Cryptography

Dr. Risala Tasin Khan

Professor

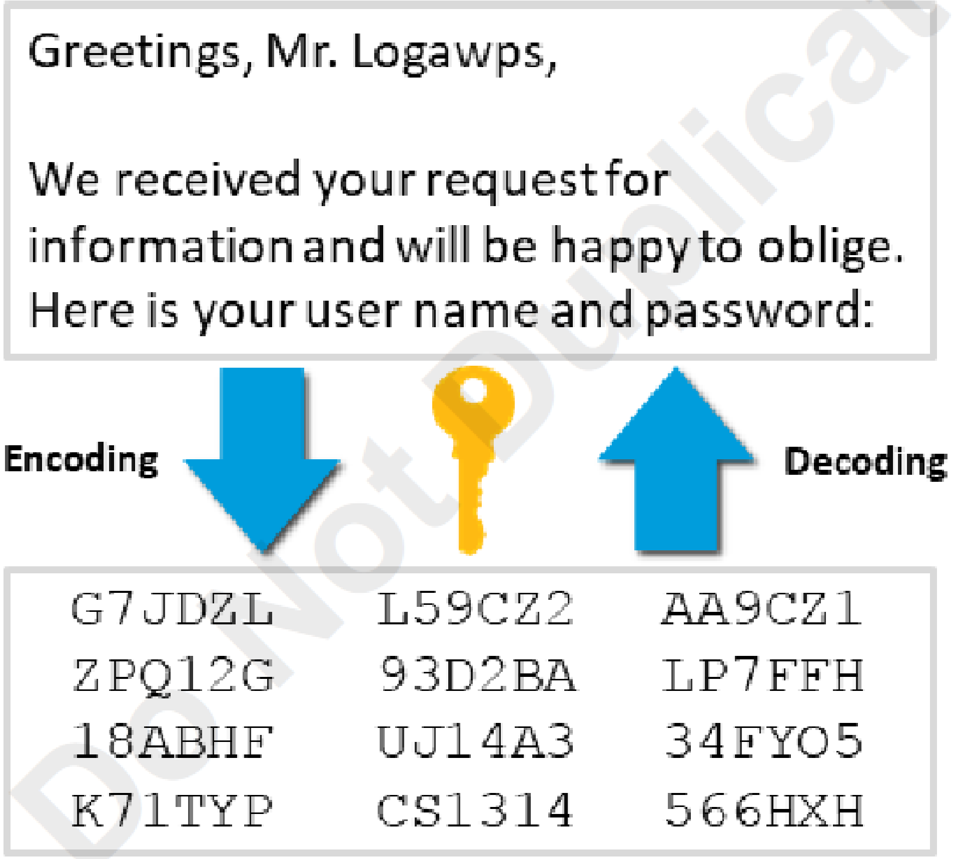
IIT, JU

Agenda

- Goals of Cryptography
- Cryptographic concept
- Symmetric and Asymmetric Cryptography
- Hashing Algorithm
- Data encryption Standard
- Digital signature
- Cryptographic attack

Cryptography

- Cryptography is the science of hiding information, most commonly by encoding and decoding a secret code used to send messages.
- The practice of cryptography is thought to be nearly as old as the written word. Current cryptographic science has its roots in mathematics and computer science, and relies heavily upon technology.



Greetings, Mr. Logawps,

We received your request for information and will be happy to oblige. Here is your user name and password:

G7JDZL	L59CZ2	AA9CZ1
ZPQ12G	93D2BA	LP7FFH
18ABHF	UJ14A3	34FYO5
K71TYP	CS1314	566HXH

Figure 1-20: Cryptography.

Cont…

- We can define cryptography as:
  - Cryptography is a physical process that scrambles information by rearrangement and substitution of content, making it unreadable to anyone except the person capable of unscrambling it.

General Idea behind Cryptography

- In cryptography, the message “Happy Valentine’s Day” may be concealed by-
  - ❖ substituting or replacing or shifting each symbol with another. For example, we can scramble the above message as “Gzoox Uzkdmshmd’r Czx” through shifting to one character before the actual character. If the symbols are digits, we can replace 3 with 7, 2 with 6 and son on.
  - ❖ changing or transposing the location of the symbols of each word of the message, such as “Pyaph Tniv’saelne Yda”.
- Figure below illustrate the process.

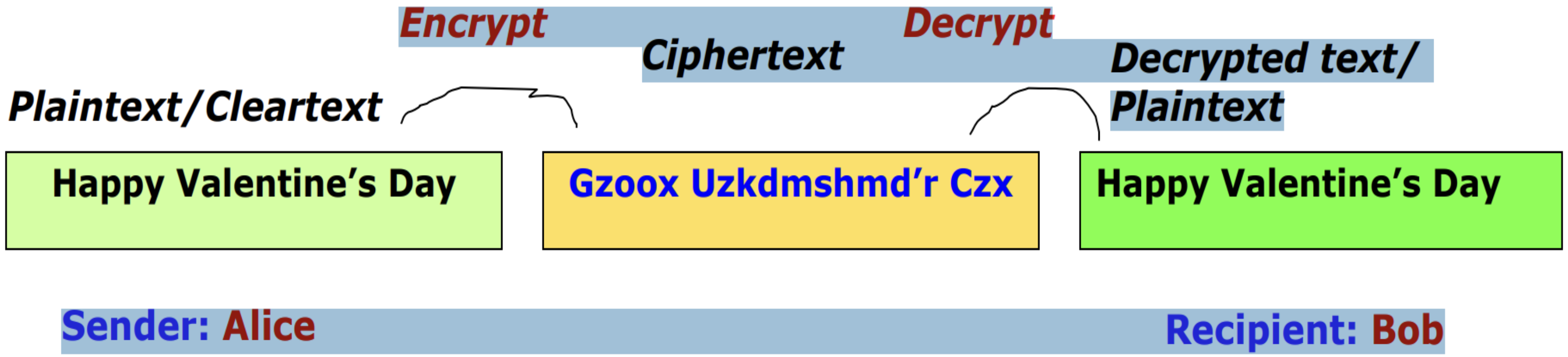


Figure: General idea behind cryptography

Four Security Needs provided by Cryptography

- Modern cryptography concerns itself with the following four objectives:

