Cryptography

1 MIN

Video – Cryptography

Early in the previous topic, cryptography is mentioned as part of the CIA information security triad. In this topic, you will get a deeper dive into the many types of cryptography and how they are used to secure the network.

Click Play in the figure to view a video about cryptography.

1 MIN

Securing Communications

Organizations now provide support to secure the data as it travels across risks. This may include internal traffic, but it is even more important to protect the data that travels outside of the organization to branch sites, telecommuters, and partners.

These are four elements of secure communications:

- **Data Integrity** – Guarantees that the message was not altered. Any changes to data in transit will be detected. Integrity is ensured by implementing either of the Secure Hash Algorithms (SHA-2 or SHA-3). The MD5 message digest algorithm is still widely in use but is inherently insecure and creates vulnerabilities to replays. The use of MD5 should be avoided.
- **Origin Authentication** – Guarantees that the message is not a forgery and does actually come from whom it states. Many modern networks ensure authentication with protocols such as hash message authentication code (HMAC).
- **Data Confidentiality** – Only the sender and intended receiver should "see" the message. If the message is intercepted, it should be encrypted so that it is unusable by anyone other. Data confidentiality is implemented using symmetric and asymmetric encryption algorithms.
- **Non-repudiation** – This means that the sender cannot deny that the message was not sent. Non-repudiation relies on the fact that only the sender has the unique characteristics or signature for the message is created.

Cryptography can be used almost anywhere that there is data communication. In fact, the trend is toward all communication being encrypted.

1 MIN

Data Integrity

Hash functions are used to ensure the integrity of a message. They guarantee that message data has not changed accidentally.

In the figure, the sender is sending a $100 money transfer to Alex.

The hash algorithm works as follows:

1. The sending device feeds the message into a hashing algorithm and computes its fixed-length input or **modified fingerprint**.
2. The hash value is then appended to the message and sent to the receiving device.
3. The receiving device removes the hash from the message and feeds the remaining message into the same hashing algorithm. If the computed hash is equal to the one that was attached to the message, the message has not been altered during transit. If the hashes are not equal, as shown in the figure, then the integrity of the message can no longer be trusted.

1 MIN

Hash Functions

There are four well-known hash functions.

MD5 with 128 bit Digest

Developed by Ron Rivest and used in a variety of Internet applications, MD5 is a one-way function that produces a 128-bit hashed message. MD5 is considered to be a legacy algorithm and should be avoided and used only when no better alternatives are available. It is recommended that SHA-2 or SHA-3 be used instead.

In the figure, a plaintext message is passed through an MD5 hash function. The result is a 128 bit hashed message.

SHA Hashing Algorithm

Developed by the U.S. National Security Agency (NSA) in 1995, it is very similar to the MD5 hash functions. Several versions exist. SHA-1 creates a 160-bit hashed message and is slightly slower than MD5. SHA-1 has known flaws and is a legacy algorithm.

In the figure, a plaintext message is passed through a SHA hash function. The result is a hashed message.

SHA-2

Developed by the NSA, it includes SHA-224 (224-bit), SHA-256 (256-bit), SHA-384 (384-bit), and SHA-512 (512-bit). If you are using SHA-2, then the SHA-256, SHA-384, and SHA-512 algorithms should be used whenever possible.

SHA-3

SHA-3 is the newest hashing algorithm and was introduced by NIST as an alternative and eventual replacement for the SHA-2 family of hashing algorithms. SHA-3 includes SHA-224 (224-bit), SHA-256 (256-bit), SHA-384 (384-bit), and SHA-512 (512-bit). The SHA-3 family are even generation algorithms and should be used when possible.

While hashing can be used to detect accidental changes, it cannot be used to guard against deliberate changes that are made by a threat actor. There is no unique hashing information from the sender is the hashing procedure. This means that anyone can modify the hash, as long as they have the correct hash function.

For example, when the message traverses the network, a potential attacker could intercept the message, change it, recalculate the hash, and append it to the message. The receiver will have no means of determining that the message is now compromised.

