3.1 Cryptographic principles

Cryptographic principles

Cryptography is the science of secret writing with the goal of hiding the meaning of a message. ***Cryptanalysis*** is the science of breaking cryptography. ***Cryptology*** is the study of cryptography and cryptanalysis.

Cryptography provides four main security services:

- ***Confidentiality***: Data can only be viewed by authorized entities. Confidentiality is also called ***secrecy***.
- ***Integrity***: Data cannot be modified in an unauthorized manner since the data was created, stored or transmitted by an authorized entity. The data integrity security service does not prevent unauthorized data modifications, but provides a means to detect unauthorized data modifications.
- ***Message authentication***, or ***data origin authentication***: A message receiver can verify the source of a message. The message authentication security service implies the message integrity.
- ***Non-repudiation***: A message sender cannot deny sending a message. The non-repudiation security service provides assurance to a message receiver that the message originated from the sender.

PARTICIPATION ACTIVITY | 3.1.1: Cryptographic security services.

How to use this tool

Confidentiality **Integrity** **Non-repudiation** **Message authentication**

	Data can only be viewed by authorized entities.
	Data cannot be modified in an unauthorized manner since the data was created, stored or transmitted by an authorized entity.
	A message receiver can verify the source of a message.

A message sender cannot deny sending a message.

Reset

PARTICIPATION ACTIVITY 3.1.2: Cryptographic security services.

1) A message is sent over an insecure network. Which security service ensures the message receiver that the message was not modified in transit?
   - Confidentiality
   - Integrity
   - Message authentication
   - Non-repudiation

2) A user receives a message from a sender. Which security service enables the receiver to verify the source of the message?
   - Confidentiality
   - Integrity
   - Message authentication
   - Non-repudiation

3) A file containing personally identifiable information is stored on a computer connected to the Internet. Which security service can prevent unauthorized users from viewing the file?
   - Confidentiality
   - Integrity
   - Message authentication
   - Non-repudiation

4) A user receives a message from a message sender, but the sender denies sending the message. Which security

service ensures the user that the message originated from the sender?

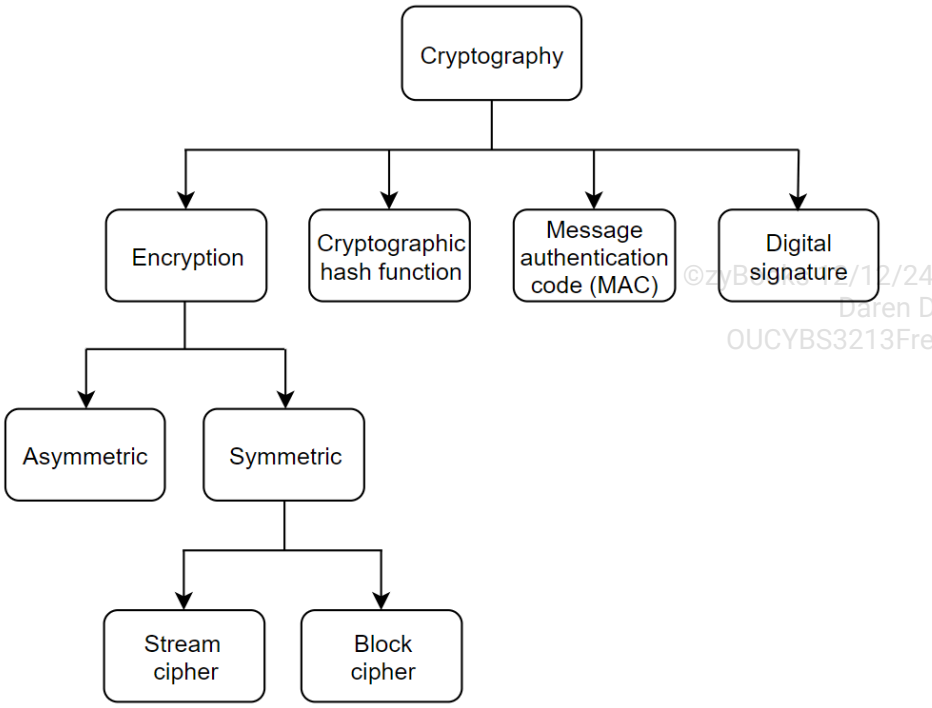
- Confidentiality
- Integrity
- Message authentication
- Non-repudiation

Cryptographic primitives

Cryptographic primitives are the basic building blocks of cryptography. A cryptographic primitive provides a security service. Four cryptographic primitives exists:

- ***Encryption***: Encryption uses an algorithm and a key to hide the meaning of a message. Encryption can be symmetric or asymmetric. In ***asymmetric encryption*** two different, but mathematically related keys are used for encryption and decryption. In ***symmetric encryption*** the same key is used for encryption and decryption. Symmetric encryption can use stream or block ciphers. A stream cipher encrypts data one bit at a time. A block cipher encrypts data one block-of-bits at a time. Encryption provides the confidentiality security service.
- ***Cryptographic hash function***: A cryptographic hash function outputs a fixed-length string for a variable-length input string. A cryptographic hash function provides the data integrity security service.
- ***Message authentication code (MAC)***: A message authentication code uses a cryptographic hash function and symmetric encryption to provide the data integrity and authenticity security services.
- ***Digital signature***: A digital signature uses a cryptographic hash function and asymmetric encryption to provide the data integrity, authenticity, and non-repudiation security services.

Figure 3.1.1: Cryptographic primitives.



PARTICIPATION ACTIVITY 3.1.3: Cryptographic primitives.

How to use this tool

Message authentication code (MAC) **Digital signature** **Cryptographic hash function**

Encryption

	Provides the data confidentiality security service.
	Provides the data integrity security service.
	Provides the data integrity and authenticity security services.
	Provides the data integrity, origin authentication, and non-repudiation security services.

Reset

PARTICIPATION ACTIVITY 3.1.4: Cryptographic primitives.

1) Which cryptographic primitives are used by a digital signature?
   - A cryptographic hash function and symmetric encryption
   - A cryptographic hash function and asymmetric encryption
   - A cryptographic hash function and message authentication code (MAC)

2) Which cryptographic primitive(s) is used by a message authentication code (MAC)?
   - A cryptographic hash function
   - A cryptographic hash function and symmetric encryption
   - Asymmetric encryption

3) Which cryptographic primitive(s) is used to provide the confidentiality security service?
   - Digital signature
   - Message Authentication code (MAC)
   - Encryption

©zyBooks 12/12/24 18:01 2172291
Daren Diaz
OUCYBS3213FreezeFall2024

3.2 Historical cryptosystems

Historical cryptosystems

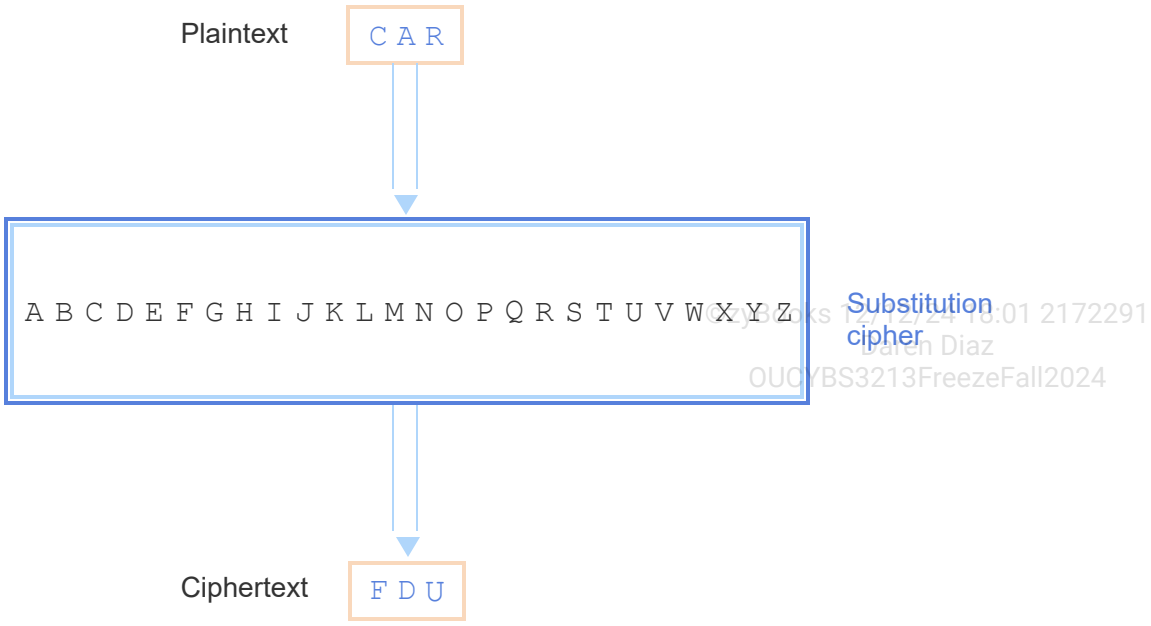
Historical cryptosystems predate the computer age. Unlike modern cryptosystems that use techniques based on the difficulty of solving mathematical problems, historical cryptosystems used simple operations to scramble letters of the alphabet to hide a message's meaning. Historical cryptosystems no longer provide any meaningful security because those systems are easily solved, but techniques used by historical cryptosystems form the basis of modern day cryptography. Two such techniques are substitution and transposition.

Substitution cipher

A ***substitution cipher*** is a cipher which substitutes or replaces one element of plaintext with a different element to generate a ciphertext. In a ***monoalphabetic substitution cipher***, a letter in a plaintext is replaced by a fixed letter to generate a ciphertext. Ex: The letter 'C' in a plaintext is always replaced by the letter 'X'. In a ***polyalphabetic substitution cipher***, a specific letter in a plaintext is replaced by different letters in a ciphertext. Ex: The letter 'C' in a plaintext may be replaced by the letters 'P', 'T', 'X', or any other letter.

A ***shift cipher*** is a substitution cipher which replaces a plaintext's letter by a letter that is a fixed number of positions to the right of the letter in the alphabet. For a shift cipher, the fixed number of positions is the encryption key. The ***Caesar cipher*** is a shift cipher which shifts each plaintext's letter by 3 positions to the right, with the letter 'Z' wrapping back to the letter 'A'. The encryption key of the Caesar cipher is 3. Ex: The Caesar cipher replaces the letter 'A' in a plaintext with the letter 'D' in a ciphertext. The letter 'D' is 3 positions to the right of the letter 'A'.

PARTICIPATION ACTIVITY 3.2.1: Caesar cipher.



Animation content:

Step 1: Substitution cipher consists of the English alphabet A through Z. Plaintext box appears at the top, and a ciphertext box appears underneath. Step 2: C is the chosen plaintext letter, appearing in the plaintext box. Step 3: From the location in the alphabet for the letter C, an arrow appears and moves 5 spaces to the right to the letter H. Step 4: The letter H appears in the ciphertext box.

Animation captions:

1. A substitution cipher replaces each letter of a plaintext with a different letter in a ciphertext.
2. The Caesar cipher shifts each letter three positions to the right to find the substitute letter.

The ***ROT13 cipher*** is a monoalphabetic substitution cipher which rotates each letter of a plaintext 13 places to the right in the alphabet. The ROT13 is short for rotate by 13 places. The encryption key of the ROT13 cipher is 13. Ex: The ROT13 cipher replaces the letter 'A' in a plaintext with the letter 'N' in a ciphertext. The letter 'N' is 13 positions to the right of the letter 'A'.

PARTICIPATION ACTIVITY | 3.2.2: Substitution cipher.

How to use this tool

Polyalphabetic substitution cipher **Monoalphabetic substitution cipher**

ROT13 cipher **Caesar cipher**

	A cipher which shifts each letter 3 positions to the right of the letter in the alphabet.
	A cipher which shifts each letter 13 positions to the right of the letter in the alphabet.
	A cipher which replaces a plaintext's letter by a fixed letter.

A cipher which replaces a plaintext's letter by different letters in a ciphertext.

Reset

PARTICIPATION ACTIVITY 3.2.3: Substitution cipher.

1) A cipher generates the ciphertext by rearranging the letters in a plaintext. Is the cipher a substitution cipher?
   - No
   - Yes

2) What does the Caesar cipher generate for the letter 'C'?
   - The letter 'F'.
   - The letter 'H'

3) A substitution cipher replaces a letter of a plaintext by shifting 5 positions to the right. What does the cipher generate for the letter 'F'?
   - The letter 'K'.
   - The letter 'M'

4) What does the ROT13 cipher generate for the letter 'B'?
   - The letter 'O'.
   - The letter 'J'

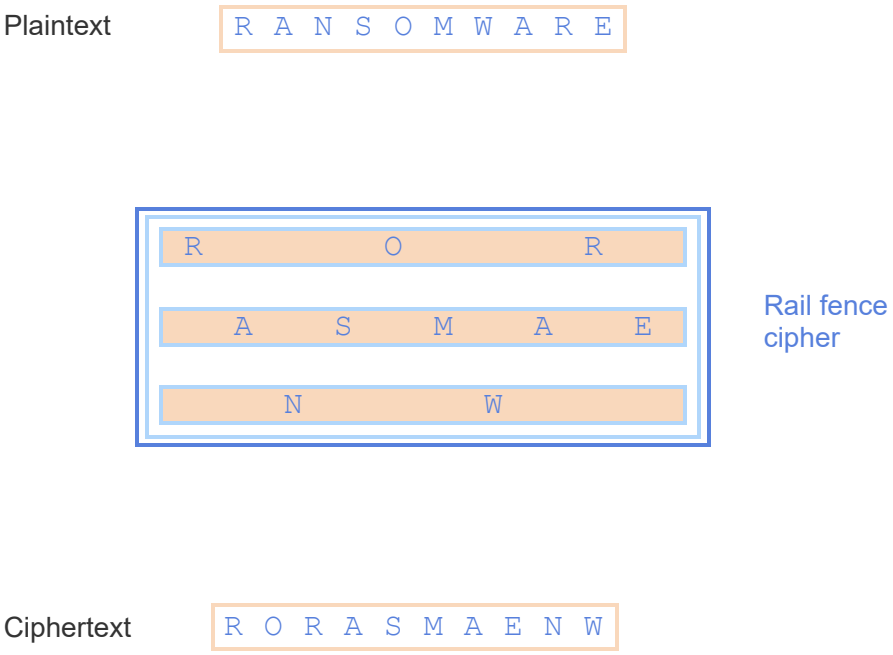
Transposition cipher (rail fence cipher)

A ***transposition cipher***, or ***permutation cipher***, is a cipher which reorders elements of a plaintext in the ciphertext without adding or removing elements. The elements of a ciphertext are in a different order than the elements of the plaintext. A transposition cipher does not replace an element of a plaintext in the ciphertext. Ex: A cipher that reverses the order of letters in the plaintext is a transposition cipher. The word 'elephant' would then become 'tnahpele'.

A ***rail fence cipher*** is a transposition cipher in which a plaintext's letters are written diagonally downwards on successive *rails* of an imaginary fence. After reaching the bottom rail, a plaintext's

letters are written diagonally upwards on successive rails until reaching the top rail. The process continues until all of a plaintext's letters are written on the rails. The ciphertext is read rail-by-rail from the top rail to the bottom rail. The key of a rail fence cipher is the number of rails.

PARTICIPATION ACTIVITY 3.2.4: Rail fence cipher.



Animation content:

Static image: A box at the top labeled "Plaintext" contains the text "RANSOMWARE." A box in the middle labeled "Rail fence cipher" has three lines of text. The first line has "R" then three spaces then "O" then three spaces then "R" then one space. The second line has one space then "A" then one space then "S" then one space then "M" then one space then "A" then one space then "E." The third line has two spaces then "N" then two paces then "W" then two spaces. A box at the bottom labeled "Ciphertext" contains the text "RORASMAENW."

Step 1: A transposition cipher rearranges a plaintext's letters to generate a ciphertext.

Three boxes appear. The box at the top is labeled "Plaintext," the box in the middle is labeled "Rail fence cipher," and the box at the bottom is labeled "Ciphertext." The text "RANSOMWARE" appears in the Plaintext box.

Step 2: In a rail fence cipher with a key of 3, a plaintext's letters are written diagonally downwards and upwards on 3 successive rails of an imaginary fence.
A copy of each letter of "RANSOMWARE" moves from the Plaintext box into the Rail fence cipher box one at a time. The "R" moves into the first spot of the first line of the Rail fence cipher box. The "A" moves into the second spot of the second line. The "N" moves into the third spot of the third line. The "S" moves into the fourth spot of the second line. The "O" moves into the fifth spot of the first line. The "M" moves into the sixth spot of the second line. The "W" moves into the seventh spot of the third line. The "A" moves into the eighth spot of the second line. The "R" moves into the ninth spot of the first line. The "E" moves into the tenth spot of the second line.
Step 3: The ciphertext is constructed by placing the letters on each rail next to each other, starting from the top rail and ending with the bottom rail.
The top line of the Rail fence cipher is highlighted in blue. Copies of the letters "R," "O," and "R" move from the first line into the Ciphertext box. The second line is highlighted in blue. Copies of the letters "A," "S," "M," "A," and "E" move from the second line into the Ciphertext box. The third line is highlighted in blue. Copies of the letters "N" and "W" move from the third line into the Ciphertext box. The Ciphertext box now reads "RORASMAENW."

Animation captions:

1. A transposition cipher rearranges a plaintext's letters to generate a ciphertext.
2. In a rail fence cipher with a key of 3, a plaintext's letters are written diagonally downwards and upwards on 3 successive rails of an imaginary fence.
3. The ciphertext is constructed by placing the letters on each rail next to each other, starting from the top rail and ending with the bottom rail.

PARTICIPATION ACTIVITY | 3.2.5: Transposition cipher.

How to use this tool

Permutation cipher **Rail fence cipher**

	A cipher in which a plaintext's letters are written diagonally downwards and upwards on successive rails of an imaginary fence.
	A cipher which reorders, or scrambles elements of a plaintext in a ciphertext without adding or removing elements.

Reset

PARTICIPATION ACTIVITY 3.2.6: Transpsition cipher.

1) A cipher replaces a letter in a plaintext with a letter five positions to the left of the letter. Is the cipher a transposition cipher?
   - No
   - Yes

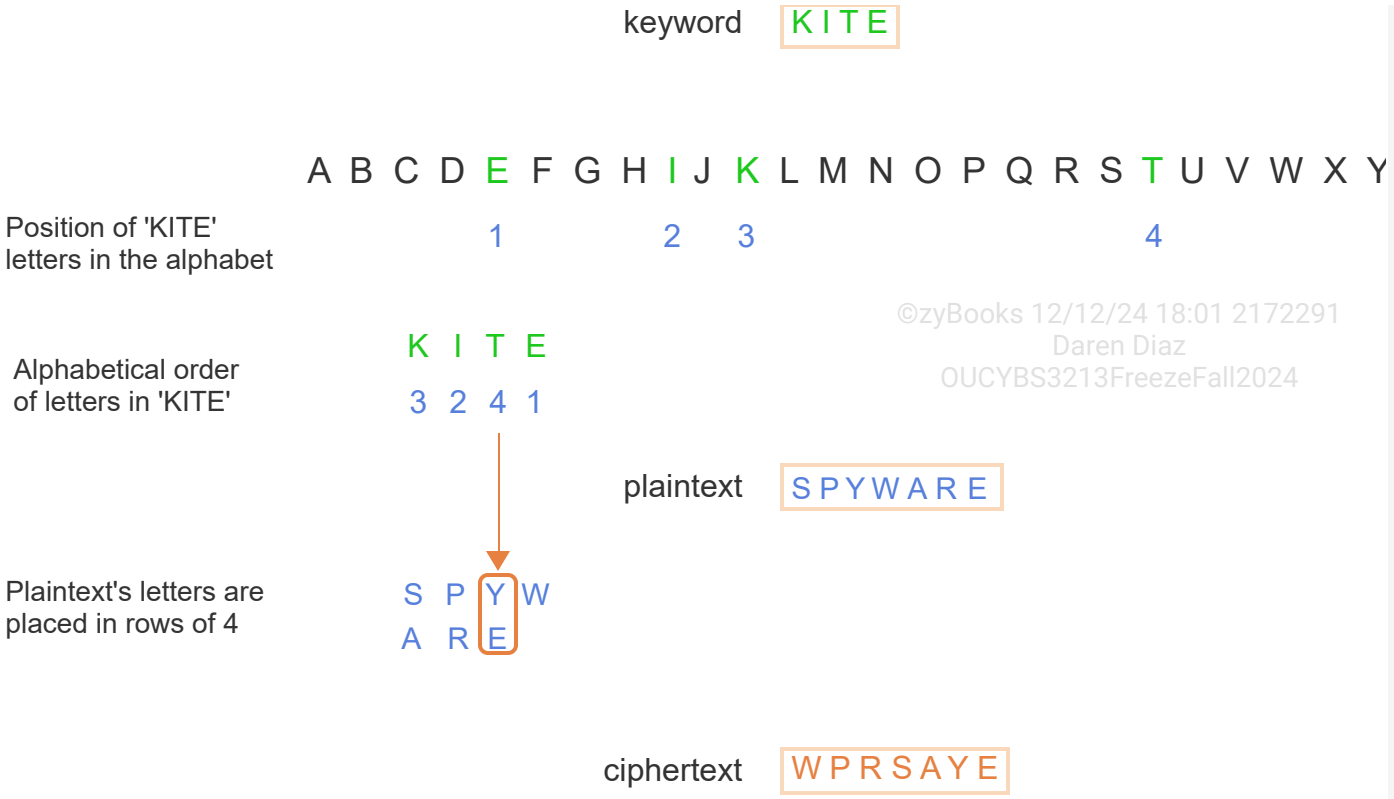
2) A cipher generates a ciphertext by replacing a plaintext's letters. Is the cipher a transposition cipher?
   - No
   - Yes

3) A transposition cipher reverses the order of letters in a plaintext. What is the ciphertext for 'WALK'?
   - 'KLAW'
   - 'KLWA'

Transposition cipher (columnar cipher)

A ***columnar cipher*** is a transposition cipher in which a plaintext's letters are placed in rows of length equal to the cipher's keyword length and then read in columns to generate the ciphertext. The alphabetical order of the letters in the cipher's keyword is the order in which the columns are read. Ex: A columnar cipher's keyword is "RUN". Since the length of the keyword is 3, the plaintext's letters are placed in rows of length 3. The alphabetical order of the letters in "RUN" is "2 3 1", because the letter 'N' is followed by the letter 'R' and then the letter 'U' in the alphabet. The third column's letters are read first, then the first column's letters, and finally the second column's letters.

