Private-Key Cryptography

1. **Single Key Usage**: Uses one shared key between sender and receiver.
2. **Symmetric Encryption**: Both parties use the same key, making them equals.
3. **Security Issue**: If the key is leaked, communication can be compromised.
4. **Forging Risk**: The receiver could forge a message and claim it was sent by the sender.

Public-Key Cryptography

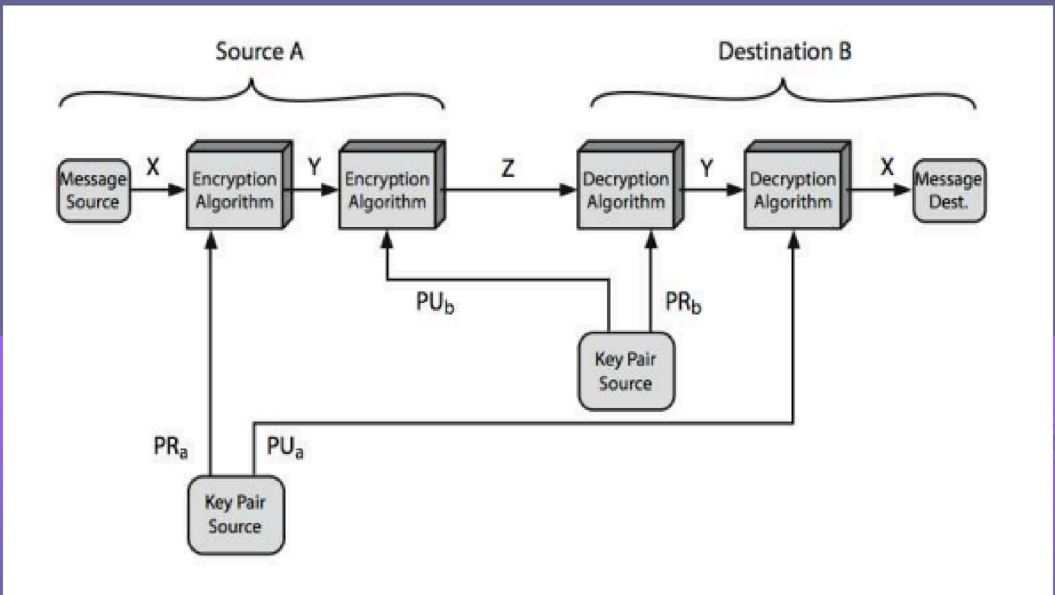
1. **Two Keys**: Utilizes a public key (known to everyone) and a private key (kept secret).
2. **Asymmetric Encryption**: The two parties are not equals since they use different keys.
3. **Mathematical Foundation**: Based on complex number theory, making it secure and functional.
4. **Enhances Security**: Complements, rather than replaces, private-key cryptography.

Symmetric vs Public-Key

Conventional Encryption	Public-Key Encryption
Needed to Work:	*Needed to Work:*
1. The same algorithm with the same key is used for encryption and decryption.	1. One algorithm is used for encryption and decryption with a pair of keys, one for encryption and one for decryption.
2. The sender and receiver must share the algorithm and the key.	2. The sender and receiver must each have one of the matched pair of keys (not the same one).
Needed for Security:	*Needed for Security:*
1. The key must be kept secret.	1. One of the two keys must be kept secret.
2. It must be impossible or at least impractical to decipher a message if no other information is available.	2. It must be impossible or at least impractical to decipher a message if no other information is available.
3. Knowledge of the algorithm plus samples of ciphertext must be insufficient to determine the key.	3. Knowledge of the algorithm plus one of the keys plus samples of ciphertext must be insufficient to determine the other key.

-

Public-Key Cryptosystems



Publickey encryption has 6 integredients.

