Encryption Mechanism

By Naveen PS

MAKERDEMY
LEARN EXPLORE MAKE

Symmetric and Asymmetric Encryption

Computer encryption is based on the science of cryptography, which has been used as long as humans have wanted to keep information secret. Most forms of cryptography in use nowadays rely on computers, simply because a human-based code is too easy for a computer to crack. Cryptosystems use a set of procedures known as cryptographic algorithms, or ciphers, to encrypt plain text messages into cipher text or encrypted messages or decrypt cipher text messages into plain text.

Auguste Kerckhoff in 1883 stated that encryption algorithms should be made public and the "keys" be kept secret, which is Kerckhoff's Principle. Computer encryption systems generally belong in one of two categories: symmetric encryption and asymmetric or public-key encryption.

In symmetric encryption, the sender and receiver use a separate instance of the same key to encrypt and decrypt messages. Symmetric encryption heavily relies on the fact that the keys must be kept secret. Distributing the key in a secure way is one of the primary challenges of symmetric encryption, which is known as the "key distribution problem." The key that is the vital component in symmetric cryptography and we cannot afford to lose it or misplace it. If the individual keys are misplaced, the message can be decrypted by malicious actors.

The main advantage of symmetric cryptography is that it is much faster than asymmetric cryptography. The most important disadvantages of symmetric encryption are the key distribution problem and the key management problem. When the number of connected users grows, so does the number of required keys. Management of an increasing number of secret keys becomes "key management problem." Further, symmetric cryptography ensures only the 'confidentiality' of the transmitted or stored data. It cannot be used to ensure integrity and/or authenticity.

When connecting to a website on the public internet it becomes more complicated and symmetric encryption, by itself, won't work because you don't control the other end of the connection. How do you share a secret key with each other without the risk of someone on the internet intercepting it in the middle? In November 1976, a paper published in the journal IEEE Transactions

on Information Theory by Diffie and Hellman, titled "New Directions in Cryptography," addressed this problem and offered up a solution: public-key encryption.

Also known as asymmetric encryption, public key cryptography is used as a method of assuring the confidentiality, authenticity and non-repudiation of electronic communications and data storage. Public-key encryption uses two different keys at once, a combination of a private key and a public key. The private key must remain confidential to its respective owner, while the public key is made available to everyone via a publicly accessible repository or directory. To decode an encrypted message, a computer must use the public key, provided by the originating computer, and its own private key. Although a message sent from one computer to another won't be secure since the public key used for encryption is published and available to anyone, anyone who picks it up can't read it without the private key.

The key pair is based on prime numbers of long length. Both the public and private keys are computed together at the same time, in the same mathematical process, using “trapdoor” functions. The main characteristic of “trapdoor” functions is that they are easy to compute in one direction, yet difficult to compute in the opposite direction (finding its inverse) without special information.

The main disadvantage of asymmetric encryption is that it is slow when compared with symmetric encryption. This is because of the mathematical complexity involved in asymmetric encryption and therefore requires much more computing power to sustain. It is not suitable for long sessions because of the processing power it takes to keep it going.

Use Cases of Symmetric Encryption

Banking Sector. Due to the better performance and faster speed of symmetric encryption, symmetric cryptography is typically used for bulk encryption of large amounts of data. Applications of symmetric encryption in the banking sector include:

Payment applications, such as card transactions where PII (Personal Identifying Information) needs to be protected to prevent identity theft or fraudulent charges without huge costs of resources. This helps lower the risk involved in dealing with payment transactions on a daily basis.

Validations to confirm that the sender of a message is who he claims to be.

Data at rest. Data at rest is data that is not actively moving from device to device or network to network such as data stored on a hard drive, laptop, flash drive, or archived/stored in some other way. Data protection at rest aims to secure inactive data stored on any device or network. While data at rest is sometimes considered to be less vulnerable than data in transit, attackers often find data at rest a more valuable target than data in motion. For protecting data at rest, enterprises can simply encrypt sensitive files prior to storing them and/or choose to encrypt the storage drive itself.

