. The data block size varies between algorithms.
* If the blocks are too small, padding may be used to fill the space. However, regardless of the kind of hashing used, the output, or hash value, always has the same set length.
* The hash function is then applied as many times as the number of data blocks.

**What Does a Cryptography Hash Function Do?**

A hash function in cryptography takes a plaintext input and produces a hashed value output of a particular size that cannot be reversed. However, from a high-level viewpoint, they do more.

* **Secure against unauthorized alterations:**It assists you in even minor changes to a message that will result in the generation of a whole new hash value.
* **Protect passwords and operate at various speeds:** Many websites allow you to save your passwords so that you don't have to remember them each time you log in. However, keeping plaintext passwords on a public-facing server is risky since it exposes the information to thieves. Websites commonly use hash passwords to create hash values, which they then store.

**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](https://www.geeksforgeeks.org/message-authentication-requirements/) 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](https://www.geeksforgeeks.org/mac-address-in-computer-network/).
* Digital signatures encrypt message hash values using a user's [private key](https://www.geeksforgeeks.org/blockchain-public-key-cryptography/).
* The digital signature may be verified by anybody who knows the user's [public key](https://www.geeksforgeeks.org/public-key-encryption/).

**Stream Ciphers**

In stream cipher, one byte is encrypted at a time while in block cipher ~128 bits are encrypted at a time. Initially, a key(k) will be supplied as input to pseudorandom bit generator and then it produces a random 8-bit output which is treated as keystream. The resulted keystream will be of size 1 byte, i.e., 8 bits. Stream ciphers are fast because they encrypt data bit by bit or byte by byte, which makes them efficient for encrypting large amounts of data quickly. Stream ciphers work well for real-time communication, such as video streaming or online gaming, because they can encrypt and decrypt data as it’s being transmitted.

**Key Points of Stream Cipher**

1. Stream Cipher follows the sequence of pseudorandom number stream.
2. One of the benefits of following stream cipher is to make cryptanalysis more difficult, so the number of bits chosen in the Keystream must be long in order to make cryptanalysis more difficult.
3. By making the key more longer it is also safe against brute force attacks.
4. The longer the key the stronger security is achieved, preventing any attack.
5. Keystream can be designed more efficiently by including more number of 1s and 0s, for making cryptanalysis more difficult.
6. Considerable benefit of a stream cipher is, it requires few lines of code compared to block cipher.

**Encryption**



For Encryption,

* Plain Text and Keystream produces Cipher Text (Same keystream will be used for decryption.).
* The Plaintext will undergo XOR operation with keystream bit-by-bit and produces the Cipher Text.

**Example:**

*Plain Text : 10011001*

*Keystream : 11000011*

*““““““““““`*

*Cipher Text : 01011010*

**Decryption**

For Decryption,

* Cipher Text and Keystream gives the original Plain Text (Same keystream will be used for encryption.).
* The Ciphertext will undergo XOR operation with keystream bit-by-bit and produces the actual Plain Text.

**Example:**

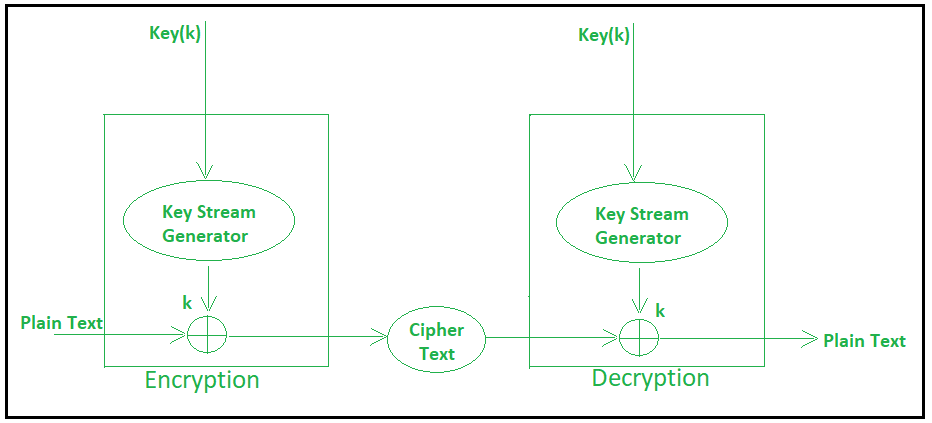
*Cipher Text : 01011010*

*Keystream : 11000011*

*“““““““““““*

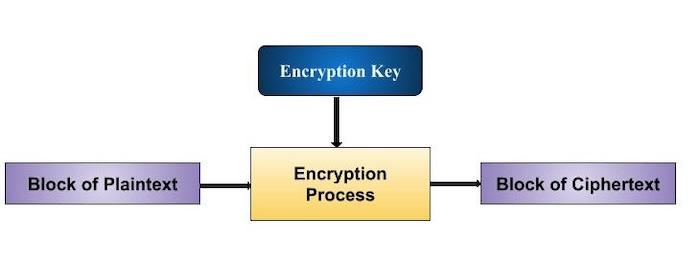
*Plain Text : 10011001*

Decryption is just the reverse process of Encryption i.e. performing XOR with Cipher Text.



**Block Cipher**

The basic scheme of a block cipher is depicted as follows −



A block cipher takes a block of plaintext bits and generates a block of ciphertext bits, generally of same size. The size of block is fixed in the given scheme. The choice of block size does not directly affect to the strength of encryption scheme. The strength of cipher depends up on the key length.

Block Size

Though any size of block is acceptable, following aspects are borne in mind while selecting a size of a block.

* **Avoid very small block size** − Say a block size is m bits. Then the possible plaintext bits combinations are then 2m. If the attacker discovers the plain text blocks corresponding to some previously sent ciphertext blocks, then the attacker can launch a type of ‘dictionary attack’ by building up a dictionary of plaintext/ciphertext pairs sent using that encryption key. A larger block size makes attack harder as the dictionary needs to be larger.
* **Do not have very large block size** − With very large block size, the cipher becomes inefficient to operate. Such plaintexts will need to be padded before being encrypted.
* **Multiples of 8 bit** − A preferred block size is a multiple of 8 as it is easy for implementation as most computer processor handle data in multiple of 8 bits.