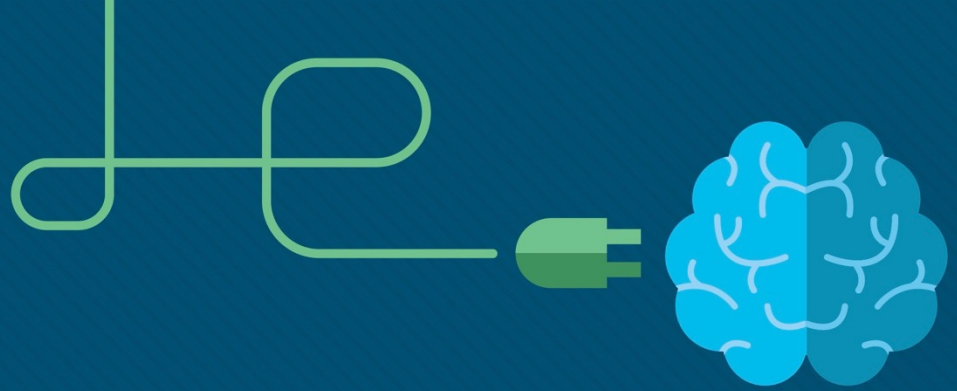


Chapter 7: Cryptography and the Public Key Infrastructure

Information Security

Dr. Ayman Aljarbouh

7.1 Cryptography

Module Objectives

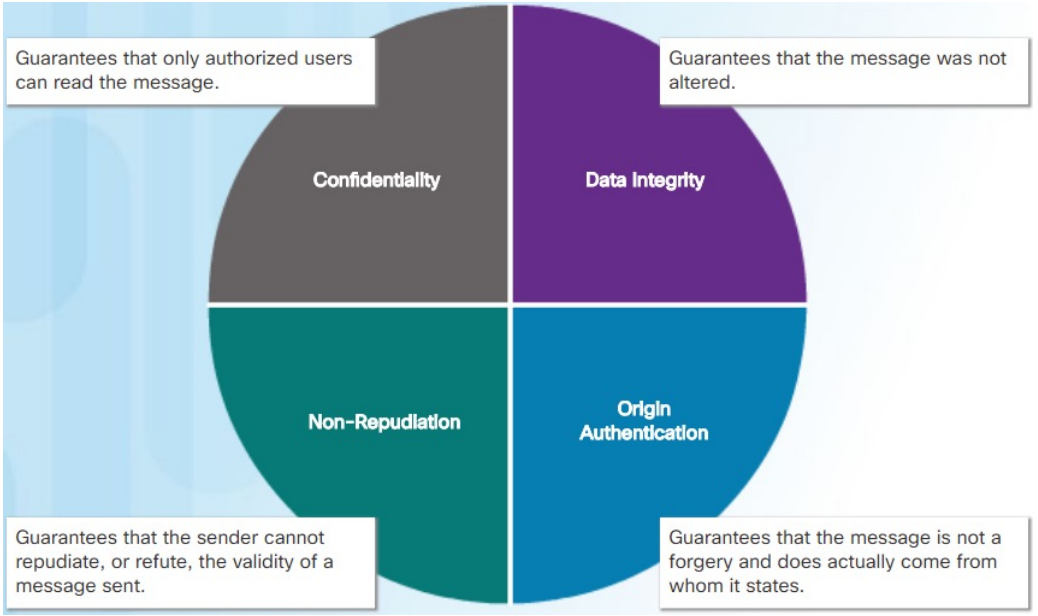
Module Title: Cryptography

Module Objective: Use tools to encrypt and decrypt data.

Topic Title	Topic Objective
What is Cryptography?	Use cryptography to secure communications.
Integrity and Authenticity	Explain the role of cryptography in ensuring the integrity and authenticity of data.