The best way to encrypt data at rest is by whole disk or full disk encryption. Full disk encryption has several benefits compared to regular file or folder encryption, or encrypted vaults. Nearly everything including the swap space and the temporary files is encrypted. Encrypting these files is important, as they can reveal important confidential data. With a software implementation, the bootstrapping code cannot be encrypted, however. For example, BitLocker Drive Encryption leaves an unencrypted volume to boot from, while the volume containing the operating system is fully encrypted. In addition, the decision of which individual files to encrypt is not left up to users' discretion. This is important for situations in which users might not want or might forget to encrypt sensitive files.

Use Case of Asymmetric Encryption: Digital Signatures

Computer encryption is based on the science of cryptography, which has been used as long as humans have wanted to keep information secret. Most forms of cryptography in use nowadays rely on computers, simply because a human-

based code is too easy for a computer to crack. Cryptosystems use a set of procedures known as cryptographic algorithms, or ciphers, to encrypt plain text messages into cipher text or encrypted messages or decrypt cipher text messages into plain text.

Auguste Kerckhoff in 1883 stated that encryption algorithms should be made public and the “keys” be kept secret, which is Kerckhoff’s Principle. Computer encryption systems generally belong in one of two categories: symmetric encryption and asymmetric or public-key encryption.

Symmetric Encryption

In symmetric encryption, the sender and receiver use a separate instance of the same key to encrypt and decrypt messages. Symmetric encryption heavily relies on the fact that the keys must be kept secret. Distributing the key in a secure way is one of the primary challenges of symmetric encryption, which is known as the “key distribution problem.” The key that is the vital component in symmetric cryptography and we cannot afford to lose it or misplace it. If the individual keys are misplaced, the message can be decrypted by malicious actors.

The main advantage of symmetric cryptography is that it is much faster than asymmetric cryptography. The most important disadvantages of symmetric encryption are the key distribution problem and the key management problem. When the number of connected users grows, so does the number of required keys. Management of an increasing number of secret keys becomes “key management problem.” Further, symmetric cryptography ensures only the ‘confidentiality’ of the transmitted or stored data. It cannot be used to ensure integrity and/or authenticity.

Asymmetric Encryption

When connecting to a website on the public internet it becomes more complicated and symmetric encryption, by itself, won’t work because you don’t control the other end of the connection. How do you share a secret key with

each other without the risk of someone on the internet intercepting it in the middle? In November 1976, a paper published in the journal IEEE Transactions on Information Theory by Diffie and Hellman, titled "New Directions in Cryptography," addressed this problem and offered up a solution: public-key encryption.

Also known as asymmetric encryption, public key cryptography is used as a method of assuring the confidentiality, authenticity and non-repudiation of electronic communications and data storage. Public-key encryption uses two different keys at once, a combination of a private key and a public key. The private key must remain confidential to its respective owner, while the public key is made available to everyone via a publicly accessible repository or directory. To decode an encrypted message, a computer must use the public key, provided by the originating computer, and its own private key. Although a message sent from one computer to another won't be secure since the public key used for encryption is published and available to anyone, anyone who picks it up can't read it without the private key.

The key pair is based on prime numbers of long length. Both the public and private keys are computed together at the same time, in the same mathematical process, using “trapdoor” functions. The main characteristic of “trapdoor” functions is that they are easy to compute in one direction, yet difficult to compute in the opposite direction (finding its inverse) without special information.

The main disadvantage of asymmetric encryption is that it is slow when compared with symmetric encryption. This is because of the mathematical complexity involved in asymmetric encryption and therefore requires much more computing power to sustain. It is not suitable for long sessions because of the processing power it takes to keep it going.

Use Cases of Symmetric Encryption

Banking Sector. Due to the better performance and faster speed of symmetric encryption, symmetric cryptography is typically used for bulk encryption of large

amounts of data. Applications of symmetric encryption in the banking sector include:

Payment applications, such as card transactions where PII (Personal Identifying Information) needs to be protected to prevent identity theft or fraudulent charges without huge costs of resources. This helps lower the risk involved in dealing with payment transactions on a daily basis.

Validations to confirm that the sender of a message is who he claims to be.