PARTICIPATION ACTIVITY 3.2.7: Columnar transposition cipher.



Animation content:

Static figure: Keyword of KITE on the top, the alphabet underneath the keyword.
Step 1: The position of KITE's letters in the alphabet. The position of KITE letters are listed under the alphabet. Number 1 under the letter E, number 2 under the letter I, number 3 under the letter K, and number 4 under the letter T.
Step 2: The alphabetical order of letters in KITE (letter K is followed by the letter I, followed by letter T, followed by letter E in the alphabet). The alphabetical order of the letters in KITE is 3 2 4 1.
Step 3: The plaintext's letters are placed in rows of length equal to the length of the cipher's keyword which is 4. The plaintext is SPYWARE and the SPYWARE's letters are placed in rows of length 4. Column one is S A, column two is P R, column three is Y E, and column four is W.
Step 4: The alphabetical order of the letters in 'KITE' is the order in which the columns are read (fourth column is read first, second column is read second, first column is read third, third column is read fourth). The ciphertext is WPRSAYE.

Animation captions:

1. The position of 'KITE' letters in the alphabet.
2. The alphabetical order of letters in 'KITE' (letter 'K' is followed by the letter 'I', followed by letter 'T', followed by letter 'E' in the alphabet).
3. The plaintext's letters are placed in rows of length equal to the length of the cipher's keyword which is 4.

4. The alphabetical order of the letters in 'KITE' is the order in which the columns are read (fourth column is read first, second column is read second, first column is read third, third column is read fourth).

PARTICIPATION ACTIVITY 3.2.8: Columnar cipher.

1) A columnar transposition cipher's keyword is "DIGIT". What is the length of the rows used in the encryption process?
   - ○ 5
   - ○ 4

2) The length of the rows used in a columnar cipher encryption process is 3. Which one of the following words may be the encryption key?
   - ○ Golf
   - ○ Cat

3) What is the alphabetical order of letters in "spy"?
   - ○ 3 1 2
   - ○ 2 1 3

4) What is the ciphertext of "threat" using a columnar cipher with keyword "spy"?
   - ○ rthate
   - ○ hatert

CHALLENGE ACTIVITY 3.2.1: Historical cryptosystems.

581480.4344582.qx3zqy7

Start

What is the ciphertext of "software" using a shift cipher (key = 2)?

Ex: abcdef

What is the ciphertext of "software" using a Caesar cipher?

What is the ciphertext of "software" using a ROT13 cipher?

1 2 3

Check Next

3.3 Encryption

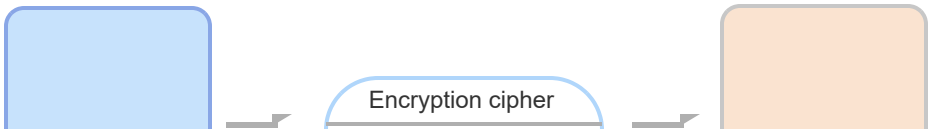
Encryption

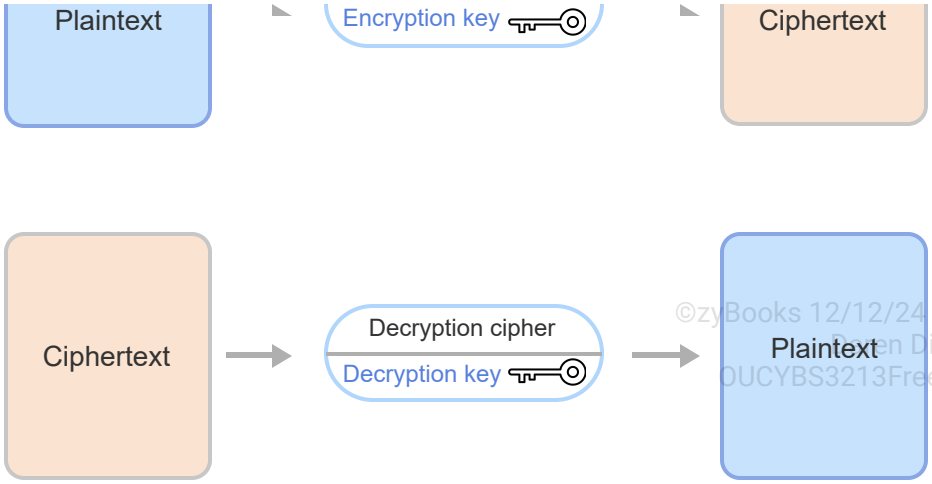
Encryption is the process of encoding or scrambling a message. ***Decryption*** is the process of decoding or unscrambling a message. The encryption and decryption process follow a specific sequence of instructions called an algorithm. A ***cipher*** is an encryption or decryption algorithm.

An encryption algorithm transforms a plaintext into a ciphertext using a key. A ***key*** is a string of bits. A ***key space*** is a set of all possible keys. A ***plaintext*** is an unencrypted or unscrambled message. A ***ciphertext*** is an encrypted or scrambled message. A decryption algorithm transforms a ciphertext into a plaintext using a decryption key.

Key length is the size of the encryption key measured in bits. Longer keys enhance security by increasing the number of possible combinations, thereby making the encryption more resilient against brute-force attacks and unauthorized decryption attempts. To enhance encryption strength for shorter or simpler keys, key stretching techniques are employed. ***Key stretching*** is the process of artificially increasing a key's length and complexity. Key stretching makes keys more resistant to brute-force attacks, even if initially weak.

PARTICIPATION ACTIVITY 3.3.1: Encryption and decryption.





Animation content:

Static image: A diagram with two rows. The first row starts with a box labeled "Plaintext." The Plaintext box points to an oval labeled "Encryption cipher" with a key icon and "Encryption key." The Encryption cipher oval points to a box labeled "Ciphertext." The second row starts with a box labeled "Ciphertext." The Ciphertext box points to an oval labeled "Decryption cipher" with a key icon and "Decryption key." The Decryption cipher oval points to a box labeled "Plaintext."

Animation captions:

1. An encryption cipher encrypts a plaintext with an encryption key to generate a ciphertext.
2. A decryption cipher decrypts a ciphertext with a decryption key to generate a plaintext.

PARTICIPATION ACTIVITY | 3.3.2: Cryptographic principles.

How to use this tool

Encryption **Key space** **Plaintext** **Ciphertext** **Cipher** **Key** **Decryption**

	The process of encoding or scrambling a message.
	The process of decoding or unscrambling a message.

	An unencrypted or unscrambled message.
	An encrypted or scrambled message.
	A string of bits.
	The set of all possible keys.
	An encryption or decryption algorithm.

Reset

PARTICIPATION ACTIVITY 3.3.3: Encryption.

1) What is the purpose of encryption?
   - To compress data
   - To permanently alter a message's content
   - To encode or scramble a message for security

2) What is the role of a cipher in encryption?
   - To decode messages only
   - To store encrypted messages
   - To serve as an encryption or decryption algorithm

3) How does increasing the key length affect an encryption method's security?
   - Reduces security by making the key easier to guess
   - Decreases the number of possible combinations for keys
   - Enhances security by increasing the number of possible combinations

4) What are the two required components for decrypting a ciphertext?

- Decryption algorithm and encryption key
- Decryption algorithm and decryption key
- Encryption algorithm and encryption key

5) What are the two required components for encrypting a plaintext?

- Decryption algorithm and encryption key
- Decryption algorithm and decryption key
- Encryption algorithm and encryption key

6) What is the downside to excessively long key lengths in a practical application environment?

- Reduce the encryption strength per bit of key length
- Slow down encryption/decryption, affecting system performance
- Make encryption algorithms less secure due to increased predictability

Properties of encryption algorithms

A secure encryption algorithm should have two properties:

- ***Confusion*** is a secure encryption algorithm property that ensures that changing a single bit of an encryption key impacts most of the ciphertext bits. The confusion property hides the relationship between a ciphertext and the encryption key.
- ***Diffusion*** is a secure encryption algorithm property that ensures that changing a single plaintext bit changes about half of the ciphertext bits and changing a single ciphertext bit changes about

half of the plaintext bits. The diffusion property hides the relationship between a plaintext and a ciphertext.

PARTICIPATION ACTIVITY 3.3.4: Properties of encryption algorithms.

For each of the following scenarios, which property or properties of a secure cipher is not met?

1) Alice can easily find the relationship between a ciphertext and the key used to generate the ciphertext.
   - Diffusion
   - Confusion
   - Confusion and diffusion

2) Alice can easily find the relationship between a ciphertext and the key used to generate the ciphertext, and the relationship between a ciphertext and the plaintext.
   - Diffusion
   - Confusion
   - Confusion and diffusion

3) Alice can easily find the relationship between a ciphertext and the plaintext.
   - Diffusion
   - Confusion
   - Confusion and diffusion

Kerckhoffs's principle

Kerckhoffs's principle states that the security of a cryptographic algorithm should depend on the secrecy of the key, not on the secrecy of the algorithm. In other words, how a message is encrypted can be known, but the key used to encrypt the message should be kept a secret. Kerckhoffs's principle is applied to all contemporary encryption algorithms. The details of the most widely used encryption algorithms are publicly known and have been subject to extensive cryptanalysis.

PARTICIPATION ACTIVITY 3.3.5: Kerckhoffs's principle.

Is the Kerckhoffs's principle followed in each of the following scenarios?

1) Alice and Bob use secret cryptographic algorithms, and keep the key secret.

   - Yes, because Alice and Bob use secret cryptographic algorithms.
   - Yes, because Alice and Bob keep the key secret.

2) Alice and Bob use publicly known and secure cryptographic algorithms. Alice sends the encryption key to Bob using an insecure email service.

   - No, because Alice used an insecure email service to send the key to Bob.
   - No, because Alice and Bob use publicly known and secure cryptographic algorithms.

3) Alice and Bob use publicly known and secure cryptographic algorithms, and keep the key secret.

   - Yes, because Alice and Bob use publicly known and secure cryptographic algorithms.
   - Yes, because Alice and Bob keep the key secret.

3.4 Symmetric encryption: Stream cipher

Symmetric encryption

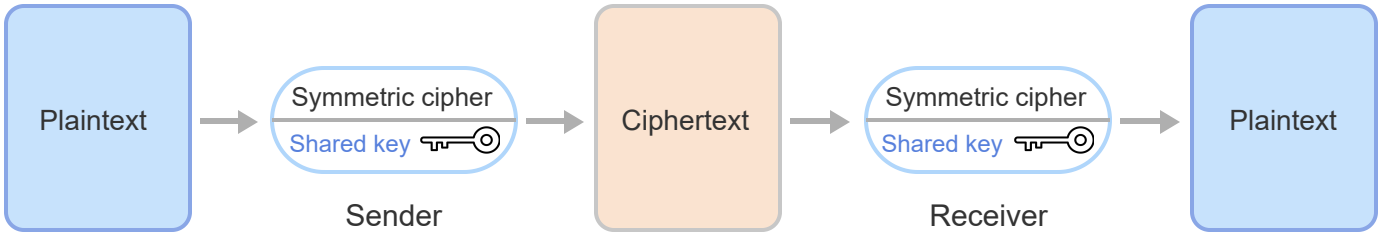
A ***symmetric encryption*** is an encryption algorithm that uses the same key to encrypt and decrypt data. A ***symmetric key*** or ***secret key*** is the key used in symmetric encryption.Two classes of symmetric encryption exist:

- A ***stream cipher*** is a symmetric encryption algorithm that encrypts data one bit at a time.

- A ***block cipher*** is a symmetric encryption algorithm that encrypts data one block at a time.

Symmetric encryption is faster and more efficient than asymmetric encryption since symmetric encryption uses shorter keys and simpler operations. Symmetric encryption is used in many cryptographic protocols such as HTTPS and IPSec.

PARTICIPATION ACTIVITY 3.4.1: Symmetric encryption.



Animation content:

Static image: A box labeled "Plaintext" with an arrow pointing to an oval labeled "Sender." The Sender oval contains "Symmetric cipher," a key icon, and "Shared key." The Sender oval has an arrow pointing to a box labeled "Ciphertext." The Ciphertext box has an arrow pointing to an oval labeled Receiver. The Receiver oval contains "Symmetric cipher," a key icon, and "Shared key." The Receiver oval has an arrow pointing to another box labeled "Plaintext."

Animation captions:

1. The data sender and receiver share a symmetric key.
2. The data sender encrypts a plaintext using the shared key to generate the ciphertext.
3. The data receiver decrypts the ciphertext using the shared key to generate the plaintext.

PARTICIPATION ACTIVITY 3.4.2: Symmetric encryption.

How to use this tool

Stream cipher **Secret key** **Symmetric encryption** **Block cipher**

	An encryption algorithm that uses the same key to encrypt and decrypt data.
	The key used in symmetric encryption.
	An encryption algorithm that encrypts data one bit at a time.
	An encryption algorithm that encrypts data one block at a time.

Reset

PARTICIPATION ACTIVITY 3.4.3: Symmetric encryption.

1) In symmetric encryption, encrypted data can be decrypted by any key.
   - ○ True
   - ○ False

2) A data sender should share the encryption key with the data receiver before encrypting the data.
   - ○ True
   - ○ False

3) How can Alice and Bob exchange a secure message using symmetric encryption?
   - ○ Alice and Bob use public keys for encryption and decryption

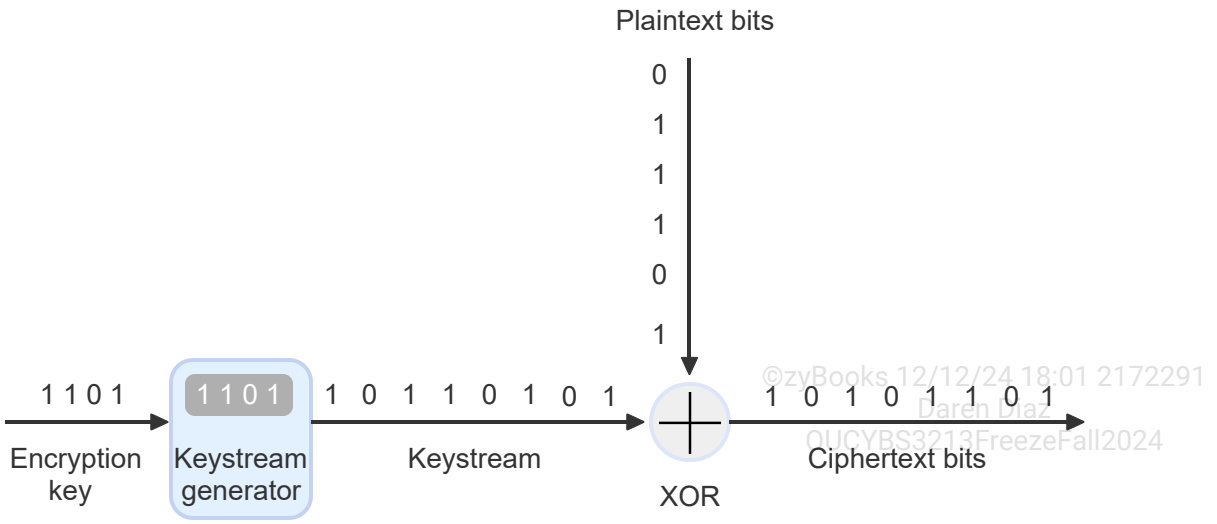
- Alice and Bob cannot securely exchange data using symmetric encryption
- Alice and Bob use a shared secret key to encrypt and decrypt data

Stream cipher

A stream cipher encrypts data one bit at a time. A stream cipher converts an encryption key into a keystream. A ***keystream*** is a continuous bit stream. A ***keystream generator*** is an algorithm that creates a keystream. A data sender and receiver use the same key and keystream generator to encrypt or decrypt data.

A stream cipher is computationally efficient because the encryption and decryption consist of a simple operation, such as exclusive or (XOR). Since a stream cipher encrypts data one bit at a time, an error in encryption does not propagate; a one bit error in encryption results in a one bit error in decryption. A stream cipher is commonly used in communication applications that require fast encryption and decryption of a data stream. Ex: The ***A5/1*** is a stream cipher used for encrypting data exchanged between a mobile phone and a base station.

PARTICIPATION ACTIVITY 3.4.4: Stream cipher.



Animation content:

Static image: A diagram showing a stream cipher starts on the left with four bits above an arrow labeled "Encryption key" pointing to the right. The arrow points to a blue box with four bits labeled "Keystream generator." The Keystream generator box has an arrow with eight bits labeled "Keystream" pointing to an icon labeled "XOR." A second arrow labeled "Plaintext bits" points to the XOR icon and has six bits. The XOR icon has an arrow pointing to the right labeled "Ciphertext bits" with eight bits.
Step 1: A keystream generator uses an encryption key to generate a keystream.
An arrow and a blue box appear. The arrow is labeled "Encryption key" and has four bits. The blue box is labeled "Keystream generator." A copy of the four bits moves from the arrow into the blue Keystream generator box. An arrow labeled "Keystream" appears pointing from the Keystream generator box to the right. Eight bits appear above the Keystream arrow.
Step 2: An exclusive or (XOR) operation is performed on keystream bits and plaintext bits.
An arrow labeled "Plaintext bits" appears from the top pointing downward. Six bits appear next to the arrow and an icon labeled "XOR" appears below the Plaintext bits arrow and to the right of the Keystream arrow.
Step 3: A stream cipher encrypts plaintext bits with a keystream one bit at a time to generate ciphertext bits.
An arrow labeled "Ciphertext bits" appears pointing from the XOR icon to the right. The rightmost bit above the Keystream arrow and the bottom bit next to the Plaintext bits arrow moves onto the XOR icon. The bits disappear, and a single bit appears and moves above the Ciphertext bits arrow. The process repeats with a single bit from the Keystream arrow and a single bit from the Plaintext bits arrow until the Ciphertext bits arrow has eight bits.

Animation captions:

1. A keystream generator uses an encryption key to generate a keystream.
2. An exclusive or (XOR) operation is performed on keystream bits and plaintext bits.
3. A stream cipher encrypts plaintext bits with a keystream one bit at a time to generate ciphertext bits.

PARTICIPATION ACTIVITY 3.4.5: Stream ciphers.

How to use this tool

Keystream **Stream cipher** **Keystream generator**

	A symmetric encryption algorithm that encrypts data one bit at a time.
	A continuous bit stream that is generated based on an encryption key.
	An algorithm that outputs a continuous bit stream given an input key.