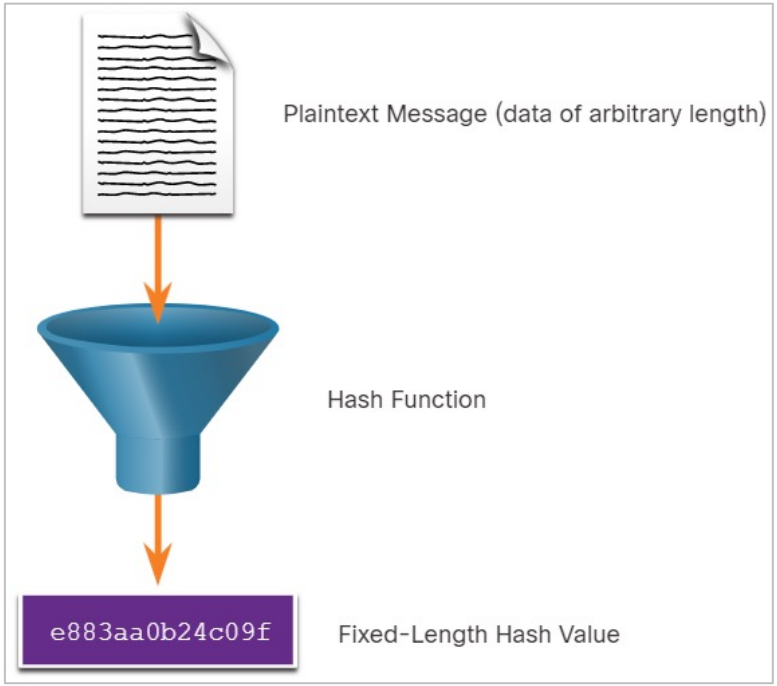
Securing Communications

- Information security concerns protecting network infrastructure devices and securing data as it travels on the network.
- Cryptography helps realize the four objectives of information security:
  - **Data Confidentiality** - only authorized users can read the data.
  - **Data Integrity** - the data has not been altered by unauthorized parties.
  - **Origin authentication** - the data has actually originated at the expected source.
  - **Non-repudiation** – the integrity of the message is irrefutable by the sender.



Cryptographic Hash Functions

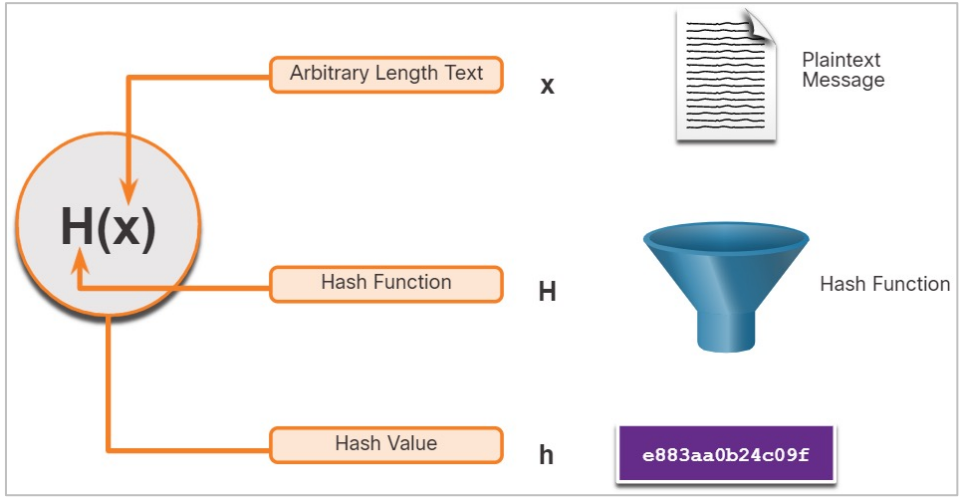
- Hashes are used to verify and ensure data integrity.
- Hashing is based on a one-way mathematical function that is relatively easy to compute, but significantly harder to reverse.
- A hash function takes a variable block of binary data, called the message, and produces a fixed-length, condensed representation, called the hash.
- The resulting hash is also sometimes called the message digest, digest, or digital fingerprint.
- With hash functions, it is computationally infeasible for two different sets of data to come up with the same hash output.
- Every time the data is changed or altered, the hash value also changes.