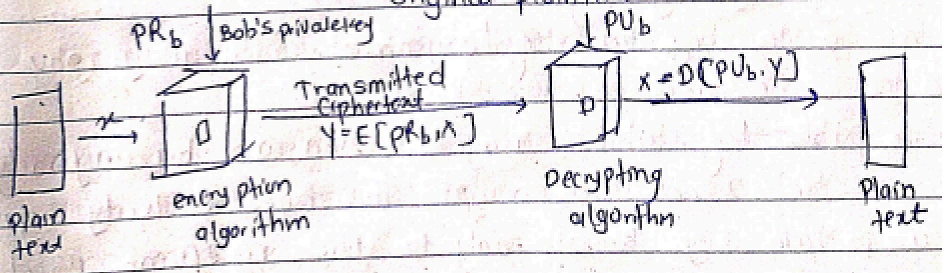
① Plain Text - readable message or data that is fed into the algorithm as input.

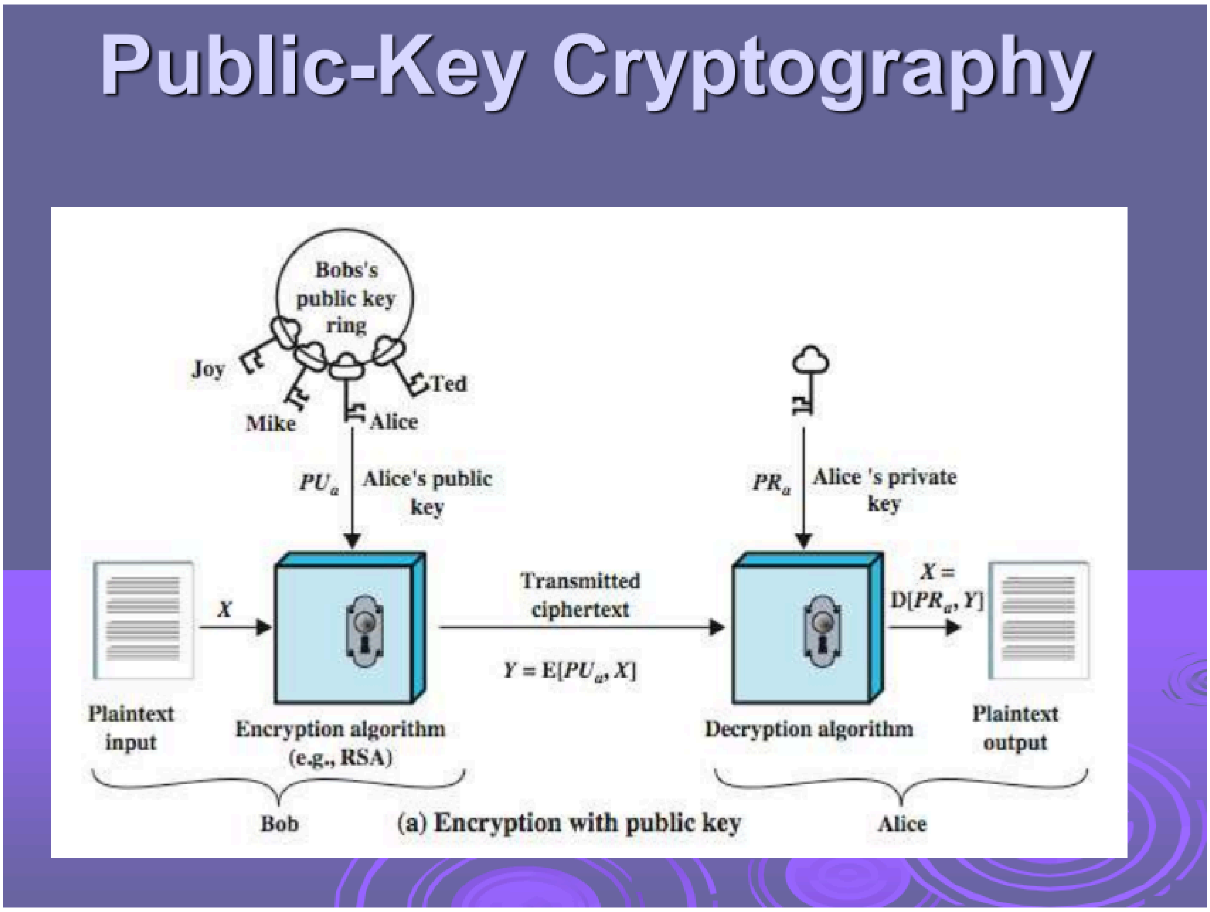
② Encryption algorithm - performs various transformations on the plain text

③ public & private keys - pair of keys that have been selected so that if one is used for encryption, the other is used for decryption.