Reset

PARTICIPATION ACTIVITY 3.4.6: Stream cipher.

1) A cipher that encrypts a message 256-bits at a time is a stream cipher.
   - True
   - False

2) In a stream cipher, errors in 5 bits during encryption result in errors in 6 bits of a ciphertext.
   - True
   - False

3) A data receiver uses the same key that the data sender uses to encrypt data, but not the same keystream generator. Can the data receiver decrypt a ciphertext?
   - Yes. A receiver only needs an encryption key to decrypt the data.
   - No. A receiver should use the same keystream generator and the same key that the sender used to encrypt data.
   - Yes. All stream ciphers use the same keystream generator.

4) A deterministic generator is an algorithm that outputs the same bit sequence given the same input key. Is a stream cipher's keystream generator a deterministic generator?

- Yes. The bit sequence in a keystream should be known during encryption.
- Yes. The data receiver requires the same bit sequence in a keystream to correctly decrypt data.
- No. A stream cipher's keystream generator should output a different keystream given the same input key to increase randomness.

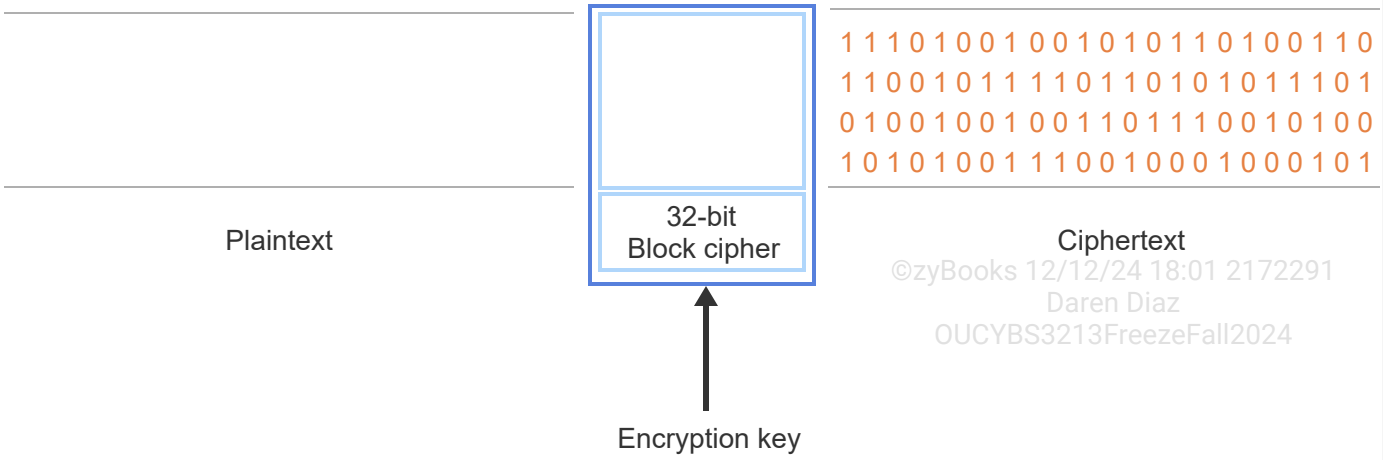
3.5 Symmetric encryption: Block cipher

Block cipher

A ***block cipher*** is a symmetric cipher that encrypts a data block of fixed size. A block cipher only encrypts a single data block at a time. Ex: International Data Encryption Algorithm (IDEA) is a block cipher that encrypts data in 64-bit blocks. Block ciphers are used in other cryptographic primitives such as cryptographic hash functions and message authentication code (MAC).

For a variable-length plaintext, the data must first be partitioned into separate blocks. If the length of a plaintext is not a multiple of a block cipher's block size, a plaintext's last block is filled out with random bits to make a full block. The process of adding bits to a plaintext's last block is called ***padding***. Ex: If a block cipher's block size is 64 bits, an 84-bit plaintext is split into one block of 64 bits and a second block of 20 bits. The second block is padded with '64 - 20' or 44 random bits to create a 64-bit block.

PARTICIPATION ACTIVITY 3.5.1: Block cipher.



Animation content:

Static image: A diagram showing a 32-bit block cipher. The left side of the diagram is labeled "Plaintext." A box in the middle is labeled "Encryption key" and contains an empty box and a box labeled "32-bit Block cipher." The right side of the diagram is labeled "Ciphertext" and contains four lines with 24 bits each.
Step 1: A block cipher uses the encryption key to encrypt a block of plaintext bits.
Four lines of text appear on the left in the "Plaintext" section. The first two lines contain 24 bits. The third line contains four bits, four spaces, and then 16 more bits. The fourth line contains eight spaces and then 16 bits. The plaintext shifts to the right eight spaces so that the right eight bits of each line are in the box labeled "32-bit Block cipher." "Encryption key" appears below the box, the right eight bits of the plaintext lines change to different values.
Step 2: The output of a block cipher is a block of encrypted ciphertext bits.
All of the text shifts to the right eight spaces. The right eight characters of each line move to the "Ciphertext" section. The middle eight bits of the lines of plaintext move into the "32-bit Block cipher" box. The bits in the Block cipher box change to different values. The text shifts to the right so that the changed values are now in the "Ciphertext" section. The first eight bits of each line of plaintext are in the Block cipher box. The last line is completely blank and the third line has four blank spaces.
Step 3: If a plaintext's last block is smaller than a block cipher's block size, a block cipher pads the block with random bits to make a full block.
Binary bits are added to the blank spaces in the third and fourth lines of the plaintext within the Block cipher box.
Step 4: All the ciphertext blocks have the same size as the block cipher's block size.
The bits within the Block cipher box change values. The text shifts to the right so that all of the text is in the "Ciphertext" section.

Animation captions:

1. A block cipher uses the encryption key to encrypt a block of plaintext bits.

2. The output of a block cipher is a block of encrypted ciphertext bits.
3. If a plaintext's last block is smaller than a block cipher's block size, a block cipher pads the block with random bits to make a full block.
4. All the ciphertext blocks have the same size as the block cipher's block size.

A block cipher is more efficient than a stream cipher when the size of a plaintext is known before the encryption process begins. A stream cipher is more efficient than a block cipher when the size of a plaintext is unknown or a plaintext is in a continuous stream.

PARTICIPATION ACTIVITY 3.5.2: Block cipher.

How to use this tool

IDEA **Padding** **Block cipher**

	A block cipher which encrypts a plaintext 64-bit block at a time.
	A symmetric encryption algorithm which encrypts a data block of fixed size.
	The process of filling out a plaintext's last block with random bits to create a full block

Reset

PARTICIPATION ACTIVITY 3.5.3: Block cipher.

1) Is a data receiver required to use the same block cipher key a data sender used to encrypt a ciphertext?
   - ○ Yes
   - ○ No

2) A cipher encrypts data 64-bits at a time. Is the cipher a block cipher?

- Yes
- No

3) Which type of cipher is more efficient for encrypting a live video stream?
   - Stream cipher
   - Block cipher

4) A 128-bit block cipher is used to encrypt a 300-bit plaintext. How many ciphertext blocks are generated by the cipher?
   - 3 blocks
   - 2 blocks

5) A 128-bit block cipher is used to encrypt a 148-bit plaintext. How many padding bits does the block cipher add to the plaintext's last block?
   - 108 bits
   - 128 bits

Modes of operation

A ***mode of operation*** specifies how a block cipher applies a single-block operation to a plaintext larger than the block cipher's block size. Five common modes of operation exist:

- In ***Electronic Code Book (ECB)*** mode each plaintext block is encrypted separately to generate a ciphertext block. In ECB mode a ciphertext only depends on an encryption key and a plaintext. Identical plaintext blocks are encrypted into identical ciphertext blocks. Since each plaintext block is encrypted separately, errors in one plaintext block results in errors in only one ciphertext block.
- In ***Cipher Block Chaining (CBC)*** mode each plaintext block is XORed with the previous ciphertext block before being encrypted. As a result, all the ciphertext blocks are computationally *chained* together. To create a unique ciphertext every time the encryption is performed, an ***initialization vector (IV)*** is used in the encryption process of a plaintext's first block. An initialization vector is a randomly generated sequence of bits of size equal to the cipher's block size. Since the ciphertext blocks are computationally chained, errors propagate in CBC mode. An error in one ciphertext block results in errors in all the subsequent ciphertext blocks.
- In ***Cipher Feedback (CFB)*** mode a block cipher operates as a stream cipher. Similar to CBC mode, an IV is used in the encryption process of the first plaintext block and ciphertext blocks are computationally chained. Each plaintext block is encrypted and XORed with the previous

ciphertext block to produce the current ciphertext block. The encryption and decryption processes are the same in CFB mode. Similar to CBC mode, errors propagate in CFB mode.

- In ***Output Feedback (OFB)*** mode a block cipher generates keystream blocks which are XORed with the plaintext blocks to create the ciphertext blocks. Chaining dependencies do not exist in OFB mode because each block is created independently of plaintext and ciphertext blocks. In OFB mode encryption and decryption processes are the same and errors do not propagate.
- In ***Counter (CTR)*** mode a block cipher operates as a stream cipher. A data sender and receiver use a synchronized counter which computes a new shared value each time a ciphertext block is exchanged. The CTR mode has positional dependency because a ciphertext block depends on the position of the current plaintext block. Since a ciphertext block does not depend on any other blocks, errors do not propagate in CTR mode.

PARTICIPATION ACTIVITY | 3.5.4: Modes of operation.

How to use this tool

Counter (CTR) **Cipher Feedback (CFB)** **Output Feedback (OFB)**

Electronic Code Book (ECB) **Cipher Block Chaining (CBC)**

	Each plaintext block is XORed with the previous ciphertext block before being encrypted.
	Each plaintext block is encrypted separately.
	A ciphertext block depends on the position of the current plaintext block.
	Each plaintext block is encrypted and XORed with the previous ciphertext block.
	Each block is created independently of plaintext and ciphertext blocks.

Reset

PARTICIPATION ACTIVITY 3.5.5: Modes of operation.

1) In which mode of operation each plaintext block is XORed with the encrypted previous block?

Check Show answer

2) In which mode of operation each plaintext block is encrypted separately to generate a block of ciphertext?

Check Show answer

3) In which mode of operation an initialization vector (IV) is used in the encryption process of a plaintext's first block?

Check Show answer

4) In which mode of operation a synchronized counter is used by data sender and receiver?

Check Show answer

5) In which mode of operation each ciphertext block is created independently of plaintext blocks and other ciphertext blocks?

Check Show answer

Data Encryption Standard (DES) and 3DES

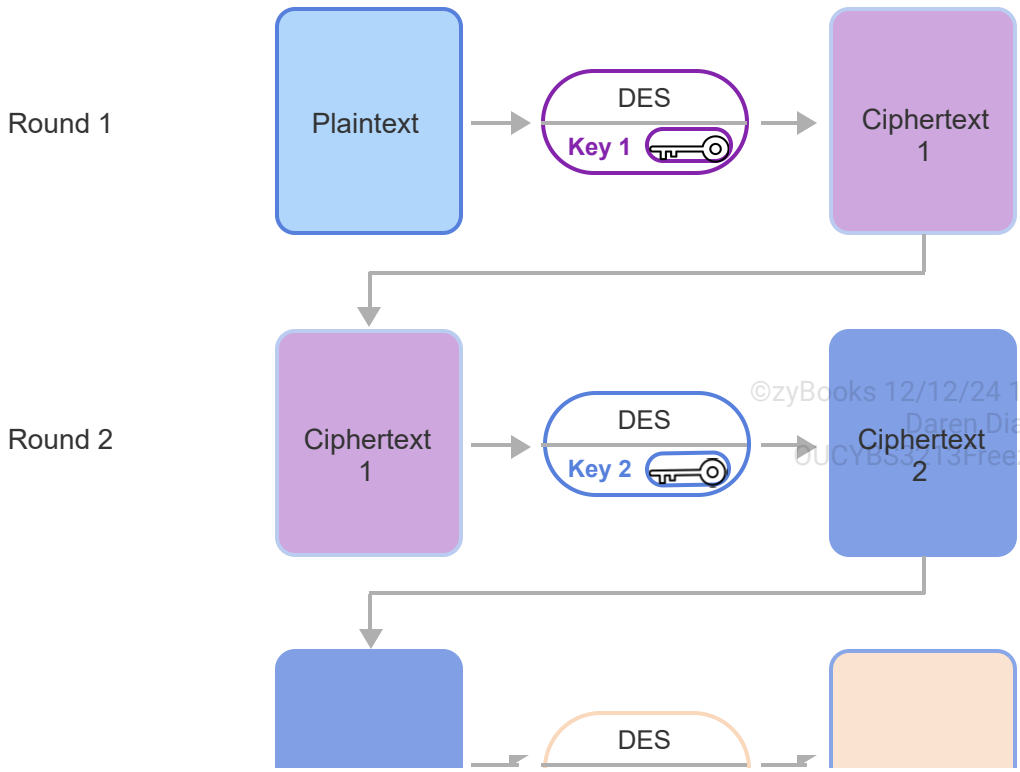
Data Encryption Standard (DES) is a symmetric block cipher which encrypts data in 64-bit blocks. DES uses 56-bit keys. DES is not considered a secure encryption algorithm because a 56-bit key can be broken with modern cryptanalytic techniques. ***Triple Data Encryption Standard (TDES)***, or ***3DES***, is a symmetric block cipher which applies DES three consecutive times, or rounds, to each 64-bit block. One, two, or three identical or different keys can be used at each of the three DES rounds.

Three keying options exist in 3DES:

- Keying option 1: Three independent 56-bit keys are used in each DES round, for a total of 3 x 56 = 168 independent key bits. Keying option 1 is the most secure keying option.
- Keying option 2: Two independent 56-bit keys are used in each DES round, for a total of 2 x 56 = 112 independent key bits.
- Keying option 3: Three identical 56-bit keys are used in each DES round. Keying option 3 is the most insecure keying option and is for providing backward compatibility with DES.

3DES was designed to improve DES by increasing the encryption key size without the need to design a new symmetric block cipher.

PARTICIPATION ACTIVITY | 3.5.6: 3DES.



Animation content:

Static image: A diagram showing three rounds of encryption. Round one starts with a box labeled "Plaintext." The Plaintext box has an arrow pointing to a purple oval labeled "DES" with a key outlined in purple and "Key 1." The purple DES oval has an arrow pointing to a purple box labeled "Ciphertext 1." The "Ciphertext 1" box in round one has an arrow pointing to the first box in round two, labeled "Ciphertext 1." The Round 2 Ciphertext 1 box has an arrow pointing to a blue oval labeled "DES" with a key outlined in blue and "Key 2." The blue "DES" oval has an arrow pointing to a blue box labeled "Ciphertext 2." The "Ciphertext 2" box in round two has an arrow pointing to the first box in round three, labeled "Ciphertext 2." The Round 3 Ciphertext 2 box has an arrow pointing to an orange oval labeled "DES" with a key outlined in orange and "Key 3." The orange oval has an arrow pointing to an orange box labeled "Ciphertext 3."

Animation captions:

1. 3DES applies DES three consecutive times, or rounds, to each 64-bit block of plaintext. Key 1 is used in the first DES round.
2. Ciphertext generated from the first DES round is used as input to DES in the second round with Key 2. Key 2 can be different or identical to Key 1.
3. Ciphertext generated from second DES round is used as input to DES in the third round with Key 3. Key 3 can be different or identical to Key 1 and Key 2.

PARTICIPATION ACTIVITY 3.5.7: DES and 3DES.

1) 3DES block size is 56 bits.
   - True
   - False

2) 3DES is always more secure than DES.
   - True
   - False

3) 3DES keying option 2 is the most secure keying option.

- True
- False

Advanced Encryption Standard (AES)

Advanced encryption standard (AES) is a symmetric block cipher which encrypts data in 128-bit blocks. AES uses 128, 192, or 256-bit keys. AES is the most widely used block cipher. Ex: AES is used by VPNs to create encrypted tunnels, by wireless networks to encrypt data in transit, by data storage devices to encrypt data at rest, and by HTTPS to secure communications on the Internet.

An ***encryption round*** is a set of operations that are consecutively applied to a block of plaintext bits during an encryption process. The number of encryption rounds in AES is determined by the encryption key length. A longer key corresponds to a higher number of encryption rounds and a larger keyspace, both of which make the cipher more resistant to brute force attacks.

Table 3.5.1: Advanced encryption standard (AES).

Version	Key length (bits)	Encryption rounds	Keyspace (number of possible keys)
AES-128	128	10	$2^{128} \approx 3.4 \times 10^{38}$
AES-192	192	12	$2^{192} \approx 6.2 \times 10^{57}$
AES-256	256	14	$2^{256} \approx 1.1 \times 10^{77}$

PARTICIPATION ACTIVITY 3.5.8: DES, 3DES, and AES.

How to use this tool

3DES AES DES

	A symmetric block cipher which consecutively applies another encryption algorithm three times to each 64-bit block.
	A symmetric block cipher which encrypts data in 64-bit blocks.
	A symmetric block cipher which encrypts data in 128-bit blocks.

Reset

PARTICIPATION ACTIVITY 3.5.9: Advanced encryption standard (AES).

1) Which one of the following key lengths can be used in AES?
   - 56 bits
   - 256 bits
   - 1024 bits

2) Why is AES-192 more secure than AES-128?
   - Because AES-192 has a lower number of encryption rounds
   - Because AES-192 has a larger keyspace
   - Because AES-192 uses 14 encryption rounds

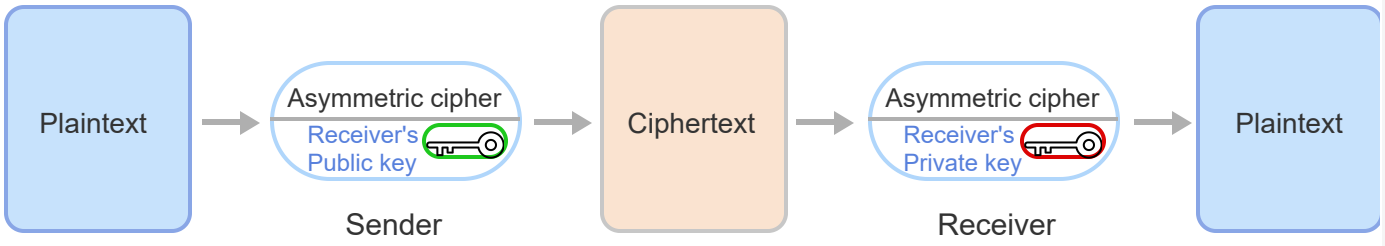
3) Why a mobile device with limited battery capacity may use AES-128 instead of AES-256?
   - Because AES-128 uses a larger key than AES-256
   - Because AES-128 performs more encryption rounds than AES-256
   - Because AES-128 encryption uses less energy than AES-256

3.6 Asymmetric encryption

Asymmetric encryption

In ***asymmetric encryption*** two different, but mathematically related keys are used to encrypt and decrypt a message. Asymmetric encryption is also known as ***public-key encryption***, and the two related keys are called a public key and a private key, known together as a ***key-pair***. A ***public key*** is not secret and is openly available. A ***private key*** is secret and is only available to the key-pair owner. A message encrypted with a public key can only be decrypted with the related private key. The key-pair enables two entities to securely exchange a message. A message sender encrypts a message with a message receiver's public key. A message receiver decrypts a message with the related private key.

PARTICIPATION ACTIVITY 3.6.1: Asymmetric encryption.



Animation content:

Static image: A box labeled "Plaintext" with an arrow pointing to an oval labeled "Sender." The Sender oval contains "Asymmetric cipher," a key icon circled in green, and "Receiver's Public key." The Sender oval has an arrow pointing to a box labeled "Ciphertext." The Ciphertext box has an arrow pointing to an oval labeled Receiver. The Receiver oval contains "Asymmetric cipher," a key icon circled in red, and "Receiver's private key." The Receiver oval has an arrow pointing to another box labeled "Plaintext."

Animation captions:

1. A message sender obtains a message receiver's public key.
2. A message sender encrypts a plaintext using a message receiver's public key to generate a ciphertext.
3. A message receiver decrypts the ciphertext using the receiver's private key to generate a plaintext.

PARTICIPATION ACTIVITY 3.6.2: Asymmetric encryption.

How to use this tool

Public and private keys **Public key** **Private key**

	The two related keys in asymmetric encryption.
	A key that is not secret.
	A key which should be kept secret.