Data at rest. Data at rest is data that is not actively moving from device to device or network to network such as data stored on a hard drive, laptop, flash drive, or archived/stored in some other way. Data protection at rest aims to secure inactive data stored on any device or network. While data at rest is sometimes considered to be less vulnerable than data in transit, attackers often find data at rest a more valuable target than data in motion. For protecting data at rest, enterprises can simply encrypt sensitive files prior to storing them and/or choose to encrypt the storage drive itself.

The best way to encrypt data at rest is by whole disk or full disk encryption. Full disk encryption has several benefits compared to regular file or folder encryption, or encrypted vaults. Nearly everything including the swap space and the temporary files is encrypted. Encrypting these files is important, as they can reveal important confidential data. With a software implementation, the bootstrapping code cannot be encrypted, however. For example, BitLocker Drive Encryption leaves an unencrypted volume to boot from, while the volume containing the operating system is fully encrypted. In addition, the decision of which individual files to encrypt is not left up to users' discretion. This is important for situations in which users might not want or might forget to encrypt sensitive files.

Use Case of Asymmetric Encryption: Digital Signatures

As organizations move away from paper documents with ink signatures or authenticity stamps, digital signatures can provide added assurances of the

evidence to provenance, identity, and status of an electronic document as well as acknowledging informed consent and approval by a signatory. Digital signatures are used to detect unauthorized modifications to data and to authenticate the identity of the signatory. In addition, the recipient of signed data can use a digital signature as evidence in demonstrating to a third party that the signature was, in fact, generated by the claimed signatory. This is known as non-repudiation, since the signatory cannot easily repudiate the signature at a later time.

The digital signatures standard was proposed by NIST and is defined in FIPS 186-4. Digital signatures employ asymmetric cryptography and they provide a layer of validation and security to messages sent through a non-secure channel. They enforce the concepts of authentication, non-repudiation, and confidentiality.

To create a digital signature and use it along with a message between two clients, Alice and Bob, the following steps are followed:

The message that has to be digitally signed by Alice is hashed creating a message digest. Hashing is the process that is used to enforce data integrity. Hashing functions take the message and add a string value and convert it to another value (message digest). Hashing functions are one-way which means that the message digest cannot be reverted back to the message.

The message digest is encrypted with Alice's private key. This is a digital signature.

The digital signature is now attached to the message and sent to Bob.

Once the message is received, Bob decrypts the digital signature with Alice's public key. This decryption results in a message digest.

Bob also hashes the message which results in the message digest again.

If the message digests in steps 4 and 5 above are the same, then Bob can be sure that Alice has signed the message and that the content of the message is as shown. Any difference in the hash values would reveal tampering of the message.

Digital signatures are intended for use in electronic mail, electronic funds transfer, electronic data interchange, software distribution, data storage, and

other applications that require data integrity assurance and data origin authentication.

Use Case of Asymmetric and Symmetric Encryption: Messaging Applications

Messaging applications, like Signal or Whatsapp, use end-to-end encryption to protect the confidentiality and privacy of the users' communications and to authenticate the users.

In end-to-end encryption, only the data is encrypted. The headers, trailers, and routing information are not. The basis for the end-to-end encryption is the Signal Protocol, designed by Open Whisper Systems. This end-to-end encryption protocol is designed to prevent third parties and the messaging vendor from having plaintext access to messages or calls. What's more, even if encryption keys from a user's device are ever physically compromised, they cannot be used to go back in time to decrypt previously transmitted messages.

Messaging end-to-end encryption is implemented using both asymmetric and symmetric cryptography. Asymmetric encryption is used to initialize the encrypted conversation between two users, and symmetric encryption is used to for the duration of the communication. The Whatsapp Encryption Overview White Paper provides the details.

Once the application is installed on a user's smartphone, the public keys of the client are registered with the application server. The private key is not stored in the server and remains secret in the user's device. The client who wants to initiate a session, retrieves from the Whatsapp server the public keys for the recipient. Using these keys, the initiator encrypts the first message and sends it to the recipient. This message contains the parameters for establishing a symmetric session key. The recipient uses his own private key to decrypt the message. "Once a session has been established, clients exchange messages that are protected with a Message Key using AES256 in CBC mode for encryption and HMAC-SHA256 for authentication." The encrypted session needs to be re-

created only when the device is changed or when the application software is re-installed.

THE END