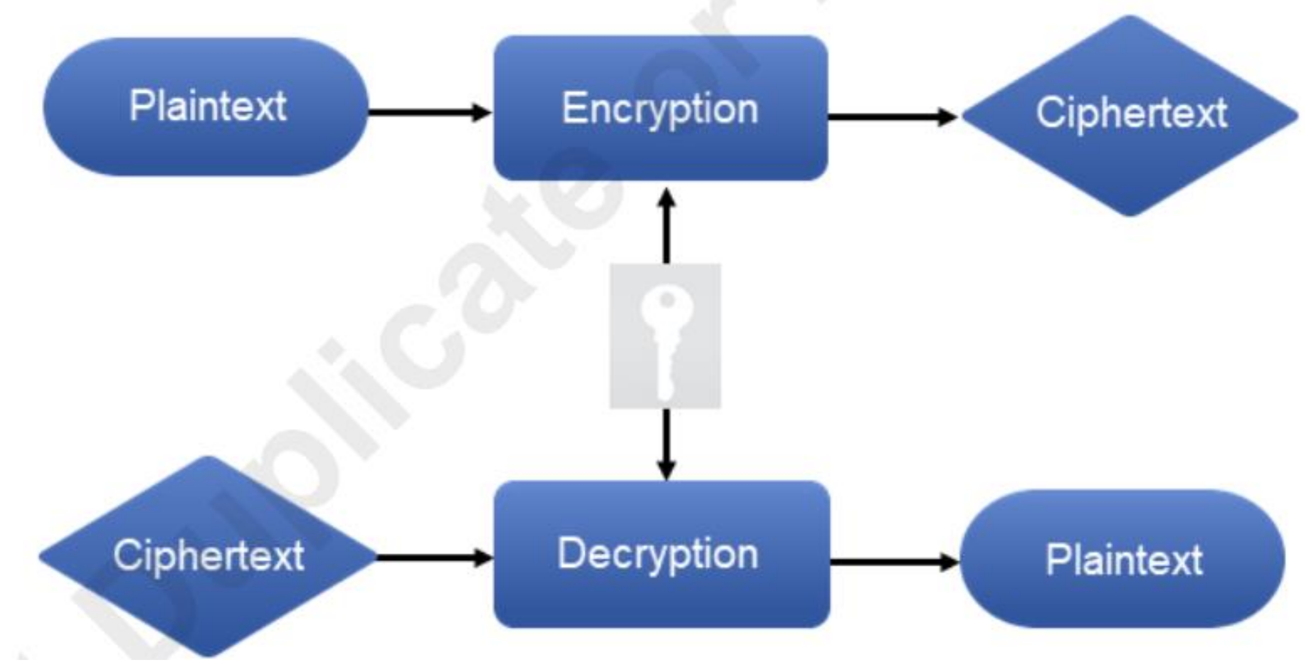
  1. **Confidentiality**: The information cannot be understood by anyone for whom it was not intended.
  2. **Integrity**: The information cannot be altered in storage or transit between sender and intended receiver without the alteration being detected.
  3. **Non-repudiation**: The creator/sender of the information cannot deny at a later stage their intentions in the creation or transmission of the information.
  4. **Authenticatio**n: The sender and receiver can confirm each other's identity and the origin/destination of the information.

Parts of Cryptographic System

- Generally, all cryptographic processes have four basic parts:
  1. **Plain Text**
  2. **Cipher Text**
  3. **Cryptographic Algorithm**
  4. **Key**

Encryption & Decryption

- **Encryption** is a cryptographic technique that converts data from plaintext form into coded, or ciphertext, form.
- **Decryption** is the companion technique that converts ciphertext back to plaintext. While the terms plaintext and clear text are both common cryptographic terms for unencrypted tex



Plaintext

Encryption

Ciphertext

Ciphertext

Decryption

Plaintext

Plain Text and Cipher Text

- **Plaintext/ Cleartext:**
  - It is the original message that is being protected.
- **Ciphertext/ Encoded text/ Encrypted text:**
  - It is the encoded message which is the result of transforming a plaintext using encryption.
- A **cipher** is an algorithm used to encrypt or decrypt data. Algorithms can be simple mechanical substitutions, but in electronic cryptography, they are generally complex mathematical functions.