Reset

PARTICIPATION ACTIVITY 3.6.3: Asymmetric encryption.

1) How can Alice send a secure message to Bob using asymmetric encryption?

- Alice encrypts a message with Bob's private key
- Alice encrypts the message with Bob's public key
- Alice encrypts the message with Alice's own public key

2) In asymmetric encryption, a private key can be used to both encrypt and decrypt a message.

- False

- True

3) If a private key is lost, the key-pair owner can no longer decrypt a message encrypted with the owner's public key.
   - False
   - True

Asymmetric algorithms

Asymmetric encryption is based on the properties of one-way functions. A ***one-way function*** is a function that is easy to compute in one direction, but difficult to compute in the opposite direction. A ***trapdoor one-way function*** is a one-way function that is easy to compute in both directions only with specific knowledge of a function's input or output. The specific knowledge is called a ***trapdoor***. Asymmetric algorithms are based on two classes of trapdoor one-way functions: integer factorization and discrete logarithm.

PARTICIPATION ACTIVITY 3.6.4: Asymmetric algorithms.

1) A trapdoor one-way function is a function which is easy to compute in both directions without having any specific knowledge of a function's input or output.
   - False
   - True

2) The two classes of trapdoor one-way functions are discrete logarithm and polynomial.
   - False
   - True

3) A function which multiplies an input value by 5 to generate an output value, is a one-way function.
   - False
   - True

Three asymmetric encryption algorithms are in use:

- RSA
  The ***RSA algorithm*** uses the mathematical properties of prime numbers in generating public and private key pairs. The security of the RSA algorithm is based on the difficulty of factoring large prime numbers (typically several hundred digits in length). Encryption and decryption using the RSA algorithm is slow and computationally expensive. The RSA algorithm is mostly used to set up a secure communication link between two parties for the purpose of exchanging a session key. A ***session key*** is a symmetric key that is only used for the duration of a single communication session. The RSA algorithm is used in asymmetric encryption and digital signatures.
- ElGamal
  The ***ElGamal algorithm*** uses discrete logarithms for generating public and private key pairs. The security of the ElGamal algorithm is based on the difficulty of solving discrete logarithms. The ElGamal encryption algorithm is probabilistic, meaning that a plaintext may be encrypted to different ciphertexts. The main disadvantage of the ElGamal encryption is that the length of a ciphertext generated by the algorithm is twice the length of a plaintext. The doubling of a ciphertext size makes the use of the algorithm inefficient in resource-limited devices and in communications over low bandwidth networks. The ElGamal algorithm is used in asymmetric encryption and digital signatures.
- Elliptic Curve Cryptography (ECC)
  The ***Elliptic Curve Cryptography (ECC) algorithm*** uses elliptic curves for generating public and private key pairs. An elliptic curve is the set of points that satisfy a specific mathematical equation. The security of the ECC algorithm is based on the difficulty of solving Elliptic Curve Discrete Logarithm Problem (ECDLP). ECC provides the same level of security as RSA but with smaller key sizes. ECC is computationally more efficient than RSA and is used in resource-limited devices such as mobile phones. The ECC algorithm is used in asymmetric encryption, digital signatures, and key exchange.

Table 3.6.1: Asymmetric algorithms.

Algorithm name	Computational efficiency	Power consumption	Typical key size (bits)	Basis of security
RSA	Low	High	1024	Difficulty of factoring the product of two large prime numbers
ElGamal	Medium	Medium	1024	Difficulty of computing discrete logarithms

ECC	High	Low	160	Difficulty of computing elliptic curve discrete logarithms

PARTICIPATION ACTIVITY 3.6.5: Asymmetric algorithms.

How to use this tool

ElGamal **ECC** **RSA**

	An asymmetric encryption algorithm based on the mathematical properties of prime numbers.
	An asymmetric encryption algorithm based on the mathematical properties of discrete logarithms.
	An asymmetric encryption algorithm based on the mathematical properties of elliptic curves.

Reset

PARTICIPATION ACTIVITY 3.6.6: Asymmetric algorithms.

1) Which asymmetric encryption algorithm is ideal for use in a battery powered Internet of Things (IoT) device?

- RSA
- ECC
- ElGamal

2) Which asymmetric encryption algorithm should *not* be used in a

device with limited memory?

- RSA
- ECC
- ElGamal

3) Which asymmetric encryption algorithm can provide a high level of security with a small key size?

- RSA
- ECC
- ElGamal

Asymmetric algorithms for key establishment

Asymmetric encryption is also used for establishing symmetric keys. The ***Diffie-Hellman (DH) algorithm*** is a key exchange algorithm used for generating a shared symmetric key between two parties communicating over an insecure network such as the Internet. The DH algorithm can generate both a static key and an ephemeral key. A ***static key*** is a long-term key intended to be used over an extended time-period. The private key of a public-private key pair is a static key. An ***ephemeral key*** is a key which is used in only a single transaction. An ephemeral key is generated for each execution of a key-establishment process. Two DH methods use ephemeral keys:

- Diffie-Hellman Ephemeral (DHE) or Ephemeral Diffie-Hellman (EDH)
- Elliptic Curve Diffie-Hellman Ephemeral (ECDHE) which uses ephemeral keys generated by Elliptic Curve Cryptography (ECC)

The DHE and ECDHE provide for perfect forward secrecy. The ***perfect forward secrecy (PFS)***, or ***forward secrecy (FS)***, is a property of a key exchange algorithm which provides the assurance that a session key will not be compromised if a static key (long-term key) used for generating a session key is compromised in the future.

PARTICIPATION ACTIVITY 3.6.7: Asymmetric algorithms for key establishment.

How to use this tool

Ephemeral key **Static key** **Session key**

A key generated for each execution of a key-establishment process.

	A long-term key intended to be used over an extended time-period.
	A symmetric key that is only used for the duration of a single communication session.

Reset

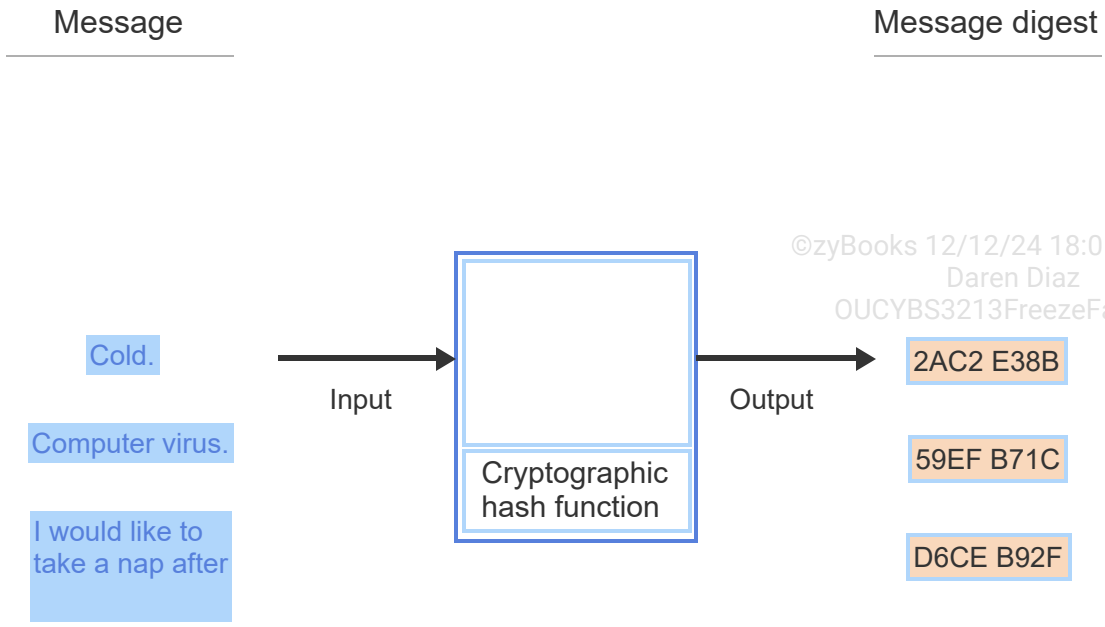
3.7 Cryptographic hash functions

Cryptographic hash functions

A ***cryptographic hash function*** is a function which maps a variable length string, or ***message***, to a fixed-size value, or ***message digest***. A message digest is also called a ***hash***. Ex: The MD5 is a cryptographic hash function that maps a message to a 128-bit hash.

A cryptographic hash function has many applications in information security, including digital signatures and message authentication code (MAC).

PARTICIPATION ACTIVITY 3.7.1: Cryptographic hash function.



coming home.

Animation content:

Static image: A diagram representing a cryptographic hash function. A list on the left side labeled "Message" contains "Cold," "Computer virus," and "I would like to take a nap after coming home." An arrow labeled "Input" points from the message list to a box labeled "Cryptographic hash function." An arrow labeled "Output" points from the Cryptographic hash function box to a list on the right side labeled "Message digest." The Message digest list contains "2AC2 E38B," "59EF B71C," and "D6CE B92F."
Step 1: The output of a hash function is a fixed-size string or message digest.
A copy of "I would like to take a nap after coming home" moves from the Message list into the Cryptographic hash function box. "I would like to take a nap after coming home" disappears. "D6CE B92F" appears in the Cryptographic hash function box and moves to the right under the "Message digest" list heading.
Step 2: The message digest length is the same for all messages.
A copy of "Computer virus" moves from the Message list into the Cryptographic hash function box. "Computer virus" disappears. "59EF B71C" appears in the Cryptographic hash function box and moves to the right under the "Message digest" list heading.
Step 3: A message's length does not impact a message digest's length.
A copy of "Cold" moves from the Message list into the Cryptographic hash function box. "Cold" disappears. "2AC2 E38B" appears in the Cryptographic hash function box and moves to the right under the "Message digest" list heading.

Animation captions:

1. The output of a hash function is a fixed-size string or message digest.
2. The message digest length is the same for all messages.
3. A message's length does not impact a message digest's length.

PARTICIPATION ACTIVITY 3.7.2: Cryptographic hash functions.

1) What is the digest length of MD5 for the word "computer"?

   - ○ 128 bits

- 256 bits
- 512 bits

2) What is the digest length of MD5 for the letter "K"?

- 128 bits
- 256 bits
- 512 bits

3) SHA-512 is a cryptographic hash function that outputs a 512-bit digest. The output of MD5 for the word "monitor" is used as an input to SHA-512. What is the length of the SHA-512 output?

- 128 bits
- 256 bits
- 512 bits

Properties of cryptographic hash functions

A cryptographic hash function has three properties:

- ***Collision resistance***: Finding two different inputs to a cryptographic hash function that have the same digest is computationally infeasible.
- ***Preimage resistance***: Finding the original message from a message digest is computationally infeasible. This property is also called the ***one-way property***.
- ***Second preimage resistance***: Finding a second input that has the same digest as any other specified input is computationally infeasible.

The avalanche effect is a desirable property of a cryptographic hash function. The ***avalanche effect*** property in a cryptographic hash function means that a small change to a message should result in a digest that is significantly different and uncorrelated with the digest of the original message.

PARTICIPATION ACTIVITY 3.7.3: Properties of cryptographic hash functions.

How to use this tool

Avalanche effect property **Second preimage resistance property**

Collision resistance property | Preimage resistance property

	Finding two different inputs to a cryptographic hash function that have the same digest is computationally infeasible.
	A small change to a message should result in a digest that is significantly different and uncorrelated with the digest of the original message.
	Finding the original message from a message digest is computationally infeasible.
	Finding a second input that has the same digest as any other specified input is computationally infeasible.

Reset

PARTICIPATION ACTIVITY 3.7.4: Cryptographic hash functions.

1) The Secure Hash Algorithm 1 (SHA-1) may generate the same message digest for different messages. Why is SHA-1 no longer considered a secure cryptographic hash function?

- The SHA-1 generates variable length message digests.
- The SHA-1 only generates message digests for fixed-size messages.
- The SHA-1 is not collision resistant.

2) Which cryptographic hash function property is not satisfied if finding the original message from a message digest is easy?

- Collision resistance
- Preimage resistance
- Second preimage resistance

3) Which cryptographic hash function property is not satisfied if finding a second input that has the same digest as any other specified input is easy?

- Collision resistance
- Preimage resistance
- Second preimage resistance

Collision attacks

*A collision, or hash collision, occurs when a hash function produces the same message digest for two different messages. A **collision attack** is an attack that attempts to find two different input messages to a hash function that produce the same message digest. Ex: A software package and the software package's hash are published so that a user can verify the software package's authenticity. An attacker attempts to create malware with the same hash as the software package so that a target believes the malware is authentic.*

*The birthday paradox is the idea that, given two groups of people, someone from the first group has the same birthday as someone from the second group. A **birthday attack** is a collision attack based on the birthday paradox. A birthday attack increases the chances of finding a collision by creating two groups of messages and searching for some message in the first group that shares a message digest with some message in the second group.*

3.8 Message Authentication Code (MAC)

Message Authentication Code (MAC)

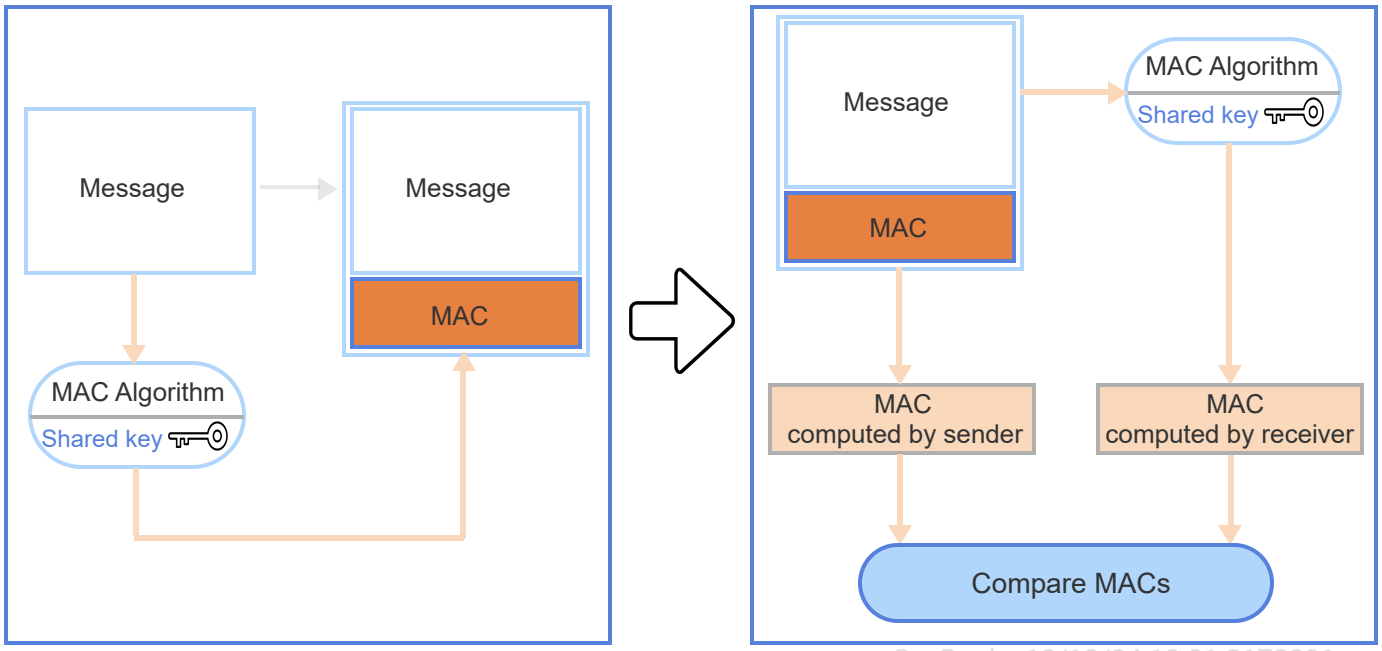
A ***message authentication code (MAC)*** is a cryptographic primitive used for verifying data integrity and authenticity. A MAC is computed by applying a MAC algorithm to a message in combination with a symmetric key. A MAC is also called a ***tag***.

A message sender computes the MAC for a message using a symmetric key shared with a message receiver. A message sender appends the MAC to a message before sending a message. A message receiver performs a two step process:

1. Computes the MAC for the message.
2. Compares the computed MAC with the MAC received with the message.

If the computed MAC is equal to the MAC received with the message, the message receiver is assured that the message was sent by the sender and the message was not modified in transit.

PARTICIPATION ACTIVITY 3.8.1: Message Authentication Code (MAC).



Animation content:

Static image: Two boxes labeled "Sender" and "Receiver" with an arrow pointing from the Sender box to the Receiver box. The Sender box has a diagram starting with a box labeled "Message." "Message" has an arrow pointing to an oval labeled "MAC Algorithm" with a key icon and "Shared key." "MAC Algorithm" has an arrow pointing to a box with two smaller boxes labeled "Message" and "MAC." The Receiver box has a diagram starting with a box that has two smaller boxes labeled "Message" and "MAC." The box has an arrow pointing to a box labeled "MAC computed by sender." "MAC computed by sender" has an arrow pointing to an oval labeled "Compare MACs." The first box containing "Message" and "MAC" has a second arrow pointing to an oval labeled "MAC Algorithm" with a key icon and "Shared key." "MAC Algorithm" has an arrow pointing to a box labeled "MAC computed by receiver." "MAC computed by receiver" has an arrow pointing to "Compare MACs."

Animation captions:

1. A message sender computes the message authentication code (MAC) for a message using a symmetric key shared with a message receiver.
2. A message sender appends the MAC to the message.
3. A receiver removes the MAC from the message.
4. A receiver computes the message MAC with the symmetric key shared with the sender.
5. A receiver compares the two MACs. If the MACs are equal, the message was not modified in transit and was sent by the sender.

PARTICIPATION ACTIVITY 3.8.2: Message authentication code (MAC).

1) What is another name for a MAC?

Check Show answer

2) What type of key is used to compute a MAC?

Check Show answer

3) What does a MAC require other than a symmetric key?

Check Show answer

PARTICIPATION ACTIVITY 3.8.3: Message authentication code (MAC).

1) A MAC is a cryptographic _____ used for verifying data integrity and authenticity.
   - ○ key
   - ○ primitive

2) A message sender appends the MAC to a message _____ sending the message to a receiver.
   - ○ after
   - ○ before

3) A message receiver computes _____ MAC(s).
   - ○ one
   - ○ two

Security services

A MAC provides two security services:

- Data integrity
  If the MAC of a received message is equal to the MAC computed by a message receiver, a message receiver is assured that the message was not modified in transit.
- Data origin authentication
  A valid MAC can only be created by the symmetric key shared between a message sender and a message receiver. Since only a message sender has the shared symmetric key to compute a MAC, a message receiver is assured that a message originated from the sender.

A MAC does not provide the non-repudiation security service. Since a message sender and a message receiver share the same symmetric key, a message sender can deny sending a message. A message sender can claim a message receiver forged a valid MAC by using the shared symmetric key.

PARTICIPATION ACTIVITY 3.8.4: Message authentication code (MAC).

1) A MAC provides data integrity and what other security service?

Check Show answer

2) If a MAC sent with a message is equal to the MAC computed by a receiver, which property of the message is maintained in transit?

Check Show answer

3) Which security service is not provided by a MAC because a message sender and receiver share the same symmetric key?

Check Show answer

Hashed message authentication code (HMAC)

A ***hashed message authentication code (HMAC)***, or ***hash-based authentication code***, is a MAC that uses a symmetric key together with a cryptographic hash function. A message sender uses a cryptographic hash function to compute a message digest. A message sender encrypts a message digest with a symmetric key shared with a message receiver.

An HMAC is denoted by the cryptographic hash function an HMAC uses to compute a message digest. Ex: HMAC-MD5 is an HMAC that uses a message digest computed by the MD5 cryptographic hash function. The HMAC size is equal to the size of the underlying cryptographic hash function. Ex: HMAC-SHA256 generates a 256-bit HMAC for a message.

PARTICIPATION ACTIVITY 3.8.5: Hashed message authentication code (HMAC).

How to use this tool

SHA512 256 bits HMAC

	A cryptographic primitive used for verifying data integrity and authenticity.
	The cryptographic hash function used by HMAC-SHA512.
	The size of an HMAC-SHA256.

Reset

PARTICIPATION ACTIVITY 3.8.6: Hashed message authentication code (HMAC).

