ENCRYPTING A KEY

Imagine I want to send a confidential message, "HELLO," to a friend using encryption.

1. **Cipher**: I choose a cipher, like AES, which is a method for encrypting my message. AES is a specific algorithm that will transform my plaintext into ciphertext.
2. **Cryptographic Key**: I need a cryptographic key to use with AES. This key is a secret value that both I and my friend know. For example, let us say the key is "1234567890123456" (a 128-bit key).
3. **Encryption Process**: I use AES with the key "1234567890123456" to encrypt my message "HELLO." The cipher (AES) applies its encryption algorithm to the plaintext message using the key, producing encrypted ciphertext that looks like a random string of characters.
4. **Decryption Process**: my friend uses the same key "1234567890123456" with AES to decrypt the ciphertext back into the original message "HELLO."

In summary, the cipher (AES) and the cryptographic key work together to ensure that my message is securely encrypted and can be decrypted only by someone who has the same key.

WHAT'S AES?

Modern symmetric ciphers, like AES (Advanced Encryption Standard), use the same key for both encryption and decryption. AES is a block cipher that encrypts data in fixed-size blocks (128 bits) and supports key sizes of 128, 192, or 256 bits. It operates through multiple rounds of processing, including substitution, permutation, and mixing steps, to ensure data confidentiality. AES is widely used due to its efficiency and strong security.

CIPHER

A cipher is a method used to encrypt or decrypt information to keep it secure. It involves a set of rules or algorithms for converting plain text (readable data) into ciphertext (encoded data) and vice versa. Ciphers can be symmetric (same key for encryption and decryption) or asymmetric (different keys for encryption and decryption). They are fundamental in ensuring the confidentiality and integrity of data.

CRYPTOGRAPHIC KEY

A cryptographic key is a piece of information used in algorithms to encrypt or decrypt data. It determines the output of the encryption process and must be kept secret to ensure data security. Keys can be symmetric (same key for both encryption and decryption) or asymmetric (different keys for encryption and decryption), depending on the encryption method used.

//FROM EARLIER DISCUSSION ON DIFFIE -HALLMAN ALGORITHM

SIGNIFICANCE OF “mod” IN KEY EXCHANGE

In key exchange algorithms, "mod" (short for modulus) plays a crucial role in ensuring security and enabling the exchange of cryptographic keys between parties. Here’s a brief explanation of its significance:

1. **Mathematical Operations**: In many key exchange algorithms, such as Diffie-Hellman, the modulus operation is used to handle large numbers by reducing them to a smaller range. This helps in performing modular arithmetic, which is essential for cryptographic computations.
2. **Security**: The modulus operation helps in creating a finite field (a set of numbers with well-defined arithmetic operations) where certain mathematical problems are hard to solve. For instance, in the Diffie-Hellman key exchange, both parties agree on a large prime number (the modulus) and a base. They then use modular exponentiation to compute their keys. The security of this process relies on the difficulty of the discrete logarithm problem in modular arithmetic.
3. **Efficient Computation**: Modular arithmetic allows for efficient computations with very large numbers, which is crucial in cryptography. By using modular operations, the computations remain manageable and efficient even when dealing with large prime numbers.

Overall, the modulus operation is integral to key exchange algorithms as it helps in securely generating and exchanging cryptographic keys while maintaining computational efficiency.

BB84 QUANTUM KEY EXCHANGE

[0 -> | , 1 -> - , + -> / , - -> \]

BB84 is a quantum key distribution (QKD) protocol developed by Charles Bennett and Gilles Brassard in 1984. It allows two parties to securely share a cryptographic key using the principles of quantum mechanics.

1. **Quantum Bits (Qubits)**: BB84 uses quantum bits (qubits) instead of classical bits. Qubits can exist in multiple states simultaneously due to quantum superposition, which adds a layer of security.
2. **Preparation**: One party, Alice, prepares qubits in one of four possible states, represented as “ - “, “ | ” , “ \” , “ / “ . These states are chosen randomly.
3. **Transmission**: Alice sends these qubits to the second party, Bob, over a quantum channel. During transmission, the qubits are encoded in these four states.