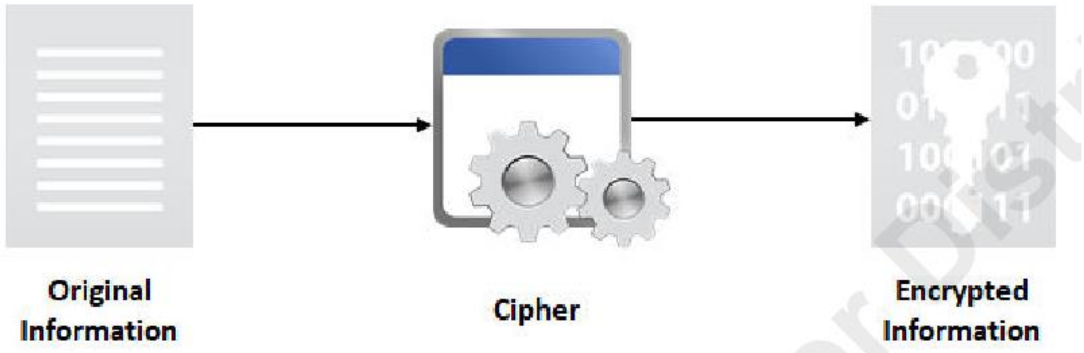
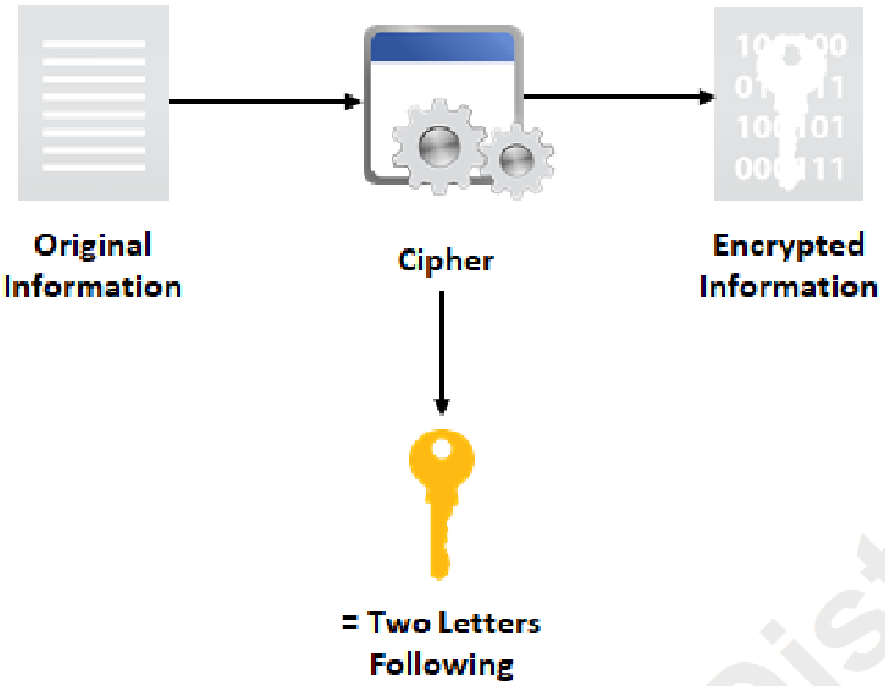


Figure 1-22: Ciphers.

Keys



- An encryption key is a specific piece of information that is used in conjunction with an algorithm to perform encryption and decryption. A different key can be used with the same algorithm to produce different cipher text.
- As the size of key, used to encrypt a message, increases, so does the difficulty in deciphering the message.

Types of Cryptography

- There are two main types of cryptography:
  1. Single key or secret key or symmetric-key cryptography
  2. Public key or asymmetric-key cryptography

Symmetric-Key Cryptography

- **Single Key:** A single secret key is used for both encryption and decryption. Imagine a padlock with one key that unlocks it and locks it.
- **Efficiency:** Symmetric algorithms are generally faster and more efficient than asymmetric algorithms, making them suitable for encrypting large amounts of data.
- **Key Management:** The biggest challenge is securely sharing the secret key between authorized parties. If someone intercepts the key, they can decrypt all communication.

Examples of Symmetric Algorithms:

- **Advanced Encryption Standard (AES):** The current industry standard for symmetric encryption, known for its security and efficiency.
- **Data Encryption Standard (DES):** An older algorithm that has been superseded by AES due to its shorter key length.
- **Triple DES (3DES):** A more secure version of DES that applies the DES algorithm three times for enhanced security.