Therefore, hashing is vulnerable to man-in-the-middle attacks and does not provide security for transmitted data. To provide integrity and origin authentication, messages must be assigned a hash that is **keyed**.

Note: Hashing algorithms only protect against accidental changes and does not protect the data from changes deliberately made by a threat actor.

1 MIN

Origin Authentication

To add authentication to integrity assurance, use a keyed-hash message authentication code (HMAC). HMAC uses an additional secret key as input to the hash function.

Click each button for an illustration and explanation about origin authentication using HMAC.

Hash Message Authentication Code (HMAC) | Creating the HMAC Value | Verifying the HMAC Value | Client Server HMAC Example

As shown in the figure, an HMAC is calculated using any cryptographic algorithm that combines a cryptographic hash function with a secret key. Hash functions are hard to reverse and computationally fast.

Only the sender and the receiver know the secret key, and the output of the hash function now depends on the hash data and the secret key. Only parties who have access to that secret key can compute the digest of an HMAC function. This defeats man-in-the-middle attacks and provides authentication of the data origin.

If two parties share a secret key and use HMAC functions for authentication, a properly constructed HMAC digest of a message that a party has received indicates that the other party was the originator of the message. This is because the other party possesses the secret key.

1 MIN

Data Confidentiality

There are two classes of encryption used to provide data confidentiality. These two classes differ in how they use keys.

Symmetric encryption algorithms such as Data Encryption Standard (DES), DES, and Advanced Encryption Standard (AES) are based on the premise that each communicating party knows the same key. Data confidentiality can also be achieved using asymmetric algorithms, including Rivest, Shamir, and Adleman (RSA) and the public key infrastructure (PKI).

Note: DES is a legacy algorithm and should not be used. DES should be avoid if possible.
The figure highlights some differences between each encryption algorithm method.

The figure shows the differences between symmetric and asymmetric encryption. Characteristics of symmetric encryption include: use the same key to encrypt and decrypt data, key lengths are short (56 bits - 256 bits), faster than asymmetric encryption, and commonly used for encrypting bulk data such as in VPN traffic. Characteristics of asymmetric encryption include: uses different keys to encrypt and decrypt data, key lengths are long (512 bits - 4096 bits), computationally taxing therefore slower than symmetric encryption, and commonly used for quick data transactions such as HTTPS when accessing your bank data.

1 MIN

Symmetric Encryption

Symmetric algorithms use the same pre-shared key to encrypt and decrypt data. If pre-shared key, also called a secret key, is known by the sender and receiver before any encrypted communications can take place.

To help illustrate how symmetric encryption works, consider an example where Alice and Bob want to securely exchange secret messages with one another through the mail system. In this example, Alice wants to send a secret message to Bob.

In the figure, Alice and Bob have identical keys to a single padlock. These keys were exchanged prior to sending any secret messages. Alice writes a secret message and puts it in a small box that she locks using the padlock with her key. Alice then sends the box to Bob. The message is safely locked inside the box as the box makes its way through the mail system. When the box arrives at the destination, he uses his key to unlock the padlock and retrieve the message. Bob can use the same lock and padlock to send a secret reply to Alice.

Symmetric Encryption Example

Symmetric encryption algorithms fall into two categories, block and stream ciphers. Block ciphers encrypt data in fixed-length blocks. The blocks are usually 64-bit or 128-bit. Padding is added if the plaintext data length is not a multiple of 8 or make a complete block. Examples of block ciphers include DES, and AES. Stream ciphers encrypt data one bit or byte at a time in a continuous string.
Examples of stream ciphers include RC4 and RC4+.

Today, symmetric encryption algorithms are commonly used with VPN traffic. This is because symmetric algorithms use less CPU resources than asymmetric encryption algorithms and can encrypt and decrypt data at a much faster rate. Symmetric encryption algorithms are also used to encrypt data on hard drives. When using this type of encryption, the length of self-test for someone to discover the key. Most encryption keys are between 112 and 256 bits. To ensure that the encryption is safe, use a minimum key length of 128 bits. Use a longer key for more secure communications.