4. **Measurement**: Bob measures the received qubits. However, he can measure in one of two bases: the standard basis " + " or the diagonal basis " X ". The choice of basis is also random.
5. **Basis Comparison**: After Bob has measured the qubits, Alice and Bob communicate over a classical channel to compare which bases they used for each qubit. They discard the qubits where their bases didn't match, retaining only the qubits measured in the same basis.
6. **Key Generation**: The remaining qubits, where the bases match, are used to form the shared secret key. Because of the principles of quantum mechanics, any eavesdropping would disturb the qubits, revealing the presence of an eavesdropper and ensuring the security of the key.

   To establish an n bits key, a total of 2.n bits must be sent (on average), since the **probability** of using the same filter orientations is **50%**

BB84's security relies on the fundamental principles of quantum mechanics, such as the no-cloning theorem (which prevents an eavesdropper from copying the qubits) and the disturbance caused by measuring qubits. This makes BB84 a foundational protocol in quantum cryptography, ensuring secure communication.

QUBITS or quantum bits:

These are the fundamental units of information in quantum computing, analogous to classical bits but with unique properties due to quantum mechanics.

1. **Superposition**: Unlike classical bits, which are either 0 or 1, qubits can exist in a state of superposition. This means a qubit can be in a combination of both 0 and 1 states simultaneously. For example, a qubit might be in a state where it is 50% 0 and 50% 1. Like " + " -> '-' and '|' " X " -> '/' , '\' .
2. **Entanglement**: Qubits can also be entangled, meaning the state of one qubit can be dependent on the state of another, no matter how far apart they are. This allows qubits to exhibit correlated behaviours that classical bits cannot.
3. **Quantum Gates**: Quantum operations are performed using quantum gates, which manipulate qubits through operations such as rotations and transformations. These gates are the quantum analogues of classical logic gates but operate on qubits in a way that leverages their superposition and entanglement properties.
4. **Measurement**: When a qubit is measured, it collapses from its superposition state to a definite classical state (either 0 or 1). The outcome of this measurement is probabilistic, based on the qubit's state prior to measurement.

Qubits are central to quantum computing because they enable parallel processing of information and can solve certain problems more efficiently than classical computers.

EXAMPLE OF BB84 QUANTUM KEY EXCHANGE

Overview of BB84 Protocol

1. **Preparation**:
    - Alice prepares qubits in one of four possible states. These states are represented in two bases:
        - **Standard Basis (Vertical and Horizontal)**:
            - **|0⟩** = "|" (classical 0)
            - **|1⟩** = "-" (classical 1)
        - **Diagonal Basis**:
            - **|+⟩** = (|0⟩ + |1⟩) / √2 (diagonal state at 45°) = /
            - **|−⟩** = (|0⟩ - |1⟩) / √2 (diagonal state at 135°) = \
2. **Transmission**:
    - Alice sends these qubits to Bob. For example, she might send:
        - **|0⟩** = "|"
        - **|1⟩** = "-"
        - **|+⟩** = "/"
        - **|−⟩** = "\"
3. **Measurement**:
    - Bob measures each qubit using either the **Standard Basis** or the **Diagonal Basis**.
    - When Bob measures a qubit in the **Standard Basis**, he gets either **|0⟩** or **|1⟩**.
    - When he measures in the **Diagonal Basis**, he gets either **|+⟩** or **|−⟩** .
4. **Basis Comparison**:
    - After measurement, Alice and Bob communicate over a classical channel to compare which bases they used for each qubit.
    - If they used the same basis, they keep the qubit result. If not, they discard it.
5. **Key Generation**:
    - The remaining qubits (those measured in the same basis) form the shared secret key.

Example

1. **Alice's Preparation**:
    - Alice randomly prepares qubits. Suppose she prepares:
        - **|+⟩**
2. **Transmission**:
    - Alice sends **|+⟩** to Bob.
3. **Bob's Measurement**:
    - Bob randomly chooses a basis to measure. If he chooses the Diagonal Basis:
        - He measures and sees **|+⟩** (which matches Alice's preparation).
4. **Basis Comparison**:
    - Alice and Bob compare their bases and find they both used the Diagonal Basis for this qubit.
5. **Key Generation**:
    - They include this qubit's result in their shared secret key.

If Bob had chosen the Standard Basis for measurement, he would get either **|0⟩** or **|1⟩**, which would not match the state **|+⟩**. Thus, they discard the result for that qubit.

This process ensures that even if an eavesdropper intercepts the qubits, their presence will disturb the qubits' states, revealing their attempt to eavesdrop and ensuring the key remains secure.