1) Which one of the following HMACs uses the Message Digest 5 cryptographic hash function?
   - HMAC-SHA2
   - HMAC-MD4
   - HMAC-MD5

2) What are the required components of an HMAC algorithm?
   - A public key and a cryptographic hash function.
   - A symmetric key and a cryptographic hash function.
   - A public key and a private key.

3) Why should a message sender and a message receiver use the same cryptographic hash function to compute an HMAC?
   - Different cryptographic hash functions require different symmetric keys.
   - Different cryptographic hash functions output different digests for the same message.
   - A message receiver cannot decrypt an HMAC if a message

- sender uses a different cryptographic hash function to compute an HMAC.

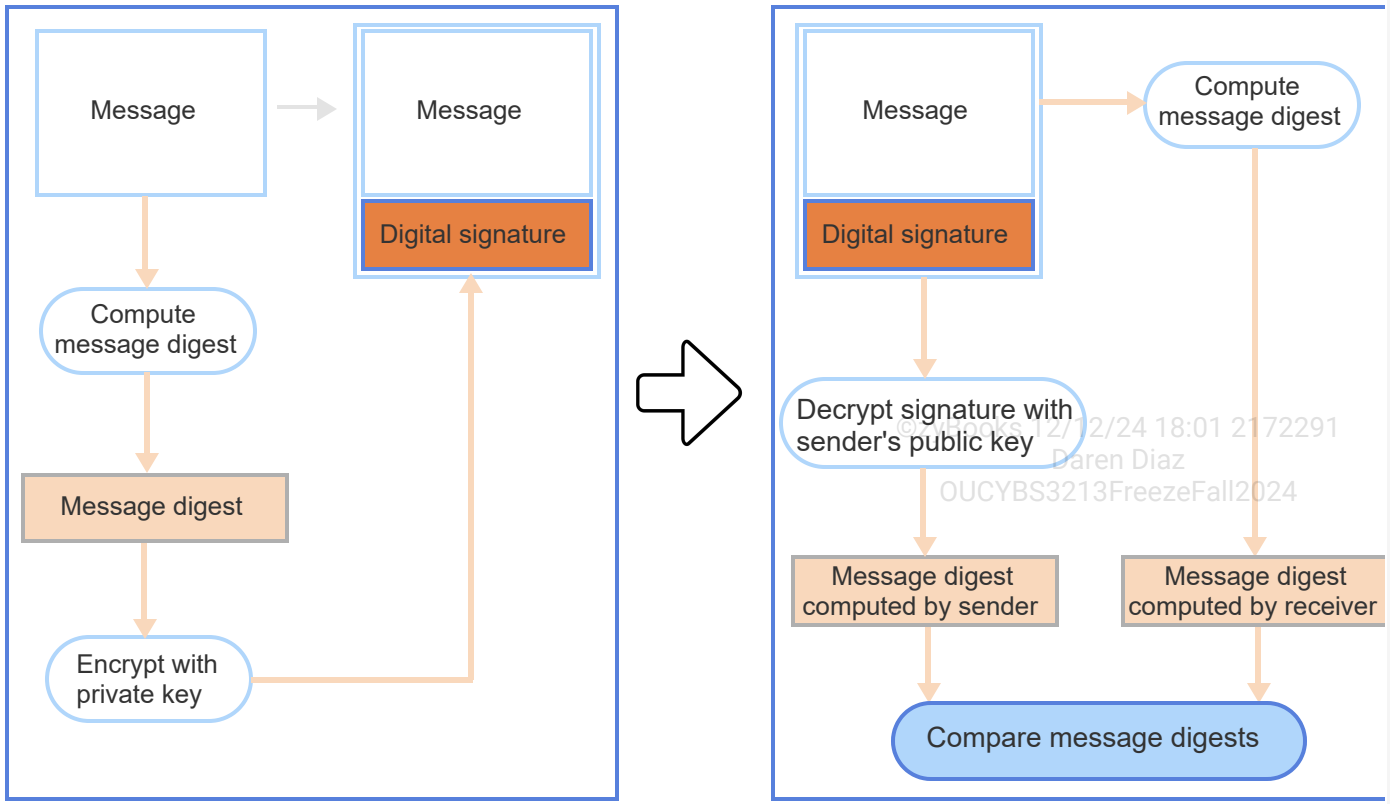
3.9 Digital signatures

Digital signatures

A ***digital signature*** is a cryptographic primitive that uses a cryptographic hash function and public key cryptography to verify the authenticity and integrity of a digital message. A message sender computes the message digest with a cryptographic hash function and applies a signing algorithm which uses the sender's private key to create a digital signature for the message. The digital signature is sent to the message receiver along with the message.

The message receiver applies a signature verification algorithm which uses the sender's public key to verify that the message was signed by the sender and not modified in transit.

PARTICIPATION ACTIVITY 3.9.1: Digital signature using the RSA algorithm.



Sender Receiver

Animation content:

Static figure: The process for sending and receiving a message with a digital signature.

Animation captions:

1. Sender computes the message digest for the message.
2. Sender encrypts the message digest with private key and appends the encrypted message digest to the message. The encrypted message digest is the sender's digital signature.
3. Receiver decrypts sender's digital signature with sender's public key to retrieve message digest.
4. Receiver computes the message digest of the received message.
5. Receiver compares the two message digests. If the message digests are equal, the message was not altered in transit and was sent by the sender.

PARTICIPATION ACTIVITY 3.9.2: Digital signatures.

How to use this tool

Sender's public key **Sender's private key** **Digital signature**

	A cryptographic scheme which uses hash functions and public key cryptography to verify message authenticity and integrity.
	The key used by the message receiver to verify the authenticity and integrity of a message
	The key used by the message sender to sign a message.

Reset

PARTICIPATION ACTIVITY | 3.9.3: Digital Signatures.

1) John received a digitally signed message from Jane, but Jane claims she never sent the message. How does the digitally signed message guarantee that the message was sent by Jane?
   - Jane used John's public key to encrypt the message digest.
   - The message digest John received with the message did not match the message digest John computed.
   - John verified the digital signature with Jane's public key.

2) In a digital signature that uses the RSA algorithm the message sender encrypts the message digest with the sender's public key.
   - True
   - False

3) The message sender and receiver can use different cryptographic hash functions to compute the message digest.
   - True
   - False

Digital signature algorithms

A digital signature requires the use of public-key cryptography. A digital signature can be based on any of the three public-key algorithm families; integer factorization, discrete logarithms, and elliptic curves. Three commonly used digital signature algorithms exist:

- The ***RSA Digital Signature Algorithm*** uses the RSA public-key cryptography. The security of the RSA algorithm is based on the difficulty of factoring large prime numbers.
- The ***Digital Signature Algorithm (DSA)*** uses a variant of the ElGamal public-key cryptography. The security of the ElGamal algorithm is based on the difficulty of solving discrete logarithm problem.

- The ***Elliptic Curve Digital Signature Algorithm (ECDSA)*** uses the elliptic curve public-key cryptography. The security of the elliptic curve algorithm is based on the difficulty of locating related points on an elliptic curve

A digital signature is identified by combining the name of a digital signature algorithm and the underlying cryptographic hash function that the digital signature scheme uses to compute a message hash. Ex: ECDSA-SHA512 is an Elliptic Curve Digital Signature Algorithm (ECDSA) that uses the Secure Hash Algorithm 512 (SHA512) hash function to compute a message hash.

PARTICIPATION ACTIVITY 3.9.4: Digital signature algorithms.

How to use this tool

Digital Signature Algorithm (DSA) **Elliptic Curve Digital Signature Algorithm (ECDSA)**

RSA Digital Signature Algorithm

	Is based on the ElGamal public-key cryptography.
	Is based on the RSA public-key cryptography.
	Is based on the elliptic curve public-key cryptography.

Reset

PARTICIPATION ACTIVITY 3.9.5: Digital signature standard.

1) Which one of the following digital signature schemes uses the Secure Hash Algorithm 2 to compute a message hash?

- RSA-SHA1
- ECDSA-SHA2
- RSA-MD5

2) Which one of the following digital signature schemes uses the Digital Signature Algorithm?

- RSA-SHA1
- DSA-MD5
- ECDSA-SHA1

3) Which one of the following digital signature schemes uses the Elliptic Curve Digital Signature Algorithm and Secure Hash Algorithm 2 cryptographic?

- DSA-MD5
- RSA-SHA1
- ECDSA-SHA2

Security services

A digital signature provides three security services:

- Data integrity
  If the message digest of a sent message matches the message digest computed by the message receiver, the message receiver is assured that the message was not modified in transit.
- Data origin authentication
  A valid digital signature can only be created by a message sender's private key. Since only a message sender has the private key which signed the message, the receiver is assured that the message originated from the sender.
- Non-repudiation
  A message sender cannot deny, or repudiate, that the message was signed. Only the message sender has the private key which signed the message.

A digital signature does not provide the data confidentiality security service. A message sender only encrypts the message digest. A message is not protected from an eavesdropper.

PARTICIPATION ACTIVITY 3.9.6: Security services.

How to use this tool

Data integrity | Non-repudiation | Origin authentication

	A message digest of a sent message matches the message digest computed by the message receiver.
	A message sender cannot deny signing a message.
	A message originated from the sender.

Reset

PARTICIPATION ACTIVITY 3.9.7: Security services.

1) How is the receiver of a signed message assured that the message was not modified in transit?

- Since the message is signed, the receiver is assured that the message was not modified in transit.
- By comparing the message hash with the message hash signed by the message sender.
- By comparing the received message with the sent message.

2) Why can't a message sender deny that the message was signed?

- Only the message sender has the public key which signed the message.
- Only the message sender has the private key which signed the message.

- Another entity signed the message, not the message sender.

3) Why is the receiver of a signed message not assured that the message remained confidential in transit?

- A signed message was not signed by the sender.
- A signed message does not provide data confidentiality.
- A signed message was not modified in transit.

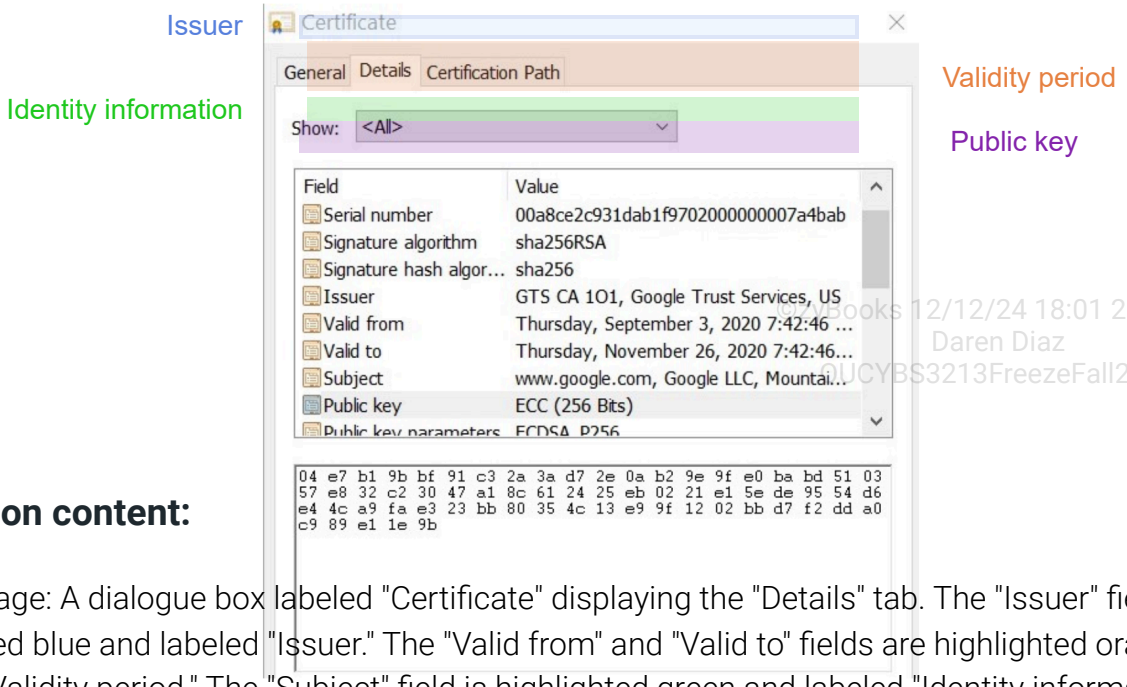
3.10 Digital certificates

Digital certificates

A ***digital certificate***, or ***public key certificate***, is an electronic document which proves the ownership of a public key. A digital certificate contains the certificate owner's public key and identity information, such as name, address, and organization. A digital certificate is only valid for a specific period, known as a certificate's validity period. A certificate is digitally signed by a certificate issuer, known as a ***Certificate Authority (CA)***, that has verified a certificate's content.

Ex: A digital certificate issued to Google includes the name of the certificate authority that issued the certificate (Issuer), the certificate's validity period (the dates the certificate is valid from and valid to), identity information on Google (Subject), and Google's public key.

PARTICIPATION ACTIVITY 3.10.1: Parts of a digital certificate.



Animation content:

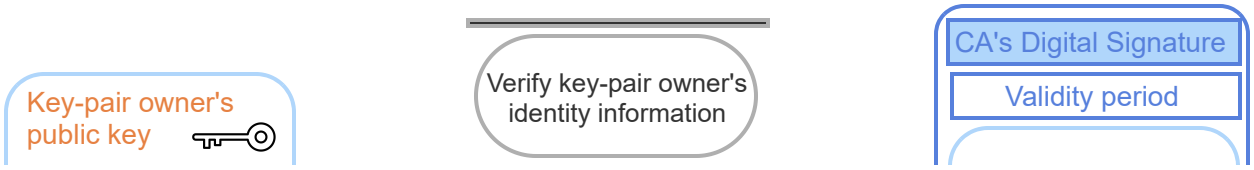
Static image: A dialogue box labeled "Certificate" displaying the "Details" tab. The "Issuer" field is highlighted blue and labeled "Issuer." The "Valid from" and "Valid to" fields are highlighted orange and labeled "Validity period." The "Subject" field is highlighted green and labeled "Identity information." The "Public key" field is highlighted purple and labeled "Public key."

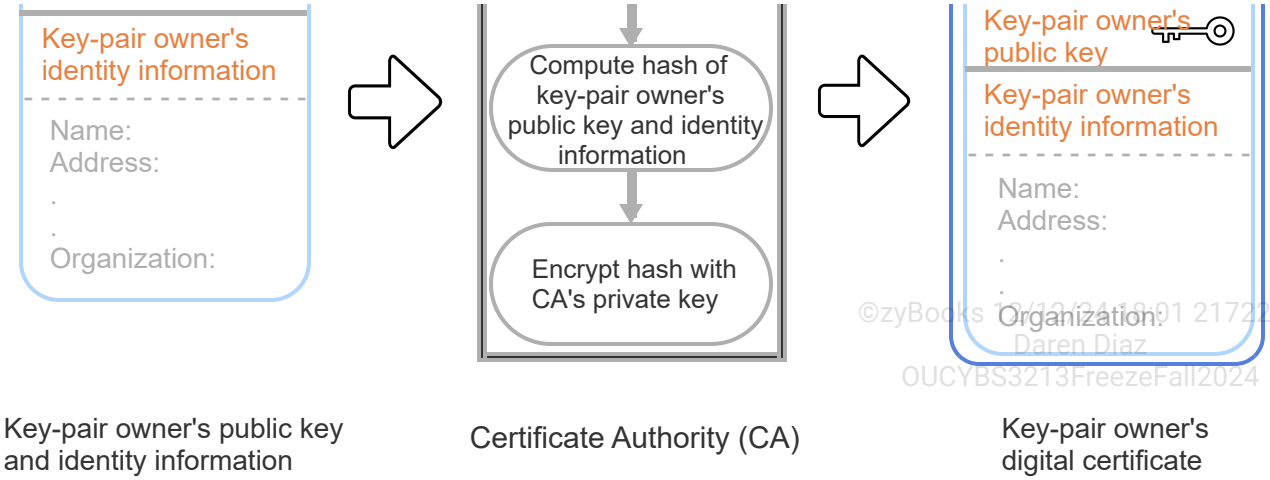
Animation captions:

1. The issuer is the name of the certificate authority that issued the certificate.
2. The validity period is the date the certificate is valid from (September 3, 2020) to the date the certificate is valid to (November 26, 2020).
3. The identity information is also known as the subject.
4. A digital certificate also contains information about the public key.

Only the certificate's owner has the corresponding private key to the public key in the certificate. The public key in a certificate is used by a certificate user to securely communicate with the certificate owner, and to validate documents digitally signed by the certificate owner's private key.

PARTICIPATION ACTIVITY 3.10.2: Digital certificates.





Animation content:

Static image: Three boxes. An arrow points from the first box to the second box. An arrow points from the second box to the third box. The first box is labeled "Key-pair wonder's public key and identity information" and is split into three sections. The top section is labeled "Key-pair owner's public key" and has a key icon. The middle section is labeled "Key-pair wonder's identity information." The third section has fields for "Name," "Address," and "Organization." The second box is labeled "Certificate Authority (CA)" and contains a three-step process. The first step is "Verify key-pair owner's identity information," the second step is "Compute hash of key-pair owner's public key and identity information," and the third step is "Encrypt hash with CA's private key." The third box is labeled "Key-pair owner's digital certificate" and has three parts. The first part is labeled "CA's Digital Signature." The second part is labeled "Validity period." The third part is a copy of the Key-pair owner's public key and identity information found in the first box.

Animation captions:

1. A key-pair owner's public key and identity information is sent to a Certificate Authority (CA).
2. A Certificate Authority (CA) verifies key-pair owner's identify information.
3. A Certificate Authority computes the hash of key-pair owner's public key and identity information.
4. A Certificate Authority digitally signs the key-pair owner's certificate by encrypting the hash of key-pair owner's public key and identity information with the Certificate Authority's private key.
5. A Certificate Authority issues a key-pair owner's digital certificate with a validity period.

PARTICIPATION ACTIVITY 3.10.3: Digital certificates.

1) How can a certificate user be assured that the public key in a digital certificate belongs to the certificate owner?

- The certificate's information on the certificate owner's identity is correct.
- The certificate has been signed by the Certificate Authority.

2) A Certificate Authority verifies the owner's identity information after issuing the digital certificate.

- No
- Yes

3) How does the Certificate Authority digitally sign a public key certificate?

- By encrypting the certificate owner's public key with the Certificate Authority's private key.
- By encrypting the certificate owner's identity information and public key with the Certificate Authority's private key.

4) A digital certificate contains the certificate owner's private key.

- False
- True

PARTICIPATION ACTIVITY 3.10.4: Digital certificate.

How to use this tool

Public key **Certificate Authority** **Public key certificate**

	Another name for a digital certificate.

	The key included in a digital certificate.
	Verifies the identity information of a certificate owner.

Reset

Digital certificate types

A digital certificate is used to securely communicate with a certificate owner, and to validate documents digitally signed by a certificate owner's private key.

A ***web server certificate***, also known as a ***domain certificate*** or ***TLS certificate***, is a certificate that is used by a web server and a web client to establish a secure connection over a network. Four types of web servers exist:

- A ***domain validation (DV) certificate*** only verifies the identity of a domain's owner.
- A ***domain extended validation (EV) certificate*** verifies the identity of a domain's owner, the domain owner's exclusive control over the domain, and the domain owner's legal and physical existence.
- A ***wildcard*** certificate validates a domain and all the domain's subdomains.
- A ***Subject Alternative Name (SAN) certificate*** is used by multiple domains owned by the same domain owner. A SAN certificate is also known as a ***Unified Communication Certificate (UCC)***.

PARTICIPATION ACTIVITY 3.10.5: Web server certificates.

1) A domain _____ certificate only verifies the identity of a domain's owner.
   - ○ extended validation (EV)
   - ○ validation certificate (DV)

2) A _____ certificate is used by multiple domains owned by the same domain owner.
   - ○ wildcard
   - ○ Unified Communication Certificate (UCC)

Several other types of digital certificates exist, including root, code signing, email, and machine certificates.

- A ***root certificate*** is created and self-signed by a Certificate Authority (CA). A ***self-signed*** certificate is a certificate in which the certificate issuer and the certificate subject are the same. A root certificate does not depend on a higher-level authority for authentication. A CA uses a root certificate to sign other certificates.
- A ***code signing certificate*** is used by a software developer, or a software publisher, to digitally sign software programs. An installation program, or an Operating System (OS), uses a code signing certificate to ensure the integrity of a program before installing the program.
- An ***email certificate*** is used by an email user to digitally sign emails. An email sender encrypts an email with the sender's private key. The email recipient decrypts the email with the public key obtained from the sender's email certificate.
- A ***machine certificate***, or ***computer certificate***, is issued to a hardware device such as a computer, a router, or a printer. A machine certificate is used to authenticate a device on a network.