Cryptographic Hash Operation

- Mathematically, the equation ***h= H(x)*** is used to explain how a hash algorithm operates.
- As shown in the figure, a hash function H takes an input x and returns a fixed-size string hash value h.

- A cryptographic hash function should have the following properties:
  - The input can be any length.
  - The output has a fixed length.
  - H(x) is relatively easy to compute for given x.
  - H(x) is one way and not reversible.
  - H(x) is collision free, meaning that two different input values will result in different hash values.

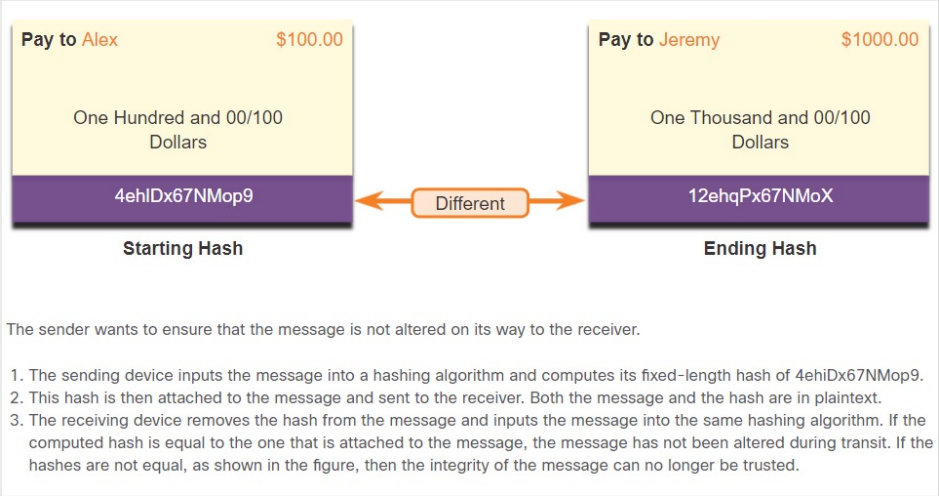


MD5 and SHA

- Hash functions are used to ensure the integrity of a message either accidentally or intentionally.
- In the figure, the sender is sending a $100 money transfer to Alex. The sender wants to ensure that the message is not altered on its way to the receiver.

There are four well-known hash functions:

- **MD5 with 128-bit digest** - A one-way function that produces a 128-bit hashed message. MD5 is a legacy algorithm.
- **SHA-1 -** Very similar to the MD5 hash functions. SHA-1 creates a 160-bit hashed message and is slightly slower than MD5.
- **SHA-2 -** If you are using SHA-2, then SHA-256, SHA-384, and SHA-512 algorithms should be used.



- **SHA-3 -** Next-generation algorithms and should be used whenever possible.

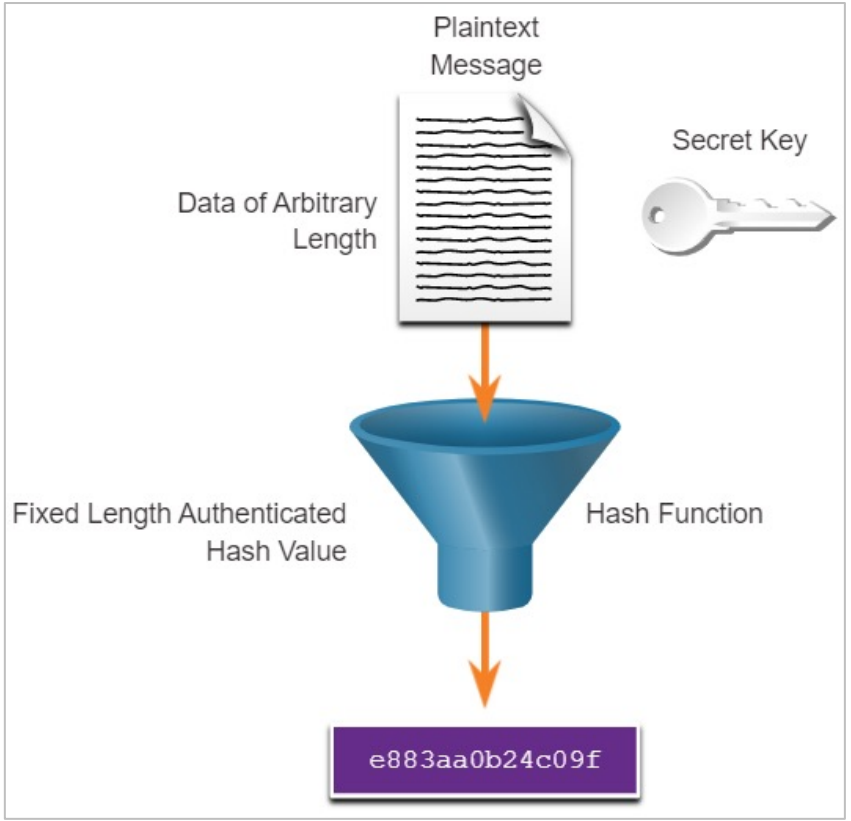
MD5 and SHA (Contd.)