Most known symmetric encryption algorithms are described in the table.

Symmetric Encryption Algorithm	Description
Data Encryption Standard (DES)	This is a legacy symmetric encryption algorithm. It uses a short key length that makes it insecure for most current uses.
AES (Triple DES)	This is the replacement for DES and applies the DES algorithm process three times. It should be avoided if possible as it is scheduled to be retired in 2023. If implemented, use very short keys lifetimes.
Advanced Encryption Standard (AES)	AES is a very fast and reliable encryption algorithm. It is recommended and mandatory for most government symmetric encryption algorithm. It offers nine combinations of key and block length by using a block length of 128, 192, or 256 bits and a key length also blocks that are 128, 192, or 256 bits long.
Rivest Cipher Symmetric Encryption Algorithm (RC4+)	RC4+ is a faster alternative symmetric encryption algorithm to DES, 3DES, and AES. It uses a 160 bit encryption key and has a lower CPU requirement than 3DES and AES. It is a proprietary cipher.
Rivest ciphers (RC) series algorithms	This algorithm was developed by Ron Rivest. Several ciphers have been developed, but RC4 is the most prevalent in use. RC4 is a cipher that was used in secure web traffic. It has been found to have multiple vulnerabilities which have made it insecure. RC4 should not be used.

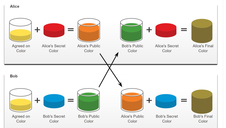
1 MIN

Asymmetric Encryption

Asymmetric algorithms, also called public-key algorithms, are designed so that the key that is used for encryption is different from the key that is used for decryption, as shown in the figure. The decryption key cannot, in any reasonable amount of time, be calculated from the encryption key and vice versa.

Asymmetric algorithms use a public key and a private key. Both keys are capable of the encryption process, but the complementary pated key is required for decryption. The encryption process cannot be accomplished with the public key requires the private key to decrypt. Asymmetric algorithms enhance confidentiality, authentication, and integrity by using this process.

Asymmetric Encryption Example



Because neither party has a shared secret, very long key lengths must be used. Asymmetric encryption can use key lengths between 512 to 4,096 bits. Key lengths greater than or equal to 2,048 bits are considered safe when key lengths are considered candidate.

Examples of protocols that use asymmetric key algorithms include:

- **Internet Key Exchange (IKE)** – This is a fundamental component of IPsec VPNs.
- **Secure Socket Layer (SSL)** – This is now deprecated but still used across Transport Layer Security (TLS).
- **Secure Shell (SSH)** – This protocol provides a secure remote access connection to network devices.
- **Pretty Good Privacy (PGP)** – This computer program provides cryptographic privacy and authentication. It is often used to increase the security of email communications.

Asymmetric algorithms are substantially slower than symmetric algorithms. Their design is based on computational problems, such as factoring extremely large numbers or computing discrete logarithms of extremely large numbers.

Because they are slow, asymmetric algorithms are typically used in low volume cryptographic mechanisms, such as digital signatures and key exchange. However, the key management of asymmetric algorithms tends to be simpler than symmetric algorithms, because usually one of the two encryption or decryption keys can be made public.

Common examples of asymmetric encryption algorithms are described in the table.