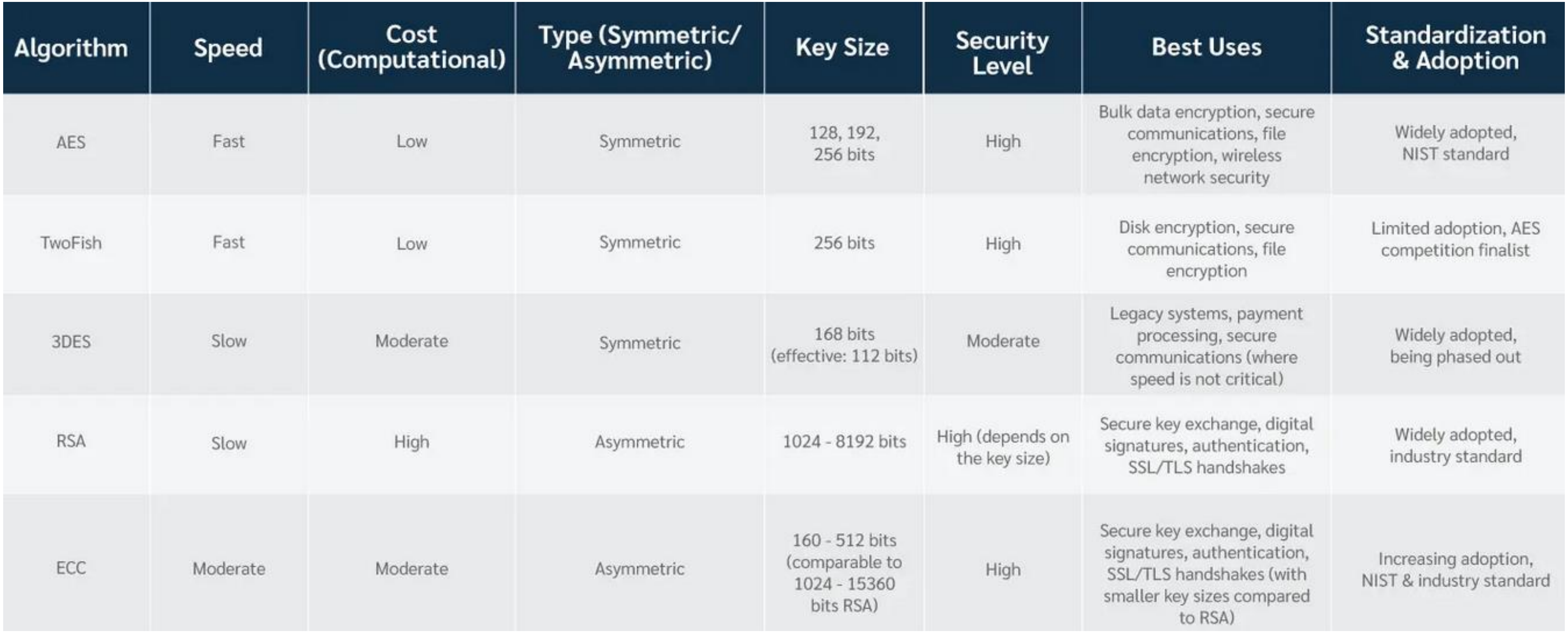
Asymmetric-Key Cryptography

- **Key Pair:** Uses a pair of mathematically linked keys: a public key for encryption and a private key for decryption. Think of it as a mailbox with a public slot for anyone to deposit messages and a private key you hold to unlock the mailbox.
- **Secure Key Distribution:** The public key can be widely distributed without compromising security. Anyone can encrypt messages with the public key, but only the holder of the private key can decrypt them.
- **Computational Cost:** Asymmetric algorithms are slower than symmetric algorithms, making them less suitable for bulk encryption.

Examples of Asymmetric Algorithms:

- **Rivest–Shamir–Adleman (RSA):** A widely used asymmetric algorithm for secure key exchange and digital signatures.
- **Elliptic Curve Cryptography (ECC):** A newer and more efficient alternative to RSA that offers similar security levels with smaller key sizes.

Types of Encryption Algorithms



Algorithm	Speed	Cost (Computational)	Type (Symmetric/ Asymmetric)	Key Size	Security Level	Best Uses	Standardization & Adoption
AES	Fast	Low	Symmetric	128, 192, 256 bits	High	Bulk data encryption, secure communications, file encryption, wireless network security	Widely adopted, NIST standard
TwoFish	Fast	Low	Symmetric	256 bits	High	Disk encryption, secure communications, file encryption	Limited adoption, AES competition finalist
3DES	Slow	Moderate	Symmetric	168 bits (effective: 112 bits)	Moderate	Legacy systems, payment processing, secure communications (where speed is not critical)	Widely adopted, being phased out
RSA	Slow	High	Asymmetric	1024 - 8192 bits	High (depends on the key size)	Secure key exchange, digital signatures, authentication, SSL/TLS handshakes	Widely adopted, industry standard
ECC	Moderate	Moderate	Asymmetric	160 - 512 bits (comparable to 1024 - 15360 bits RSA)	High	Secure key exchange, digital signatures, authentication, SSL/TLS handshakes (with smaller key sizes compared to RSA)	Increasing adoption, NIST & industry standard

Different reasons of usage of Asymmetric-Key Cryptography

- In asymmetric-key cryptography, the public-private key-pairs can be used in two different ways:
    1. **To provide message confidentiality**
    2. **To prove the authenticity of the message originator**