PARTICIPATION ACTIVITY 3.10.6: Digital certificate types.

1) A _____ certificate is used by a software publisher.
   - root
   - code signing

2) A _____ certificate is used for authenticating network devices.
   - machine certificate
   - wildcard

3) A _____ certificate is used for signing emails.
   - code signing
   - email

PARTICIPATION ACTIVITY 3.10.7: Digital certificate formats.

How to use this tool

Root certificate **TLS certificate** **Email certificate**

	A certificate used by an email user to digitally sign an email.
	A certificate used by a certificate authority (CA) to sign other certificates.
	A certificate used by a web server and a web client to establish a secure connection over a network.

Reset

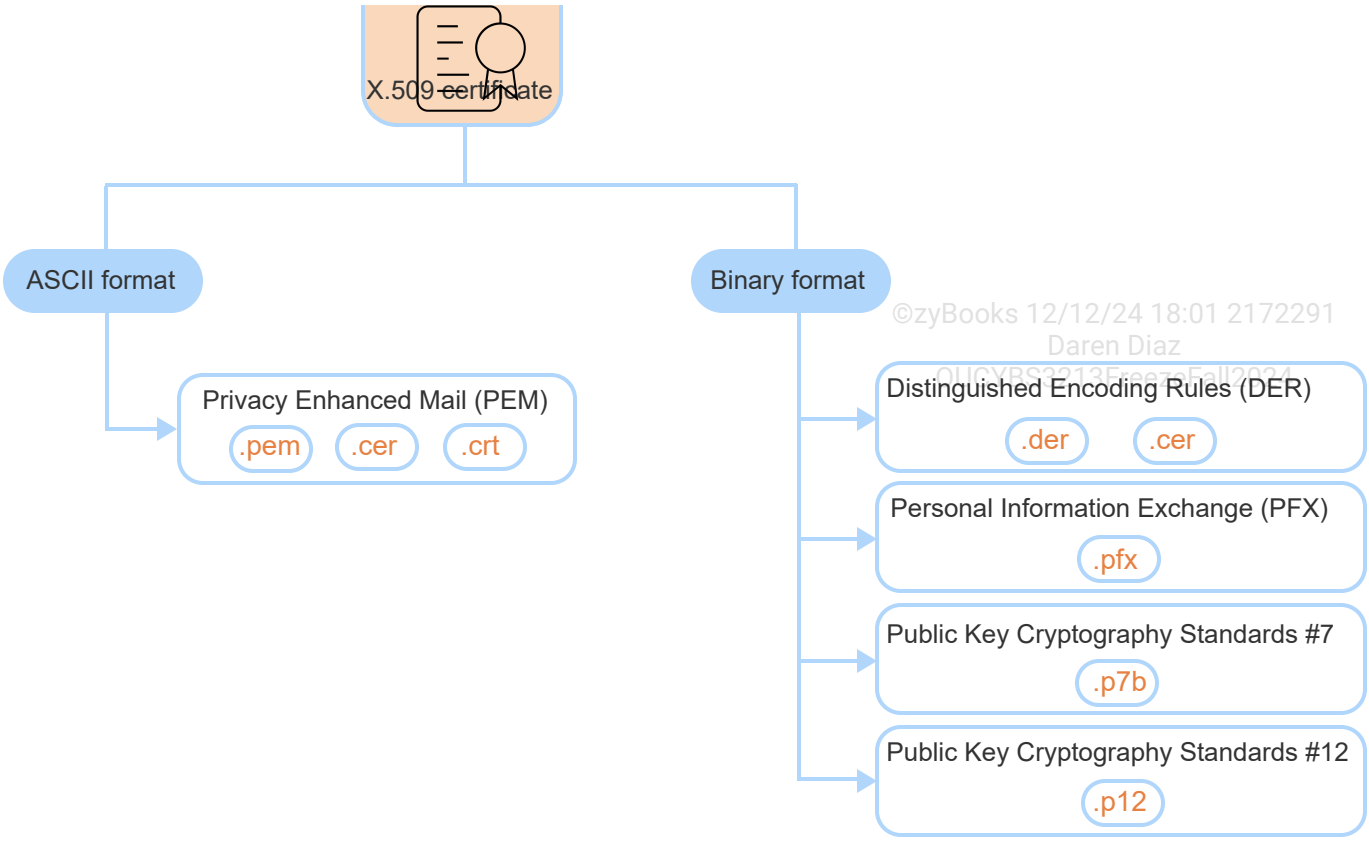
Digital certificate formats

The ***X.509*** standard defines the format of a public key certificate. An X.509 certificate consists of key-value pairs. A key represents a field name and a value represents a number, string, or list.

An X.509 certificate is saved in different formats:

- The ***Privacy Enhanced Mail (PEM)*** ASCII format. The PEM defines methods for encoding binary data using ***Base64***. The Base64 is a binary-to-text encoding scheme that represents binary data in an ASCII string format.The filename extension of a PEM file is '.pem', '.crt', or '.cer'.
- The ***Distinguished Encoding Rules (DER)*** binary format. The DER is a subset of the ***Abstract Syntax Notation One (ASN.1)*** which is a platform-independent encoding format. The DER encoding ensures that the contents of a certificate can only be encoded in one way. The filename extension of a DER file is '.der' or '.cer'.
- The ***Personal Information Exchange (PFX)*** binary format. A PFX file is a password protected archive file format that contains the certificate and the corresponding private key. The PFX format is used by a server to import a certificate and private key from a single file. The filename extension of a PFX file is '.pfx'.
- The ***Public Key Cryptography Standards #7*** (***PKCS #7***) and ***Public Key Cryptography Standards #12*** (***PKCS #12***) binary format. The PKCS #7 and PKCS #12 format defines a standard syntax for storing encrypted and signed data. A PKCS #7 or PKCS #12 is stored in DER binary or PEM ASCII formats. The filename extension of a PKCS #7 file is '.p7b', and the filename extension of PKCS #12 file is '.p12'.

PARTICIPATION ACTIVITY 3.10.8: Digital certificate formats.



Animation content:

Static image: All possible X.509 certificate formats.

Animation captions:

1. A X.509 certificate is saved in binary or ASCII format.
2. The Privacy Enhanced Mail (PEM) is an ASCII file and is saved with filename extensions of ".pem", ".crt", or ".cer".
3. The Distinguished Encoding Rules (DER) is a binary file and is saved with filename extensions of ".der" or ".cer".
4. The Personal Information Exchange (PFX) is a binary file and is saved with filename extensions of ".pfx".
5. The Public Key Cryptography Standards #7 and Public Key Cryptography Standards #12 are binary files and are saved with filename extensions of ".p7b" or ".p12".

PARTICIPATION ACTIVITY 3.10.9: Digital certificate format.

1) A Privacy Enhanced Mail (PEM) is a binary file.

- False
- True

2) A Personal Information Exchange (PFX) file contains a private key.

- False
- True

3) A PKCS #7 or PKCS #12 is stored in DER binary or PEM ASCII formats.

- False
- True

PARTICIPATION ACTIVITY 3.10.10: Digital certificate formats.

How to use this tool

Base64 **Personal Information Exchange (PFX)**

Public Key Cryptography Standards (PKCS)

	A binary-to-text encoding scheme.
	A password protected archive file format that contains the certificate and the corresponding private key.
	Define a standard syntax for storing encrypted and signed data.

Reset

Key escrow

Key escrow *refers to the secure storage and management of cryptographic keys by a trusted third party, known as an* ***escrow agent****. Key escrow ensures encrypted data remains accessible under specific circumstances, such as legal obligations, instances*

of key compromise, or secure key recovery. Key escrow is vital in environments where uninterrupted data availability enables lawful access and prevents data loss due to misplaced or unavailable keys.

CHALLENGE ACTIVITY 3.10.1: Digital certificates.

581480.4344582.qx3zqy7

Start

Select the term that completes each statement.

The corresponding private key to a digital certificate's public key is in the possession of the _____.

Pick

The _____ uses a certificate's public key to securely communicate with a certificate owner.

Pick

A digital certificate user may use the certificate to securely communicate with the _____.

Pick

1 2 3

Check Next

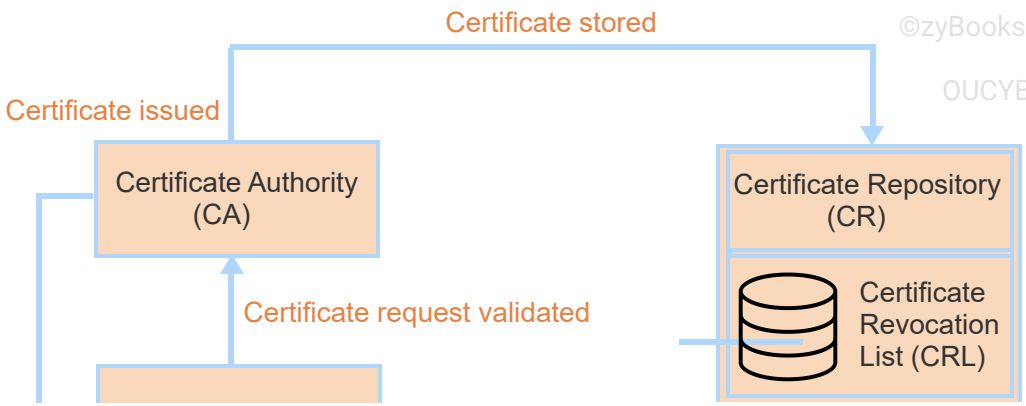
3.11 Public Key Infrastructure (PKI)

Public Key Infrastructure (PKI)

A ***Public Key Infrastructure (PKI)*** is a framework for managing digital certificates and public keys. A PKI enables users of an insecure network, such as the Internet, to securely exchange data by using public and private cryptographic keys. A PKI includes the hardware, software, people, policies, and procedures that enable the creation, renewal, revocation, and distribution of digital certificates. A PKI's components are Certificate Authority (CA), Registration Authority (RA), Certificate Repository (CR), Certificate Policy (CP), and Certificate Practice Statement (CPS).

- A ***Certificate Authority (CA)*** issues, renews, revokes, and distributes digital certificates. A CA is a third-party trusted by both the certificate's owner, the subject, and the certificate user, the relying party. The relying party relies upon the accuracy of the binding between the certificate's public key and the certificate owner's identity.
- A ***Registration Authority (RA)*** verifies the identity of a digital certificate applicant. A certificate applicant sends a ***Certificate Signing Request (CSR)*** to an RA. A ***CSR*** contains the applicant's public key and includes the applicant's name, organization, department, physical and email addresses. An RA validates the applicant's identity and approves or rejects the applicant's CSR. If the RA approves the CSR, the RA forwards the CSR to the CA for the issuing of the applicant's digital certificate.
- A ***Certificate Repository (CR)***, or ***central directory***, stores the digital certificates issued by a CA. A CR also contains the ***Certificate Revocation List (CRL)***. A ***CRL*** is a list of certificates a CA has revoked prior to the expiration of certificates' validity periods.
- A ***Certificate Policy (CP)*** defines the structure of a PKI, describes a PKI's entities and roles, and specifies a PKI's procedures and operational requirements.
- A ***Certificate Practice Statement (CPS)*** describes how a CA issues, renews, revokes, and distributes certificates. A CPS helps a certificate user to decide whether or not to trust a PKI's certificates.

PARTICIPATION ACTIVITY 3.11.1: Public Key Infrastructure (PKI).



Animation content:

Static image: A box labeled "Certificate applicant (Subject)" with an arrow labeled "Certificate Signing Request (CSR)" pointing to a box labeled "Registration Authority (RA)." The Registration Authority box has an arrow labeled "Certificate request validated" pointing to a box labeled "Certificate Authority (CA)." The Certificate Authority box has an arrow labeled "Certificate issued" pointing back to the Certificate applicant box. The Certificate Authority box has a second arrow labeled "Certificate stored" pointing to a box with two sections: "Certificate Repository" and "Certificate Revocation List (CRL)." The Certificate Applicant box has a second arrow labeled "Subject's digital certificate" pointing to a box labeled "Relying party (certificate user)." The text "CA's public key" has an arrow pointing to the Relying party box.

Animation captions:

1. A certificate applicant, or subject, submits a Certificate Signing Request (CSR) to the Registration Authority (RA).
2. The Registration Authority (RA) validates the certificate applicant's identity and informs the Certificate Authority (CA).
3. The Certificate Authority (CA) signs and issues the certificate and sends the certificate to the applicant.
4. The Certificate Authority (CA) stores the certificate in the Certificate Repository (CR). The CR also contains the Certificate Revocation List (CRL).
5. A relying party (certificate user) verifies the subject and the subject's public key ownership by ensuring that the subject's digital certificate has been signed by a Certificate Authority (CA).

PARTICIPATION ACTIVITY | 3.11.2: PKI components.

How to use this tool

CA | CPS | CR | CP | RA

	Issues, revokes, and distributes certificates
	Verifies the identity of a certificate applicant
	Stores the issued certificates
	Defines the procedures and operational requirements of a PKI
	Describes how a CA issues, renews, revokes, and distributes certificates

Reset

PARTICIPATION ACTIVITY 3.11.3: PKI components.

1) Which authority in a PKI is responsible for validating the identity of a digital certificate applicant?

Check Show answer

2) Which authority in a PKI is responsible for issuing a digital certificate?

Check Show answer

3) A CRL contains a list of revoked _____.

Check Show answer

4) Where are issued certificates stored?

Check Show answer

Certificate life cycle

The life cycle of a certificate refers to the stages that a certificate may go through during the certificate's lifetime. The life cycle of a certificate has four stages:

- Issuance: A certificate is issued by a CA after a certificate applicant's identity is validated by an RA. An issued certificate is stored in a CR.
- Revocation: A certificate is revoked by a CA before the end of the certificate's validity period. A revoked certificate is added to the CRL.
- Suspension: A certificate's validity is temporarily suspended by a CA. The validity of a suspended certificate may be re-established, or a suspended certificate may be revoked.
- Expiration: A certificate is expired when the end of the certificate's validity period is reached. A PKI's CP defines the process of applying for a new certificate after a certificate has expired.

PARTICIPATION ACTIVITY 3.11.4: Certificate life cycle.

How to use this tool

Suspension Revocation Issuance Expiration

	A certificate is issued by a CA
	A certificate is revoked by a CA
	A certificate is suspended by a CA
	A certificate's validity period is reached

Reset

PARTICIPATION ACTIVITY 3.11.5: Certificate life cycle.

1) An unexpired certificate's revocation status can be obtained by checking the CA's _____.
   - CP
   - CPS
   - CRL

2) A certificate is stored in a CR when the certificate is _____.
   - revoked
   - issued

3) A certificate owner can revoke the certificate.
   - False
   - True

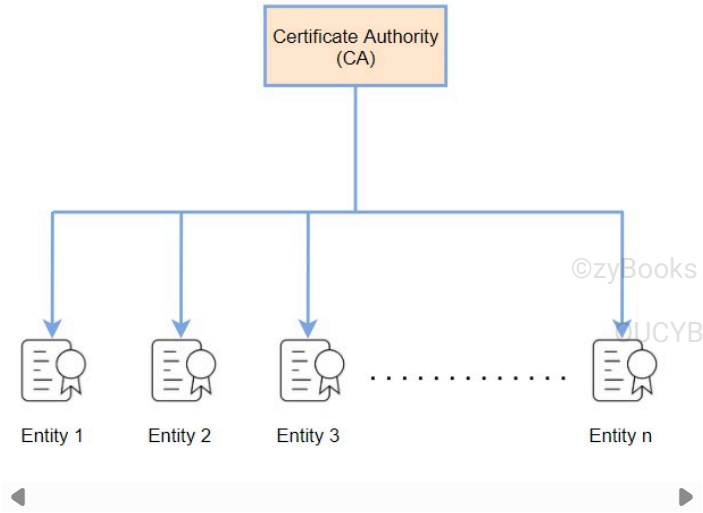
4) A suspended certificate cannot be revoked.
   - False
   - True

PKI trust models

A ***PKI trust model***, or ***PKI architecture***, describes the type of trust relationship which exists between a PKI and PKI's certificate users. A PKI trust model enables a certificate user to determine the legitimacy of a PKI's digital certificates issued to various entities. Three main PKI trust models exist: single CA, hierarchical, and bridge.

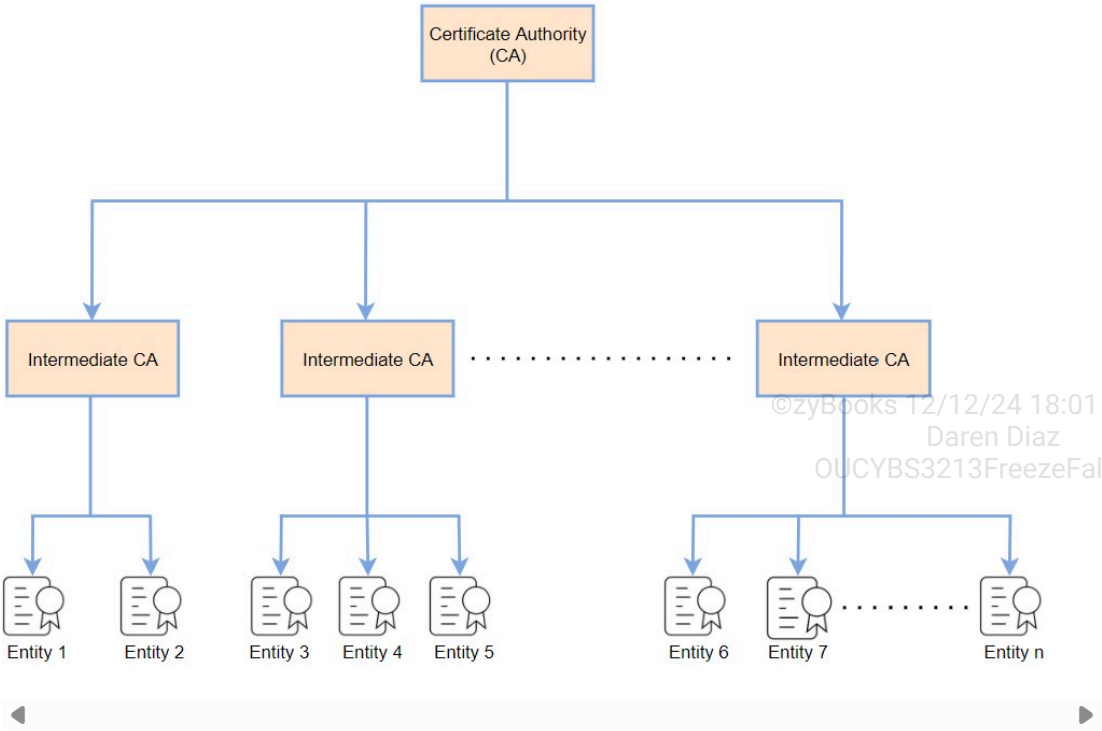
In a ***single CA trust model***, one CA issues all certificates. If the CA's private key is compromised, all PKI's certificates are revoked. A single CA trust model is not scalable to a network with a large number of entities.

Figure 3.11.1: In a single CA PKI trust model, one CA issues certificates to all entities.