Asymmetric Encryption Algorithm	Key Length	Description
Diffie-Hellman (DH)	512 - 1024, 2048, 3072, 4096	This Diffie-Hellman algorithm allows two parties to agree on a key that they can use to encrypt messages they want to send to each other. The security of this algorithm depends on the assumption that it is easy to raise a number to a certain power, but difficult to compute which power was used given the number and the outcome.
Digital Signature Standard (DSS)	512 - 1024	DSS specifies DSA as the algorithm for digital signatures. DSS is a public key algorithm based on the Elliptical Signature Algorithm. Digitalsignature creation speed is similar to RSA, but is 10 to 40 times slower for verification.
Rivest, Shamir, and Adleman encryption algorithm (RSA)	512 to 2048	RSA is for public key cryptography that is based on the current difficulty of factoring very large numbers. It is the first algorithm known to be suitable for signing as well as encryption. It is widely used in electronic commerce protocols and is believed to be sufficiently secure given sufficiently long keys and the use of cryptographic extensions.
Elliptical	512 - 1024	An asymmetric key encryption algorithm for public-key cryptography which is based on the Diffie-Hellman key agreement. A disadvantage of the Elliptical system is that the encrypted message becomes very big, about twice the size of the original message, and it is difficult to add digital signatures to the original message.
Elliptic curve technologies	224	Elliptic curve cryptography can be used to adapt many cryptographic algorithms, such as Diffie-Hellman or Elliptical. The security and efficiency of elliptic curve cryptography is that the keys can be much smaller.

1 MIN

Diffie–Hellman

Diffie-Hellman (DH) is an asymmetric mathematical algorithm where two computers generate an identical shared secret key without having communicated before. The shared secret key is never actually exchanged between the sender and receiver. However, because both parties know it, the key can be used by an encryption algorithm to encrypt traffic between the two systems.

Here are three examples of instances when DH is commonly used:

- Data is exchanged using an IPsec VPN.
- Two devices have established using an SSL or TLS VPN.
- SSH data is exchanged.

To help illustrate how DH operates, refer to the figure.

The colors in the figure will be used instead of complex long numbers to simplify the DH key agreement process. The DH key exchange begins with Alice and Bob agreeing on an arbitrary common color that does not need to be kept secret. The agreed on color in the example is yellow.

Next, Alice and Bob each select a secret color. Alice chose red while Bob chose blue. These secret colors will never be shared with anyone. The secret color represents the private key of Alice and Bob.

Next each Bob may now the shared common color (yellow) and his/her secret color to produce a public color. Therefore, Alice will mix the yellow with her red color to produce a public color of orange. Bob will mix the yellow and the blue to produce a public color of green.

Alice sends her public color (orange) to Bob and Bob sends his public color (green) to Alice.

Alice and Bob each mix the color they received with their own, original secret color (that they have and blue for Bob.). The result is a final brown color mixture that is identical to the partner's final color mixture. The brown color represents the resulting shared secret key between Bob and Alice.

The security of DH is based on the fact that it uses very large numbers in its calculations. For example, a DH 1024 bit number is roughly equal to a decimal number of 309 digits. Considering that a billion is 10 decimal digits (1,000,000,000), one can easily imagine the complexity of working with very, very, very multiple 309 digit decimal numbers.

Unfortunately, asymmetric key systems are extremely slow for any sort of scale encryption. This is why it is common to encrypt the bulk of the traffic using a symmetric algorithm, such as DES or AES and use the DH algorithm to create keys that will be used by the encryption algorithm.

1 MIN

Check Your Understanding – Cryptography

Check your understanding of cryptography by choosing the BEST answer to the following questions.

1. Which encryption method repeats an algorithm process three times and is considered very trustworthy when implemented using very strong key lengths?
   - Rivest Cipher
   - Triple DES
   - Block Cipher
   - Data Encryption Standard
   - Stream Cipher
2. Which encryption method encrypts plaintext one byte or one bit at a time?
   - Rivest Cipher
   - Block Cipher
   - Data Encryption Standard
   - Software Encryption Algorithm
   - Stream Cipher
3. Which encryption method uses the same key to encrypt and decrypt data?
   - Triple DES
   - Symmetric
   - Block Cipher
   - Data Encryption Standard
   - Asymmetric
4. Which encryption method is a stream cipher and is used to secure web traffic in SSL and TLS?
   - Rivest Cipher
   - Triple DES
   - Symmetric
   - Block Cipher
   - Data Encryption Standard

Check | Show Me | Reset