Providing Authenticity of the Message Originator

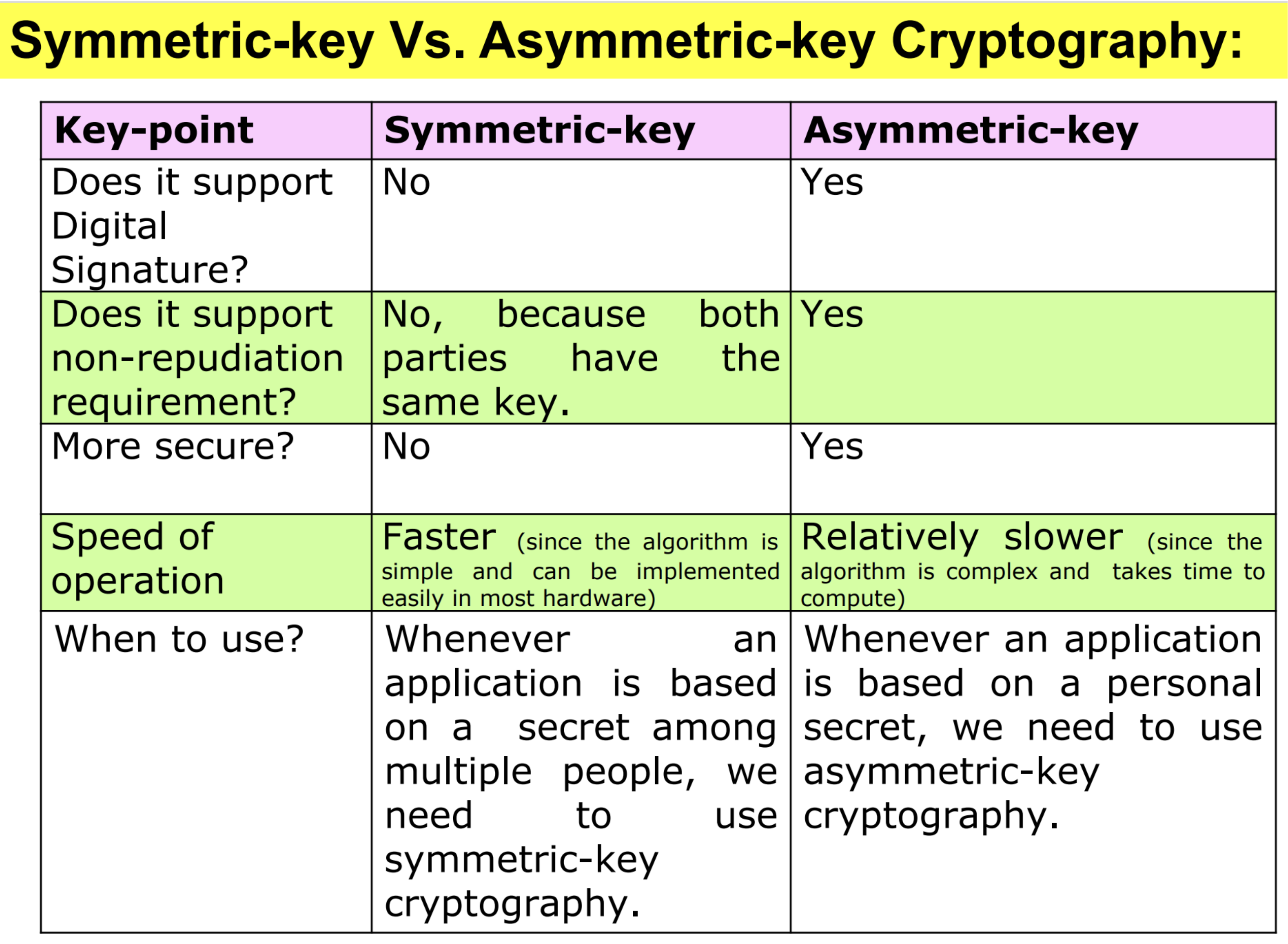
- In this way of private-public key-pairs, data encrypted with the private key can only be decrypted with the public key.
- **Use asymmetric-key encryption for authentication:**
  - Here, data is encrypted by the sender using his/her private key.
  - The private key is kept secret.
  - Data can only be decrypted by anyone using sender's public key. o The public key is freely distributed.
  - Because you are the only person who can encrypt an electronic document with your private key, anyone using your public key to decrypt the message is certain that the message really came from you.
- **Example:**
  - For example, Rassel is an e-customer. He wants to be sure that he is dealing with a legitimate vendor. Similarly, the vendor wants to make sure that Rassel is really Rassel.

Providing Message Confidentiality or Message Privacy:

- In this way of private-public key-pairs, data encrypted with the public key can only be decrypted with the corresponding private key.
- Here, data is encrypted by the sender using the recipient's public key.
- The public key is freely distributed.
- Data can only be decrypted by the recipient's private key. The private key is kept secret.
- Therefore, the data or message remains confidential or private until decoded by the recipient with his/her private key.
- **Example:**
  - Suppose that, Rassel wants to send a confidential message to Ellen. He would first acquire Ellen’s public key. Then he would use that key to encrypt the message and send it to her. If a third party intercepts the message and tries to decode it using Allen’s public key, it would not work. Because only Ellen has the private key, only she can decrypt it. If Allen wants to send a reply, she would use Rassel’s public key and Rassel would use his private key to decrypt it.

Symmetric-key Vs. Asymmetric-key Cryptography:

Key-point	**Symmetric-key**	**Asymmetric-key**
Invention	More than 2000 years (at least in primitive form)	In the mid 1970's
No. of key(s) used	Uses a single shared secret key.	Uses two separate keys: one private and one public.
Is same key used for both encryption and decryption?	Yes	No
Key length	Use shorter keys	Use longer keys
Is distribution of key easy?	Not so easy	Easy



Symmetric-key Vs. Asymmetric-key Cryptography:

Key-point	Symmetric-key	Asymmetric-key
Does it support Digital Signature?	No	Yes
Does it support non-repudiation requirement?	No, because both parties have the same key.	Yes
More secure?	No	Yes
Speed of operation	Faster (since the algorithm is simple and can be implemented easily in most hardware)	Relatively slower (since the algorithm is complex and takes time to compute)
When to use?	Whenever an application is based on a secret among multiple people, we need to use symmetric-key cryptography.	Whenever an application is based on a personal secret, we need to use asymmetric-key cryptography.

Hashing

- Hashing is a process or function that transforms plaintext into ciphertext that cannot be directly decrypted.
- The result of the hashing process is called a hash, hash value, or message digest.
- The input data can vary in length, whereas the hash length is fixed.

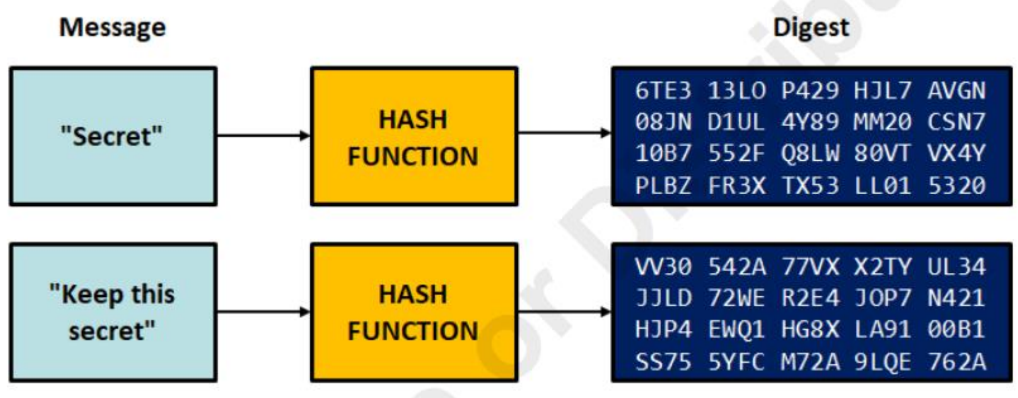


Figure 1-26: Hashing.