- 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 identifying information from the sender in the hashing procedure.
- This means that anyone can compute a hash for any data, as long as they have the correct hash function.
- Therefore, hashing is vulnerable to man-in-the-middle attacks and does not provide security to transmitted data. To provide integrity and origin authentication, something more is required.

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

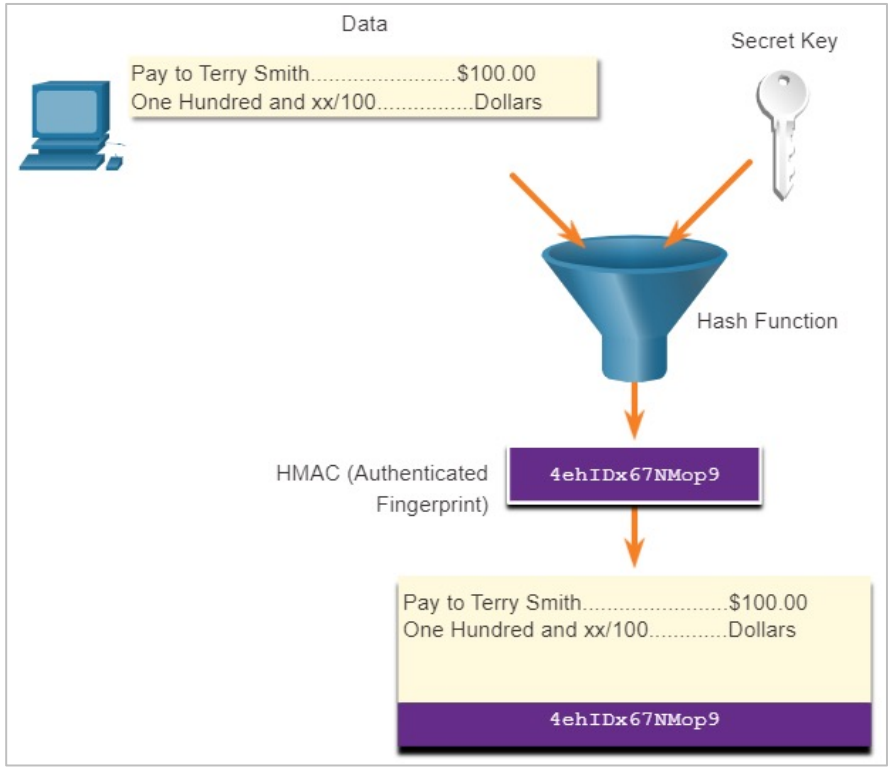
Hash Message Authentication Code

- To add authentication to integrity assurance, a keyed-hash message authentication code (HMAC) is used.
- To add authentication, HMAC uses an additional secret key as input to the hash function.
- Only the sender and the receiver know the secret key, and the output of the hash function now depends on the input data and the secret key.
- Only parties who have access to that secret key can compute the digest of an HMAC function.
- If the digest that is calculated by the receiving device is equal to the digest that was sent, the message has not been altered.