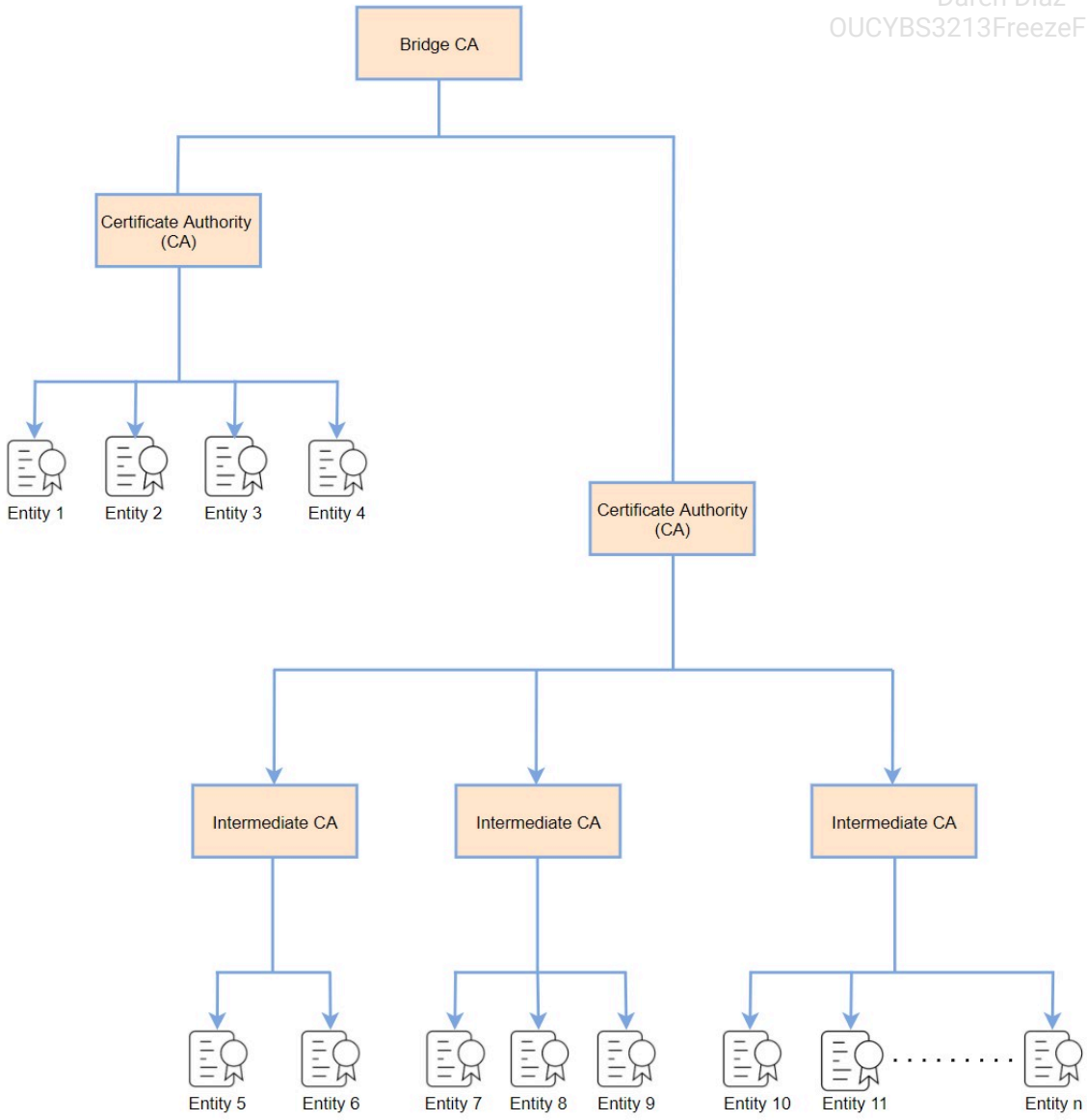
In a ***hierarchical trust model***, a ***root CA*** issues certificates to an ***intermediate CA***, and an intermediate CA issues certificates to entities. The root CA does not issue certificates to entities. A certificate user trusts the certificates issued by an intermediate CA because the intermediate CA is trusted by the root CA. The ***chain of trust*** describes the trust relationship between a certificate user and the intermediate CA which issued the certificate. The root CA is called the ***trust anchor***. The compromise of an intermediate CA's private key only impacts the certificates which the intermediate CA issued. The hierarchical PKI trust model is the most common trust model in use on the Internet.

Figure 3.11.2: In a hierarchical PKI trust model, a root CA issues certificates to intermediate CAs and intermediate CAs issue certificates to entities.



A ***bridge trust model***, or ***BCA***, links PKIs with different trust models. The bridge CA only establishes trust paths between linked PKIs and does not issue certificates to any entities.

Figure 3.11.3: In a bridge PKI trust model, a bridge CA interconnects CAs from different PKIs.



PARTICIPATION ACTIVITY 3.11.6: PKI trust models.

How to use this tool

Bridge trust model | Hierarchical trust model | Single CA trust model

	One CA issues all PKI certificates
	A root CA issues certificates to intermediate CAs
	One CA connects PKIs with different trust models

Reset

PARTICIPATION ACTIVITY 3.11.7: Public Key Infrastructure.

1) What is the most common PKI trust model in use on the Internet?
   - Single CA trust model
   - Hybrid trust model
   - Hierarchical trust model

2) A PKI trust model describes how an entity can obtain a digital certificate.
   - False
   - True

3) A PKI trust model enables a certificate user to determine the legitimacy of a PKI's digital certificates.
   - False
   - True

Key management system

*A **key management system (KMS)** is a software solution designed to manage the entire lifecycle of cryptographic keys, including key generation, distribution, and revocation. A KMS securely manages both public and private keys, automating encryption processes and strengthening security within PKI systems. A KMS enhances security by*

centralizing key management and integrating with trusted platform modules (TPMs) and hardware security modules (HSMs). Ex: In a cloud environment, a KMS could be utilized to centrally manage encryption keys, leveraging integrated TPMs and HSMs for enhanced security in storing and processing sensitive data.

CHALLENGE ACTIVITY 3.11.1: Public Key Infrastructure (PKI).

Start

Select the PKI term associated with each of the following tasks.

Completes a Certificate Signing Request (CSR)

Pick

Verifies the identity information in a Certificate Signing Request (CSR)

Pick

Suspends a digital certificate

Pick

1 2 3

Check Next

3.12 Blockchain

Blockchain

A ***digital ledger*** is an electronic system or database used to record and track transactions. Digital ledgers are utilized in finance, business, and record-keeping applications to monitor and audit transactions. Ex: Banks use digital ledgers to record and manage customer transactions. ***Blockchain*** is a decentralized and distributed digital ledger that records and links transactions across a network of computers. Ex: The Bitcoin cryptocurrency uses blockchain to record and verify transactions.

Blockchain enables the creation and maintenance of open public ledgers. An ***open public ledger*** is a transparent record-keeping system that is accessible to the public, enabling anyone to verify recorded transactions. Ex: The Ethereum blockchain is an open public ledger for smart (self-executing) contracts.

Blockchain can be used to secure and track any type of information. Ex: A system for recording property transactions may use blockchain to securely and transparently process the buying and selling of property, thus reducing the risk of fraud.

Table 3.12.1: Comparison of traditional digital ledger and blockchain.

Aspect	Traditional digital ledger	Blockchain
Architecture	Centralized	Decentralized
Security	Centralized security measures such as data encryption and access control	Use of cryptographic services to enable transparency and tamper-resistant record-keeping
Trust	Relies on a central authority such as a company or government	Relies on cryptographic verification and decentralized consensus across a network of computers
Application	General transaction recording	Cryptocurrencies, smart contracts, decentralized finance (DeFi)

PARTICIPATION ACTIVITY 3.12.1: Blockchain.

1) What is the main purpose of a digital ledger?

- Auditing cryptocurrency holdings

- Recording and tracking transactions
- Creating secure smart contracts

2) What is an advantage of the decentralized nature of blockchain?

- Faster transaction processing
- Limited transparency
- Removal of a single point of failure

3) In what way does blockchain technology potentially reduce the risk of fraud in property transactions?

- By allowing anonymous transactions to protect user privacy
- By utilizing smart contracts that automatically execute under agreed-upon conditions
- By centralizing property records in a single, government-managed database

4) How might blockchain be applied in the education sector?

- By limiting access to academic records
- By providing secure and verifiable academic credentialing
- By creating a centralized grading system

5) How can blockchain technology benefit the supply chain industry?

- By increasing centralization of data
- By limiting traceability of goods
- By improving data security and transparency

6) What is a potential advantage of using blockchain in healthcare?

- Centralized patient data storage
- Limited access to medical records
- Improved data security and integrity

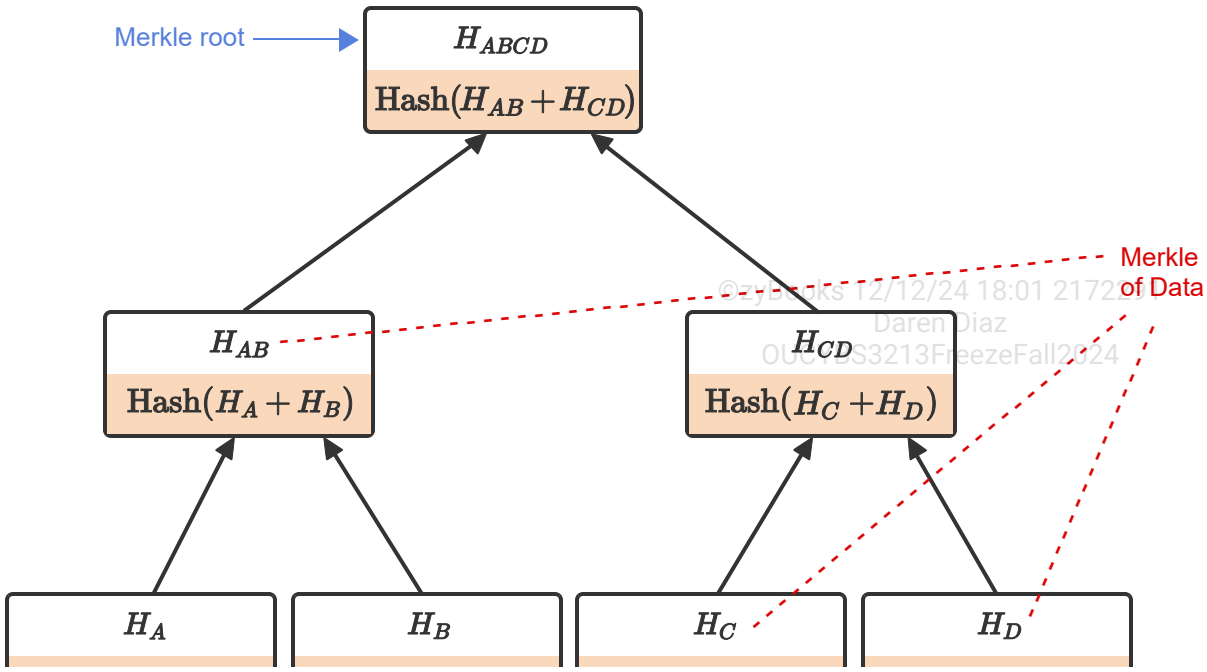
Merkle tree

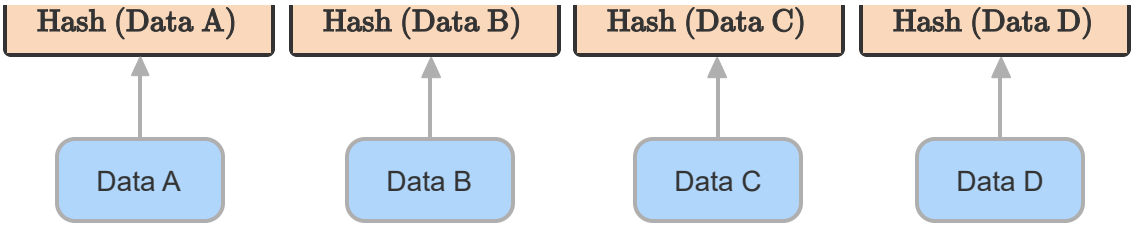
A ***Merkle tree*** is a hierarchical data structure used for verifying the integrity of large data sets. Merkle trees facilitate secure data verification in diverse applications. Ex: Merkle trees are used for transaction verification in blockchain, digital certificate revocation status in public key infrastructure (PKI), and file integrity assurance in distributed file systems.

A Merkle tree is organized in a binary tree format. Data elements are initially stored in the tree's leaves, with each leaf node containing a data element's hash. A Merkle tree is built by repeatedly hashing child node pairs to create parent nodes until reaching a single top node. A ***Merkle root***, or ***root hash***, is the topmost hash in a Merkle tree, uniquely representing the entire dataset's cryptographic hashes.

A ***Merkle path*** is the sequence of nodes connecting a data element to the root. A data element's Merkle path consists of the hashes of sibling nodes along the path from the data element to the root. A Merkle path verifies a data element's integrity and inclusion in the tree without full dataset access.

PARTICIPATION ACTIVITY 3.12.2: A Merkle tree with four data elements.





Animation content:

Static figure: Four data elements, Data A, Data B, Data C, and Data D presented in four boxes at the bottom. Pairs of element boxes are connected to boxes which represent the hash of two data elements. A box on the top is labeled Merkle root. Three dotted lines point to the Merkle path of Data element D.
Step 1: Each data element's hash is computed and stored in the tree's leaves. Ex: H(Data A), H(Data B), H(Data C), and H(Data D) are the hashes of Data A, Data B, Data C, and Data D, respectively.
Step 2: The hashes of child nodes are paired and hashed to create parent nodes. Ex: H(A) and H(B) are paired and and hashed to create node H(AB), and H(C) and H(D) are paired and hashed to create node H(CD)
Step 3: The iteration continues until a single hash, known as the Merkle root, is obtained at the top of the tree. Ex: H(ABCD) is the Merkle root.
Step 4: Ex: The Merkel path of Data D is (HD, HC, HAB), because HD and HC are necessary for computing HCD, and HAB is necessary for computing HABCD (the Merkle root).

Animation captions:

1. Each data element's hash is computed and stored in the tree's leaves. Ex: H_A, H_B, H_C, and H_D are the hashes of Data A, Data B, Data C, and Data D, respectively.
2. The hashes of child nodes are paired and hashed to create parent nodes. Ex: H_A and H_B are paired and hashed to create node H_{AB}, and H_C and H_D are paired and hashed to create node H_{CD}.
3. The iteration continues until a single hash, known as the Merkle root, is obtained at the top of the tree. Ex: H_{ABCD} is the Merkle root.
4. Ex: The Merkel path of Data D is (H_D, H_C, H_{AB}). H_D and H_C are combined to compute the parent hash H_{CD}, followed by combining H_{CD} with H_{AB} to compute H_{ABCD} (the Merkle root).

PARTICIPATION ACTIVITY 3.12.3: Merkle tree.

1) What is the main purpose of a Merkle tree?

- To replace traditional file systems
- To organize data in a linear structure
- To efficiently verify the integrity of large data sets

2) What is the final result of the hashing process in a Merkle tree?

- Root hash
- Leaf node hash
- Non-leaf node hash

3) How does a Merkle path provide cryptographic proof of data inclusion?

- By combining hashes along the tree
- By organizing data in a hierarchical structure
- By connecting a specific data element to the Merkle root

4) How does a Merkle root change if a single data element in a Merkle tree is altered?

- Merkle root remains unchanged
- Merkle root changes, affecting the entire tree's structure
- Merkle root changes, affecting only the altered data element's hash

5) What is a potential limitation of using a Merkle tree in a system with a dynamic dataset?

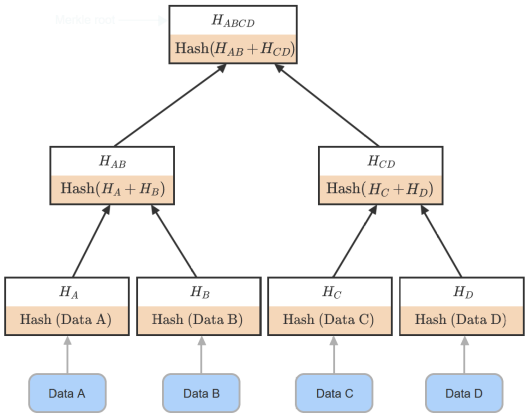
- Inefficient verification process
- Difficulty in maintaining data integrity

- Increased susceptibility to hash collisions

6) How can Merkle trees improve the management and verification of document versions within version control systems?

- By consolidating all versions into a single leaf for simplified access
- By enabling efficient comparison and identification of changes between versions
- By automating real-time updates to all versions, eliminating manual version control

7) What is the Merkle path of Data A?



- (H_A, H_C, H_{AB})
- (H_B, H_C, H_{CD})
- (H_A, H_B, H_{CD})

Blockchain block structure

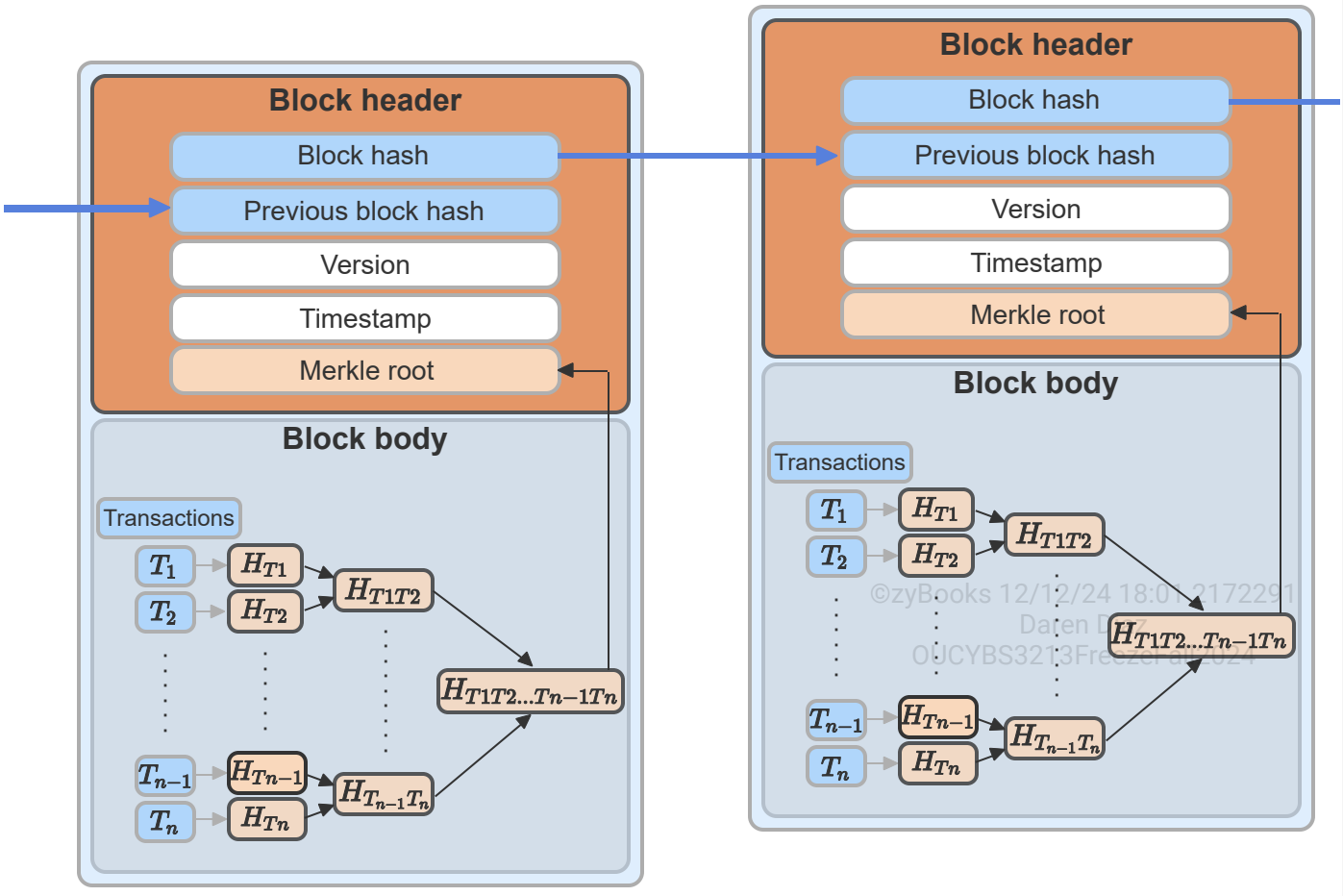
A blockchain functions within a network of computers or nodes, with each node maintaining a copy of the entire ledger. Transaction data, organized using a Merkle tree structure, is grouped into distinct blocks. Each block is composed of two components:

- The ***block header*** is the block component that provides metadata about the block and includes the block hash, previous block hash, blockchain protocol version, block creation timestamp, and Merkle root of the block's transactions.
- The ***block body*** is the block component that contains the content or payload of the block, which depending upon the blockchain's intended purpose, can consist of transactions or application-specific data.

Every new block is cryptographically linked to the preceding block through the previous block hash, forming a chain that is tamper-resistant and difficult to alter retroactively without altering all subsequent blocks. Once a block is added to the chain, the block becomes immutable (unchangeable), ensuring the integrity of the block's transactions.

The Merkle root optimizes data transmission and verification within a blockchain network. Nodes can validate the inclusion of a particular transaction in a block by computing the hashes along the transaction's Merkle path, removing the need to download and verify the entire set of transactions within the block.