Three-pass Protocol

- Besides symmetric-key and asymmetric-key cryptography, there is another protocol that one can use to send sensitive information across an insecure network.
- This protocol is called **three-pass protocol** which does not involve sending keys across the network.
- An analogy can help explain the three-pass protocol:
  - If Alice wants to send a secret message to Bob, she can send it in a box with his padlock.
  - When Bob receives the box, he sends it back to Alice with a padlock of his own.
  - After receiving the box, Alice removes her padlock and returns the box to Bob.
  - Bob can now open the box because it has only his padlock on it.

Common Cryptographic Algorithms

- **RSA algorithm:**
  - It is the most commonly used public-key algorithm, although it is vulnerable to attack.
  - It is named so after its inventors, Ron Rivest, Adi Shamir, and Len Adlemman of the Massachusetts Institute of Technology (MIT).
  - It was first published in 1978.
  - This algorithm lets you choose the size of your public key.
  - The 512-bit keys are considered insecure or weak, but the 768-bit keys are secure from everything but the National Security Administration (NSA).
  - The 1024-bit keys are secure from everything virtually.
  - RSA is embedded in major products such as Windows, Netscape Navigator etc.

Cont..

- **DES (data Encryption Standards):**
  - It was developed by IBM in 1974.
  - DES is the first private-key encryption system which is widely used commercially
- **3DES:**
  - Stronger version of DES called Tripple DES, uses three 56-bit key to encrypt each block.
  - The first key encrypts the data block, the second key decrypts the data block and the third key encrypts the same data block again.
  - The 3DES version requires a 168-bit key that makes the process quite secure and much safer than the plain DES.
- **IDEA (International Data Encryption Algorithm):**
  - It was created in Switzerland in 1991.
  - It offers strong encryption using a 128-bit key to encrypt 64-bit blocks. This system is widely used in older version of PGP (Pretty Good Privacy) system.

Advantage of Cryptograpy

- The advantages of Cryptography are:
    - It hides the message and your privacy is safe.
    - No one would be able to know what it says unless there's a key to the code.
    - You can write what ever you want and how ever you want (any theme any symbol for the code) to keep your code a secret.
    - You are able to use Cryptography during lessons without the teacher knowing. (But will take long to make the code, to figure it out and to make the key).

Disadvantage of Cryptograhy

- The disadvantages of Cryptography are:
  - Encryption takes longer computer processor time to create the code. The more complex the encryption, the more processing it will take.
  - Takes a long time to figure out the code.
  - Encryption keys can become lost rendering the associated data unrecoverable.

Key Exchanges

- Key exchange is any method by which cryptographic keys are transferred between entities, thus enabling the use of an encryption algorithm