④ Cipher text - encrypted message produced as output. depends on plaintext & key. for a given message two keys produce two different ciphertext

5) Decryption algorithm - accepts the ciphertext & the matching key, produces the original plaintext.





Why Public-Key Cryptography?

1. **Key Distribution**: Solves the problem of securely distributing keys without needing a trusted third party.
2. **Digital Signatures**: Verifies the integrity and authenticity of a message.
3. **Invention**: Publicly introduced by Whitfield Diffie & Martin Hellman in 1976.

Public-Key Cryptography Mechanism

1. **Two Keys**:
   - **Public Key**: Used to encrypt messages and verify signatures.
   - **Private Key**: Used to decrypt messages and create signatures.
2. **Asymmetry**: Encryption and decryption keys are distinct, making it impossible to deduce one from the other.

Public-Key Cryptosystems Applications

1. **Encryption/Decryption**: Ensures message secrecy.
2. **Digital Signatures**: Provides authentication.
3. **Key Exchange**: Secures the exchange of session keys.

Public-Key Requirements

1. **Infeasibility**: It should be nearly impossible to determine the private key from the public key.
2. **Efficient Operations**: Encryption and decryption must be computationally easy with the correct key.
3. **Trap-Door One-Way Function**: Allows easy computation in one direction but makes the reverse infeasible without a special piece of information (trapdoor).

Security of Public Key Schemes

1. **Brute Force Attack**: Theoretically possible but impractical due to the large key size (e.g., >512 bits).
2. **Relies on Mathematical Complexity**: Difficult problems like factoring large numbers ensure security.
3. **Large Numbers Requirement**: Security depends on using large primes and numbers.

RSA Algorithm

1. **Inventors**: Developed by Rivest, Sh amir, and Adleman in 1977.
2. **Basis**: Works on exponentiation over integers modulo a prime, utilizing large integers (e.g., 1024 bits).
3. **Security**: Relies on the difficulty of factoring large numbers.

RSA Encryption/Decryption Process

1. **Encryption**:
   - **Public Key**: Consists of PU = {e, n}.
   - **Ciphertext**: C = M^e mod n, where M is the message , where 0≤M<n
2. **Decryption**:
   - **Private Key**: Consists of PR = {d, n}.

- **Message Recovery**: M = C^d mod n.

RSA Key Setup

1. **Choose Primes**: Randomly select two large primes p and q.
2. **Compute Modulus**: n = p * q.
3. **Euler's Totient Function**: Calculate ø(n) = (p-1)(q-1).
4. **Choose Encryption Key**: e, such that 1 < e < ø(n) and gcd(e, ø(n)) = 1.
5. **Compute Decryption Key**: d such that e * d = 1 mod ø(n).
6. **Publish their public encryption key:** PU={e,n}
7. **Keep secret private decryption key:** PR={d,n}

Why RSA Works

1. **Based on Euler's Theorem**: Ensures that the encryption and decryption processes correctly recover the original message.

Example RSA Key Setup and Operation

1. **Select Primes**: p = 17, q = 11.
2. **Modulus Calculation**: n = 17 * 11 = 187.
3. **Totient Calculation**: ø(n) = 160.
4. **Choose e**: e = 7.
5. **Compute d**: d = 23 (since 7 * 23 = 161 mod 160).
6. **Public Key**: {7, 187}, **Private Key**: {23, 187}.
7. **Encrypt/Decrypt Example**:
   - Message M = 88.
   - Ciphertext C = 88^7 mod 187 = 11.
   - Decrypt to get M = 11^23 mod 187 = 88.

Efficient Encryption and Decryption

1. **Small Encryption Exponent**: Choosing a small e (e.g., 65537) makes encryption faster.
2. **Chinese Remainder Theorem (CRT)**: Used for faster decryption by breaking down calculations.

RSA Security

1. **Factorization Problem**: The difficulty of factoring the modulus n is central to RSA security.
2. **Countermeasures**:
   - **Timing Attacks**: Avoid by constant-time operations.
   - **Chosen Ciphertext Attacks**: Use padding schemes like OASP for protection.