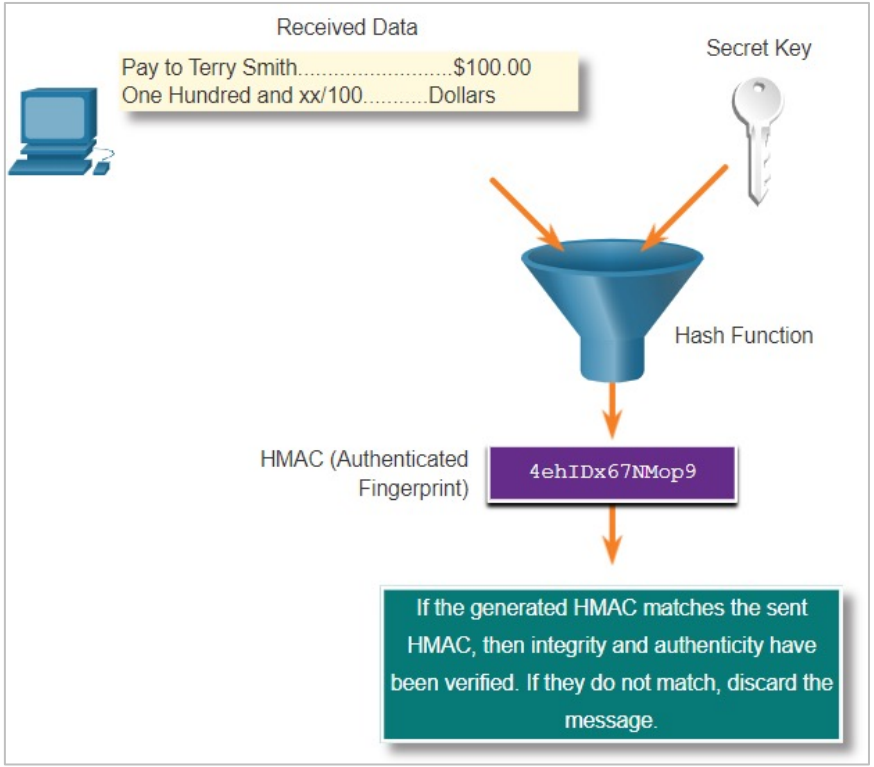
Hash Message Authentication Code

- To add authentication to integrity assurance, a keyed-hash message authentication code (HMAC) is used.
- To add authentication, HMAC uses an additional secret key as input to the hash function.
- Only the sender and the receiver know the secret key, and the output of the hash function now depends on the input data and the secret key.
- Only parties who have access to that secret key can compute the digest of an HMAC function.
- If the digest that is calculated by the receiving device is equal to the digest that was sent, the message has not been altered.



Hash Message Authentication Code

- To add authentication to integrity assurance, a keyed-hash message authentication code (HMAC) is used.
- To add authentication, HMAC uses an additional secret key as input to the hash function.
- Only the sender and the receiver know the secret key, and the output of the hash function now depends on the input data and the secret key.
- Only parties who have access to that secret key can compute the digest of an HMAC function.
- If the digest that is calculated by the receiving device is equal to the digest that was sent, the message has not been altered.



New Terms and Commands

• Hash-based Message Authentication Code (HMAC) • Data Confidentiality • Data Integrity	• Origin authentication • Non-repudiation • Secure Hash Algorithm 1 (SHA-1) • Secure Hash Algorithm 2 (SHA-2)

Lab 25 - Creating Codes

Secret codes have been used for thousands of years. Ancient Greeks and Spartans used a scytale (rhymes with Italy) to encode messages. Romans used a Caesar cipher to encrypt messages. A few hundred years ago, the French used the Vigenère cipher to encode messages. Today, there are many ways that messages can be encoded.

In this activity, you will create and encrypt messages using online tools.

Lab 26 - Hashing Things Out

In this lab, you will complete the following objectives:

- Creating Hashes with OpenSSL
- Verifying Hashes