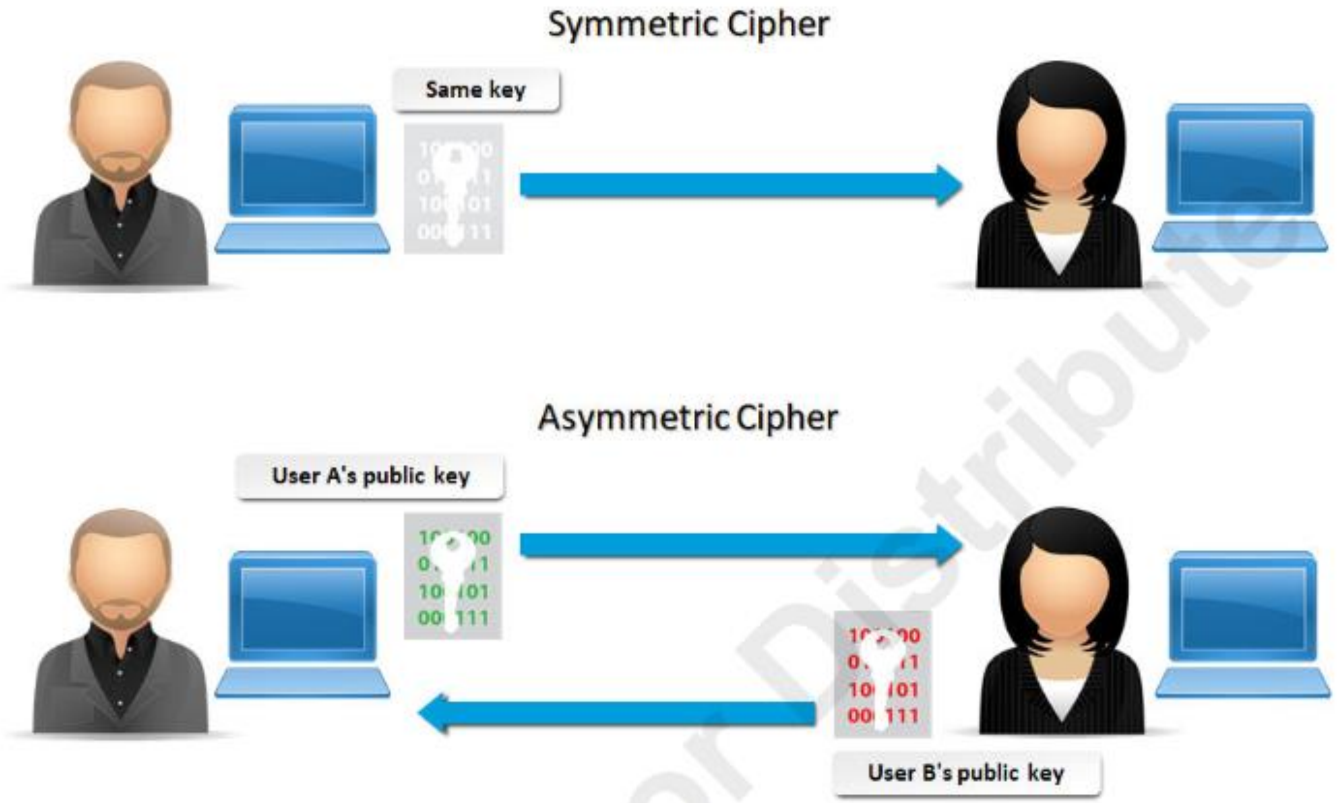


Figure 8-1: Key exchange in symmetric and asymmetric cryptography.

Exchange of Symmetric-key

- In symmetric-key cryptography, Alice and Bob use the same key for communication on the other direction. This key must be protected from access by others.
- However, Alice may need to communicate with another person, say David. Then she needs another secret key. The more keys Alice uses, the more complexity may arise to handle those keys.
- Furthermore, frequently key exchanges are usually desirable to limit the amount of data compromised if an attacker learns the key.
- Therefore, the strength of any cryptographic system rests with the key distribution technique which refers to the delivering of a key to two parties who wish to exchange data, without allowing others to see the key.
- The shared key can be exchanged between involved parties by the following ways:
  - **Face to face** (Alice can select a key and physically deliver it to Bob).
  - **Trusted third party** (A trusted third party can select the key and physically deliver it to Alice and Bob. For example, if Alice and Bob each has an encrypted connection to a third party, say David, then David can deliver a key on the encrypted links to Alice and Bob).
  - **Envelope it using asymmetric ciphers** (If Alice and Bob have previously and recently used a key, one party can transmit the new key to the other by encrypted using the old key.

Traditional Symmetric-key Cipher

- Traditional symmetric-key ciphers can be classified into two broad categories:

1. Substitution Cipher
2. Transposition Cipher.

Substitution Cipher

- A substitution cipher replaces one symbol with another.
- For example, we can replace letter A with letter D, and letter T with letter Z.
- If the symbols are digits, we can replace 3 with 7, 2 with 6.

Transposition Cipher

- A transposition cipher does not substitute one symbol for another, instead it changes the location of the symbols.
- A symbol in the first position of the plaintext may appear in the ninth position of the ciphertext.
- A symbol in the eighth position of the plaintext may appear in the first position of the ciphertext.
- For example, the plaintext characters "**hello**" may be encrypted as "**elhol**".

Modern Symmetric-key Cipher

- The traditional symmetric-key ciphers are character-oriented ciphers.
- Now-a-days, the information to be encrypted is not just text; it can also consist of numbers, graphics, audio, and video data.
- It is convenient to convert these types of data into a stream of bits, to encrypt the stream, and then to send the encrypted stream.
- So, we need **bit-oriented ciphers**.
- When data is treated as the collection of bits, it becomes larger. Mixing a larger number of symbols increases security
- **Kinds of Modern Symmetric-key Ciphers:**
    - Stream ciphers
    - Block ciphers

Block ciphers are primarily used on the Internet, while stream ciphers are used where resources are limited, such as in embedded systems or Bluetooth, although there are exceptions to this rule.

Stream Cipher

- Stream cipher encrypts a single character or bit of plaintext at a time.
- It also decrypts a single character or bit of ciphertext at a time.
- Both the encryption and decryption are performed using the same key.

Example:

- Given plaintext: 10011011110100001
    - Let the keystream be a stream of 1s and 0s.
- If we use an exclusive or (XOR) with the keystream and plaintext, we get ciphertext
- This keystream is called **periodic**, since the sequence '10' repeats over and over.

Plaintext : 10011011110100001

Keystream:10101010101010101

Ciphertext : 00110001011110100 (by XORing each plaintext bit with corresponding keystream bit)

- To decrypt this ciphertext, all we need to do is again XOR the ciphertext with the keystream:

Ciphertext : 00110001011110100

Keystream : 10101010101010101

Plaintext (XOR) : 10011011110100001

Scenario

Alice wants to send a confidential message to Bob over an insecure communication channel. They decide to use a stream cipher for encryption.

Step 1: Key Generation and Initialization:

1. Alice and Bob agree on a secret key beforehand. Let's say the key they agree upon is: 10101010
2. The stream cipher (for this example, let's use a simplified version) initializes its internal state based on this key.

Step 2: Encryption Process:

1. Alice wants to send the message "HELLO" to Bob.
2. The stream cipher generates a keystream based on the agreed-upon key. In our simplified example, let’s assume the keystream is: 01100110

Scenario(Cont..)

Step 3: Encrypting the message:

1. Each character in the message "HELLO" is converted to its binary representation:

- 'H' (ASCII 72) -> 01001000
- 'E' (ASCII 69) -> 01000101
- 'L' (ASCII 76) -> 01001100
- 'L' (ASCII 76) -> 01001100
- 'O' (ASCII 79) -> 01001111

2. Now, XOR each byte of the message with the corresponding byte of the keystream:

- 'H' (01001000) XOR keystream (01100110) = 00101110 (ASCII 46, character '.’)
- 'E' (01000101) XOR keystream (01100110) = 00100011 (ASCII 35, character '#’)
- 'L' (01001100) XOR keystream (01100110) = 00101010 (ASCII 42, character '*’)
- 'L' (01001100) XOR keystream (01100110) = 00101010 (ASCII 42, character '*’)
- 'O' (01001111) XOR keystream (01100110) = 00101001 (ASCII 41, character ')’)
- So, the encrypted message would be: ".#**)"

Scenario(Cont..)