PARTICIPATION ACTIVITY 3.12.4: Blockchain blocks.



Animation content:

Static figure: Two blocks with each block composed of two parts. The top part is labeled block header and the bottom part is labeled block body. The block header has five fields labeled block hash, previous block hash, version, timestamp, and Merkle root. The block body contains a list of transactions from T1 to Tn and the corresponding Merkle tree structure.

Step 1: A block is composed of block header and block body. The block header includes metadata about the block and the block body contains transactions.
Step 2: A block's transactions are organized in a Merkle tree structure in the block body and the Merkle root is computed.
Step 3: The Merkle root is stored in the block header, along with the block creation timestamp and blockchain protocol version.
Step 4: The hash of the previous block is added to the block header and the hash of the entire block is computed and stored.
Step 5: Every new block is cryptographically linked to the preceding block through the previous block hash, forming a chain. A second block appears with an arrow from the first block's block hash pointed to the second block's previous block hash.

Animation captions:

1. A block is composed of block header and block body. The block header includes metadata about the block and the block body contains transactions.
2. A block's transactions are organized in a Merkle tree structure in the block body and the Merkle root is computed.
3. The Merkle root is stored in the block header, along with the block creation timestamp and blockchain protocol version.
4. The hash of the previous block is added to the block header and the hash of the entire block is computed and stored.
5. Every new block is cryptographically linked to the preceding block through the previous block hash, forming a chain.

PARTICIPATION ACTIVITY 3.12.5: Blockchain data structure.

1) What components are typically found in a block within a blockchain?

- Data, Timestamp, Previous block hash, Nonce
- Data, Timestamp, Previous block hash

- Data, Timestamp, Previous block hash, Difficulty Level

2) Which term describes the feature of a blockchain that ensures once a block is added, the block cannot be altered or deleted?

- Block validation
- Data encryption
- Immutability

3) What happens once a new block is added to a blockchain?

- The added block remains mutable (changeable) for a limited time
- The added block becomes immutable
- The entire blockchain becomes mutable

4) How does a Merkle tree contribute to the efficiency of data integrity verification in blockchain?

- By verifying the entire dataset
- By using linear data structure
- By limiting hash comparisons during verification

5) What is the purpose of linking blocks together in chronological order to form a chain?

- To provide a visual representation of the blockchain
- To ensure that blocks are randomly connected for added security
- To establish a secure and chronological sequence of transactions

©zyBooks 12/12/24 18:01 2172291
Daren Diaz
OUCYBS3213FreezeFall2024

6) What role do computers in a blockchain network play in maintaining the blockchain?

- Computers only store the most recent blocks in a blockchain
- Each computer maintains a copy of the entire blockchain
- Computers are responsible for altering and updating the blockchain

7) Consider a scenario where a blockchain network chooses not to make blocks immutable. What potential consequences may arise for the reliability of the blockchain?

- Improved trust in the system
- Increased susceptibility to fraud and data manipulation
- Immutability has no impact on the reliability of the blockchain

Blockchain consensus algorithms

Since blockchain is decentralized, network nodes must verify the validity of transactions and reach consensus before a new block is added to the blockchain. A ***consensus algorithm*** enables a network of decentralized nodes to agree on the state of a shared digital ledger. Consensus algorithms are critical for achieving a common and distributed understanding of the blockchain's transaction history. Three common consensus algorithms exist:

- ***Proof of Work (PoW)*** is a blockchain consensus algorithm that requires network participants to perform a computationally intensive task, known as mining. A task typically involves finding a value (nonce) which when combined with a block's data, produces a certain hash. Ex: PoW is used by Bitcoin.
- ***Proof of Authority (PoA)*** is a blockchain consensus algorithm where validators are chosen based on the validators' identity and reputation. The authority to create new blocks is only granted to known and trusted participants. Ex: PoA is used by Celo, a blockchain platform for financial services.
- ***Proof of Stake (PoS)*** is a blockchain consensus algorithm that selects a network participant to create a new block based on the amount of assets the participant holds and is willing to stake.

Staking is the act of committing an asset as collateral to participate in the consensus process. Ex: PoS is used by Ethereum, a blockchain platform for smart contracts.

The choice of a consensus algorithm affects the scalability, security, and decentralization level of a blockchain.

Table 3.12.2: Comparison of blockchain consensus algorithms.

Aspect	PoW	PoA	PoS
Consensus algorithms	Participants solve mathematical problems	Validators chosen based on authority	Validators chosen based on stake
Energy efficiency	High energy consumption	Lower energy consumption compared to PoW	Energy efficient
Security	High security due to computational effort	Moderate security (relies on trusted nodes)	Security through economic incentives
Incentives	Participants motivated by block rewards and transaction fees	Validators motivated by maintaining network authority	Validators motivated by staking rewards

PARTICIPATION ACTIVITY 3.12.6: Blockchain consensus algorithms.

1) What is the purpose of consensus algorithms in blockchain?

   - To centralize control of the blockchain
   - To ensure agreement among nodes on the validity of transactions
   - To encrypt all data on the blockchain for security

2) Which consensus algorithm requires participants to solve complex

mathematical problems to validate transactions before adding blocks to a blockchain?

- Proof of Authority
- Proof of Stake
- Proof of Work

3) How does PoS differ from PoW?

- PoW is more energy-efficient than PoS
- PoS does not involve solving mathematical problems
- PoS and PoW are identical and follow the same principles

4) How does PoA differ from PoW?

- PoA allows for anonymous block validators
- Both PoA and PoW allow any participant to validate transactions without any prerequisites
- PoA involves a predefined set of validators, while PoW requires solving mathematical problems

5) How does PoW contribute to the security of a blockchain network?

- By limiting the number of participants
- By centralizing control
- By requiring computational work for consensus

6) What is the main factor that determines a participant's chances of being chosen to validate transactions in PoS?

- Geographic location and willingness to move close to a network node

- Computational power and willingness to solve complex mathematical problems
- Asset ownership and willingness to stake

Bitcoin's environmental impact

Bitcoin's use of the PoW consensus algorithm requires participants to solve complex mathematical problems requiring significant amounts of computational power and electricity. The energy consumption associated with Bitcoin blockchain has become substantial, leading to harmful consequences for the environment. Estimates have put Bitcoin network's total electricity consumption higher than that of entire countries, including Australia, Greece, and Switzerland. The environmental impact of Bitcoin highlights the need for using more efficient and sustainable blockchain consensus algorithms.

3.13 Obfuscation methods

Steganography

Steganography is an obfuscation technique that hides data by embedding data in a host medium without noticeably altering the medium's appearance or functionality. Ex: Embedding a text message in a PNG image without affecting the image's visible quality or ability to be rendered.

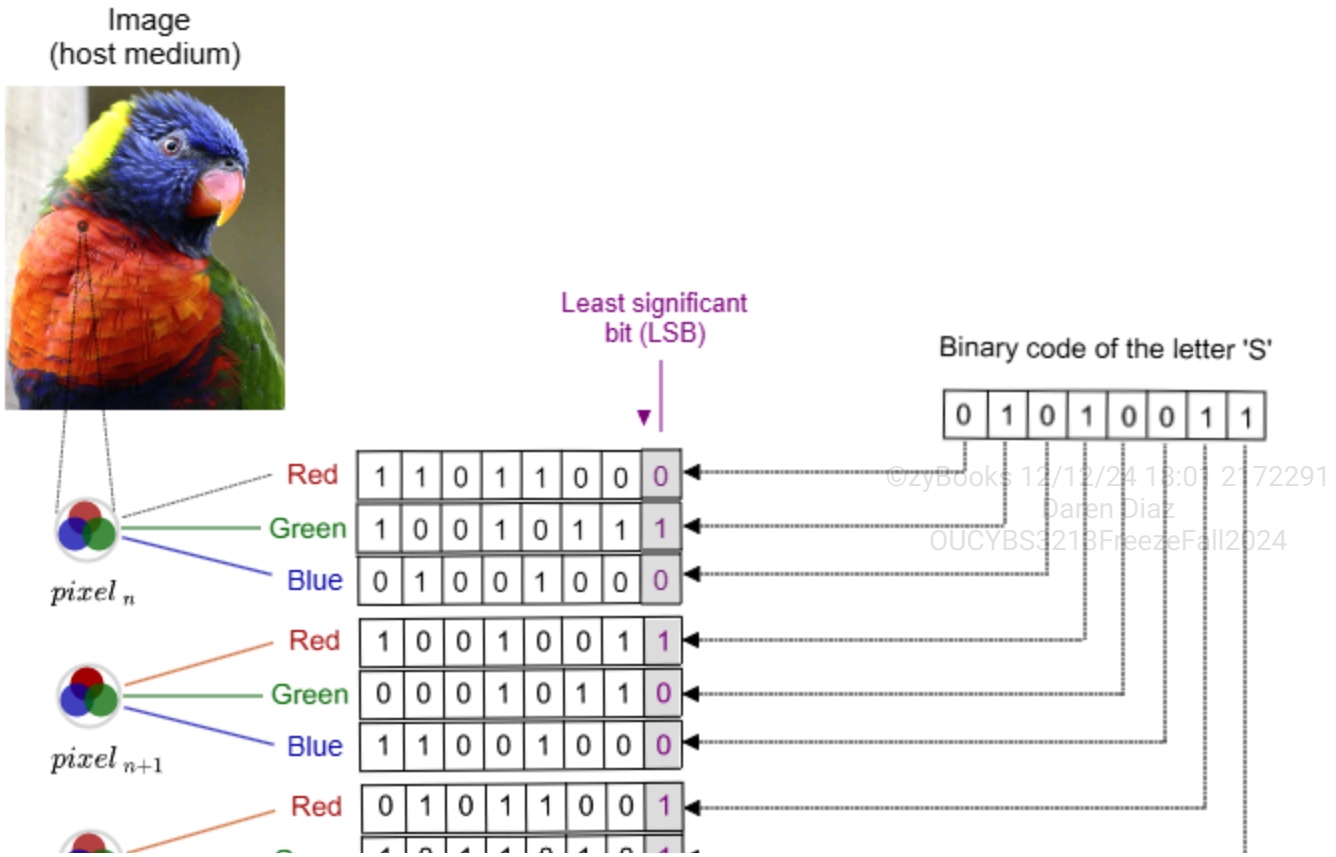
Steganography provides a covert method for protecting and transmitting sensitive data across various applications. Ex: Steganography is used to protect intellectual property by concealing copyright information within digital content, secure confidential data by embedding data within unsuspecting images, and enable secret communication by hiding data in network packets.

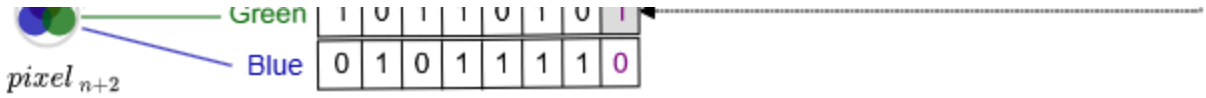
Steganographic techniques leverage the unique properties of different media types to hide data while preserving the media's perceptual integrity. Ex: Text steganography hides data within text using invisible characters such as tabs and spaces, without altering the text's readability, structure, formatting, and semantics.

Table 3.13.1: Steganography methods by media type.

Media type	Description	Example
Image	Alters pixel values to hide data	Secret messages embedded in PNG or JPEG files
Video	Embeds data within video frames or audio tracks	Data integrated into video clips
Audio	Manipulates sound elements to conceal data	Codes hidden in music or voice recordings
Text	Modifies text formatting to encode data	Hidden messages using spaces or font changes

PARTICIPATION ACTIVITY 3.13.1: Image steganography using the least significant bit (LSB) technique.





Animation content:

Static image: An image of a parrot with three circles colored red, green, and blue pointing to one of the image's pixels. To the right of the pixel, the binary representation of each primary color's density is shown. On the right side of the image the binary representation of the letter 'S'; is shown. Arrows connect the bits of the letter S binary representation to the right-most binary positions of the image pixel's binary representations.
Step 1: A digital image is composed of a grid of picture elements called pixels. Each pixel represents a single point in the image and stores specific color data.
Step 2: A pixel's color is composed of three components: red, green, and blue. Each component is an 8-bit binary value, allowing 256 intensity levels per component, ranging from 0 (0000 0000) to 255 (1111 1111)
Step 3: Since the least significant bit (LSB) is the lowest bit in a color value, changing the LSB results in subtle color changes that are undetectable to the human eye.
Step 4: The LSB image steganography technique replaces the least significant bits of the host's pixel data with bits from the message. Ex: Embedding the binary code of the letter 'S' in the host image's pixel LSBs.

Animation captions:

1. A digital image is composed of a grid of picture elements called pixels. Each pixel represents a single point in the image and stores a specific color value.
2. A pixel's color consists of three components: red, green, and blue. Each component is an 8-bit binary value, allowing 256 intensity levels per component, ranging from 0 (**0000 0000**) to 255 (**1111 1111**).
3. Since the least significant bit (LSB) is the lowest bit in a pixel's color value, changing the LSB results in subtle color changes that are undetectable to the human eye.
4. The LSB image steganographic technique replaces the least significant bits of host image's pixel color values with bits from a message. Ex: Embedding the binary code of the letter 'S' in host image's pixel LSBs.

Image sources:

1: Parrot by Scopio licensed under CC BY-NC-ND 2.0

PARTICIPATION ACTIVITY 3.13.2: Steganography.

1) What is the main purpose of steganography?

- Creating new media
- Hiding the presence of data
- Enhancing the color of digital images

2) How is a host medium manipulated for steganographic purposes?

- The medium's format is changed
- The medium is subtly changed to embed hidden data
- The medium is significantly changed to embed hidden data

3) What characteristic of the host medium is critical for successful steganography?

- Large medium size
- Medium complexity
- Unnoticeable changes

4) What is the maximum size of a message that can be embedded in a 1 MB image file using the LSB steganographic technique?

- 128 KB
- 512 KB
- 2048 KB

5) What is the potential risk of using JPEG images as host media?

- The high quality of JPEG images make modifications detectable
- JPEG compression might corrupt embedded data due to loss of bit information
- Compressed images are typically smaller and less capable of hiding data

©zyBooks 12/12/24 18:01 2172291
Daren Diaz
OUCYBS3213FreezeFall2024

6) How can data be hidden in audio without being detected?

- By increasing audio bitrate
- By making hidden data audible
- By manipulating audio signals

7) Network steganography enables covert communication by modifying network protocol fields. Which of the following TCP header fields can potentially be used to hide data?

TCP header

Source port			Destination port
Sequence number			
Acknowledgment number			
Data offset	Reserved	Flags	Window size
Checksum			Urgent pointer
Options			Padding
Data			

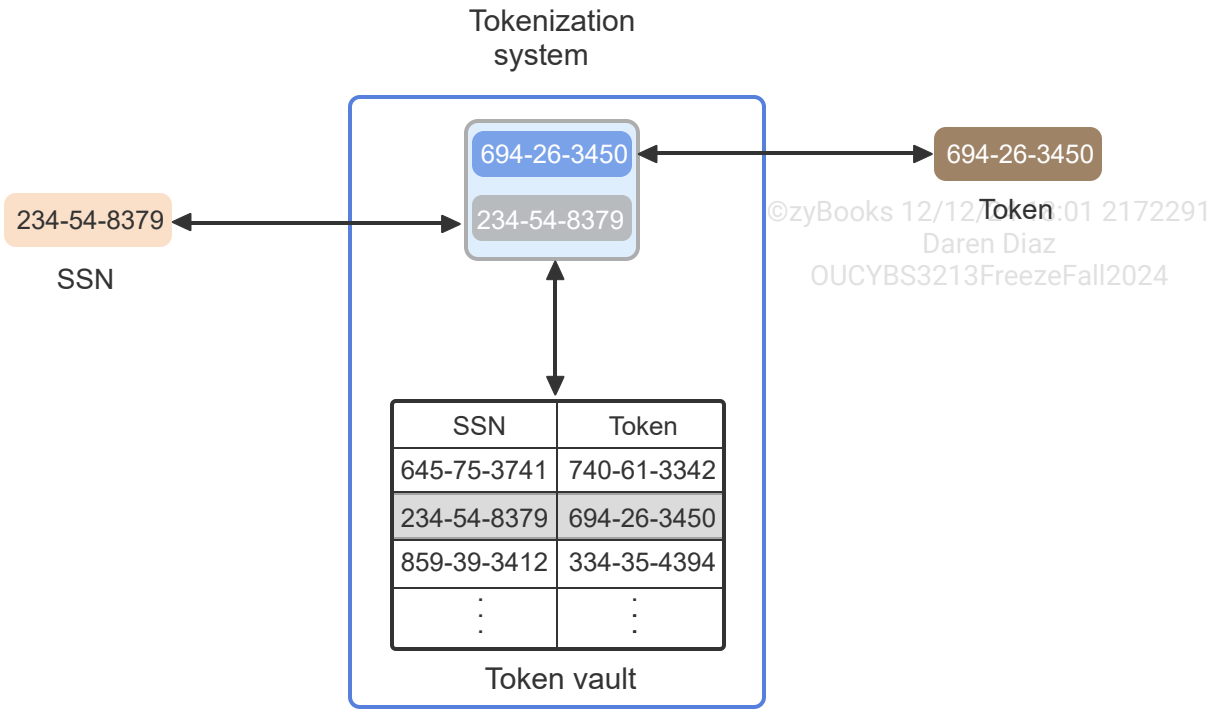
- Source port
- Window size
- Padding

Tokenization

Tokenization is an obfuscation method that replaces a sensitive data element with a non-sensitive data element, known as a ***token***. Ex: A token consisting of a random string of numbers can substitute for a social security number in a database. A ***token vault*** is a secure storage system that maintains the relationship between sensitive data and the corresponding tokens.

A token has no external meaning or exploitable value and serves only as an identifier used for mapping back to the associated sensitive data. Both tokenization and encryption are methods to protect sensitive data. However, unlike encrypted data, tokenized data is not reversible and has no mathematical relationship with the original sensitive data. Ex: Medical records are tokenized to maintain patient confidentiality while allowing data analysis without exposing sensitive information.

PARTICIPATION ACTIVITY 3.13.3: Tokenization.



Animation content:

Static image: A box labeled 'Tokenization system' contains a table labeled 'Token vault' with two columns named 'SSN' and 'Token'. Above the 'Token vault' table and within the 'Tokenization system' box, a blue box contains the text '694-26-3450' and a gray box contains '234-54-8379' underneath the blue box. To the left of the 'Tokenization system' box, the text '234-54-8379' has the label of 'SSN' and to the right of the 'Tokenization system' box, the text '694-26-3450' has the label of 'Token'.
Step 1: In tokenization, a sensitive data element such as a social security number (SSN), is replaced with a token.
Step 2: A token vault stores a mapping between SSNs and tokens.
Step 3: The tokenization system substitutes a token for a social security number, protecting sensitive data and preventing data exposure.

Animation captions:

1. Tokenization replaces a sensitive data element such as a social security number (SSN) with a token.
2. A token vault stores a mapping between SSNs and tokens.
3. The tokenization system substitutes a token for a social security number, protecting sensitive data and preventing data exposure.

PARTICIPATION ACTIVITY 3.13.4: Tokenization.

1) What is the primary purpose of tokenization?
   - To speed up data processing
   - To hide the sensitive nature of data
   - To create a new form of data encryption

2) What is the function of a token vault?
   - To encrypt data using tokens
   - To store physical copies of data
   - To maintain the relationship between sensitive data and tokens

3) Which of the following is a characteristic of a token?
   - A token serves only as an identifier
   - A token can be easily exploited for value
   - A token has significant external meaning

4) How does the absence of a mathematical relationship between a token and the original data affect the security of tokenized data?
   - Enhances security by preventing decryption
   - Increases the data's vulnerability to correlation attacks
   - Makes tokenized data more susceptible to brute force attacks

5) In what way does tokenization impact data usability compared to encryption?

- Tokenization makes data completely unusable for analytical purposes
- Tokenization restricts the use of data to only encrypted transactions
- Tokenization allows for more flexible use of the data in operational systems

3.14 LAB: Asymmetric cryptography (Walkthrough)

IT-Labs are not printable at this time.

3.15 LAB: Public Key Infrastructure (PKI) (Walkthrough)

IT-Labs are not printable at this time.

3.16 LAB: Securing email communications (Scenario)

IT-Labs are not printable at this time.