Step 4: Sending the Encrypted Message:

- Alice sends the encrypted message ".#**)" to Bob over the insecure communication channel.

Step 5:Dycription Process:

- Bob receives the encrypted message ".#**)".
- Both Alice and Bob use the same keystream (generated from the shared key) to decrypt the message.
- They apply XOR operation between each byte of the encrypted message and the corresponding byte of the keystream:
- '.' (ASCII 46) XOR keystream (01100110) = 'H' (01001000)
- '#' (ASCII 35) XOR keystream (01100110) = 'E' (01000101)
- '*' (ASCII 42) XOR keystream (01100110) = 'L' (01001100)
- '*' (ASCII 42) XOR keystream (01100110) = 'L' (01001100)
- ')' (ASCII 41) XOR keystream (01100110) = 'O' (01001111)
- After decryption, Bob reconstructs the original message **"HELLO".**

Stream Cipher (Summery)

- **Encryption:**
  - Alice uses the keystream generated from the shared secret key to encrypt each byte of her message.
- **Decryption:**
  - Bob uses the same keystream (generated from the same shared secret key) to decrypt the encrypted message and recover the original plaintext.
- **Keystream:**
  - The stream cipher's security relies heavily on the randomness and secrecy of the keystream generated from the shared key.
  - If an attacker does not know the key, they cannot easily decrypt the message.

Block Ciphers

- A symmetric-key modern block cipher encrypts an **n-bit** block of plaintext or decrypts an **n-bit** block of ciphertext together using the same secret key.
- The common values of **n** are 64, 128, 256, or 512 bits.
  - If the message has the fewer than **n bits**, padding must be added to make it an **n-bit block**.
  - If the message has more than **n bits**, it should be divided into **n-bit blocks** and the appropriate padding must be added to the last block if necessary.

Example:

Plaintext : The only thing we have to fear is fear itself

Modified plaintext : Theonlythingwehavetofearisfearitself

Plaintext blocks : Theonlyt hingweha vetofear isfearit selfXend (break the plaintext into 8-character block)

Ciphertext blocks : tylnoehT ahewgnih raefotev tiraefsi dneXfles (just reverse each plaintext block)

Ciphertext : tylnoehTahewgnihraefotevtiraefsidneXfles

Modes of Operation

- **ECB (Electronic Codebook) mode:**
  - Each block is encrypted independently.
  - This mode is less secure because identical plaintext blocks will produce identical ciphertext blocks.
- **CBC (Cipher Block Chaining) mode:**
  - Each block of plaintext is XORed with the previous ciphertext block before being encrypted, enhancing security by ensuring that even identical plaintext blocks produce different ciphertexts.

How ECB works

1. **Divide Plaintext into Blocks:**
   - The plaintext is divided into fixed-size blocks (e.g., 128-bit blocks if using AES-128).
   - If the last block is smaller than the block size, padding is typically added to make it fit.
2. **Encrypt Each Block Separately:**
   - Each plaintext block is encrypted independently using the same key and the same block cipher algorithm.
   - The output of this encryption is a corresponding ciphertext block.
3. **No Inter-Block Dependency:**
   - Since each block is treated separately, there is no relationship between how one block is encrypted and the next.
   - This lack of dependency makes ECB simpler but also introduces potential security weaknesses.

HOW CBC works

- **Cipher Block Chaining (CBC)** is a mode of operation for block ciphers.
- It improves security by using an initialization vector (IV) and chaining together blocks of plaintext, meaning each block depends on the previous one.
- This ensures that identical plaintext blocks result in different ciphertext blocks, enhancing security over simpler modes like **ECB** (Electronic Codebook).

Steps for CBC Encryption

1. **Divide the Plaintext into Blocks**:
   - The plaintext is divided into blocks of equal size (e.g., 128 bits for AES).
   - If the last block is smaller than the block size, padding is applied.

2. **Initialization Vector (IV)**:
   - An **IV** is generated randomly and used for the encryption of the first block.
   - It doesn’t need to be kept secret but must be unique for each encryption session.

3. **Encryption Process**:

   For each block P_i of plaintext:

- **First Block**:
  - XOR the first plaintext block P_i with the IV:
    $P_i \oplus IV$
  - Encrypt the result with the block cipher using the key:
  - $C1=E_K(P_i \oplus IV)$

- **Subsequent Blocks**:
    - XOR the next plaintext block P_i with the previous ciphertext block C_{i-1}:
      $P_i \oplus C_{i-1}$
    - Encrypt the result:
      $C_i = E_k(P_i \oplus C_{i-1})$
    - C_i becomes the ciphertext for block i.

4. **Final Ciphertext**:
    - After all blocks are processed, the ciphertext is the concatenation of all the ciphertext blocks.

Block Cipher (Summery)

- **Encryption:**
  - Alice uses the agreed-upon secret key to encrypt each block of plaintext using a block cipher algorithm.
- **Decryption:**
  - Bob uses the same secret key to decrypt each block of the encrypted message and recover the original plaintext.
- **Block Size:**
  - Block ciphers typically operate on fixed-size blocks of plaintext, and each block is processed independently during encryption and decryption.
- **Security:**
  - The security of a block cipher depends on the secrecy of the key and the strength of the encryption algorithm used.