ASYMMETRIC KEY CRYPTOGRAPHY

Shannon's Theory

- According to the famous information theorist **Claude Shannon**, there are two primitive operations with which strong encryption algorithms can be built: **confusion and diffusion** are two properties of the operation of a secure cipher identified in 1945 classified report ***A Mathematical Theory of Cryptography.***

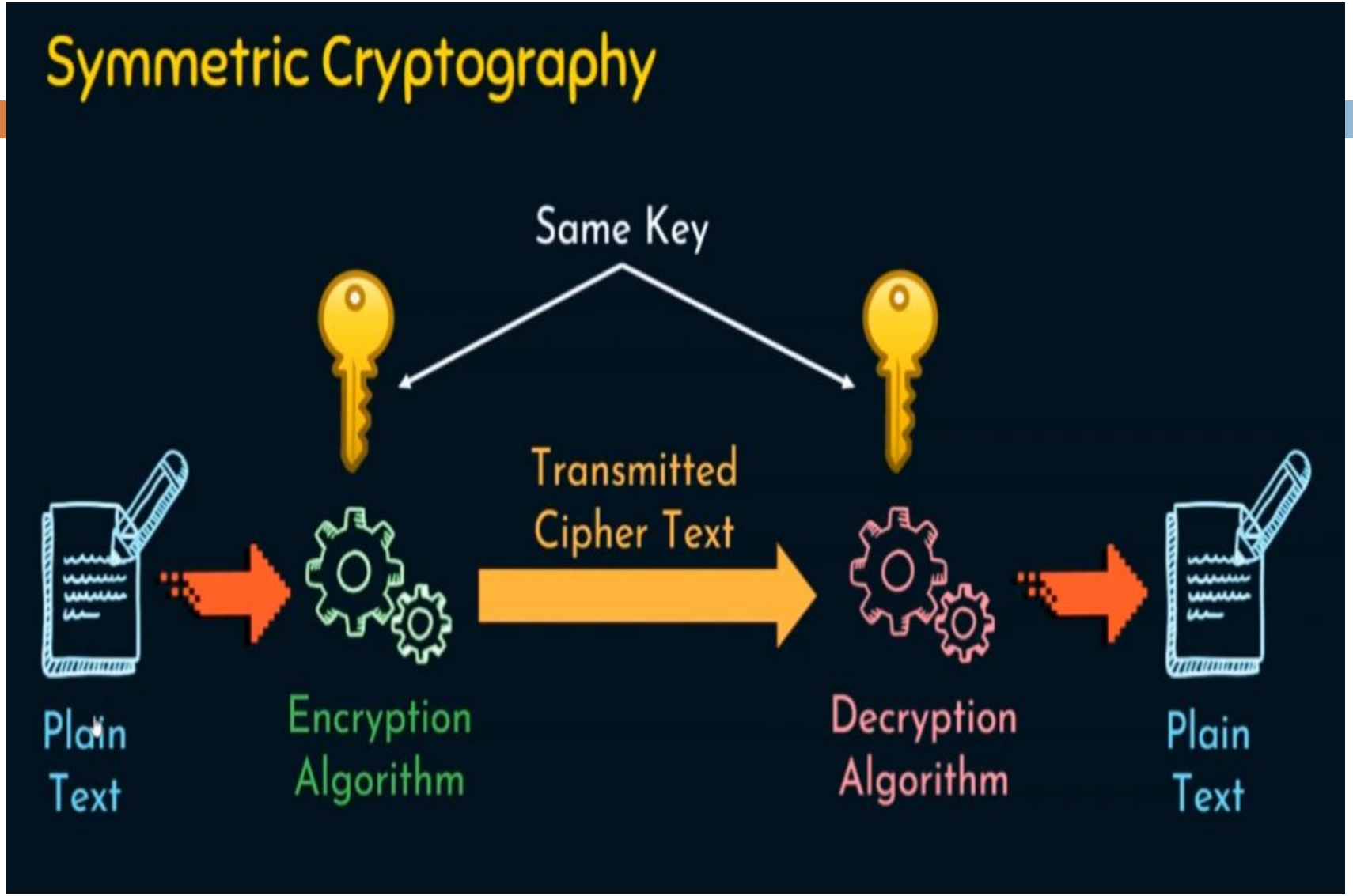
Confusion

- **Confusion** is an encryption operation where the relationship between key and ciphertext is obscured. Today, a common element for achieving confusion is substitution, which is found in both DES and AES.
- This property makes it difficult to find the key from the ciphertext and if a single bit in a key is changed, the calculation of the values of most or all of the bits in the ciphertext will be affected.
- Confusion increases the ambiguity of ciphertext and it is used by both block and stream ciphers.

Diffusion

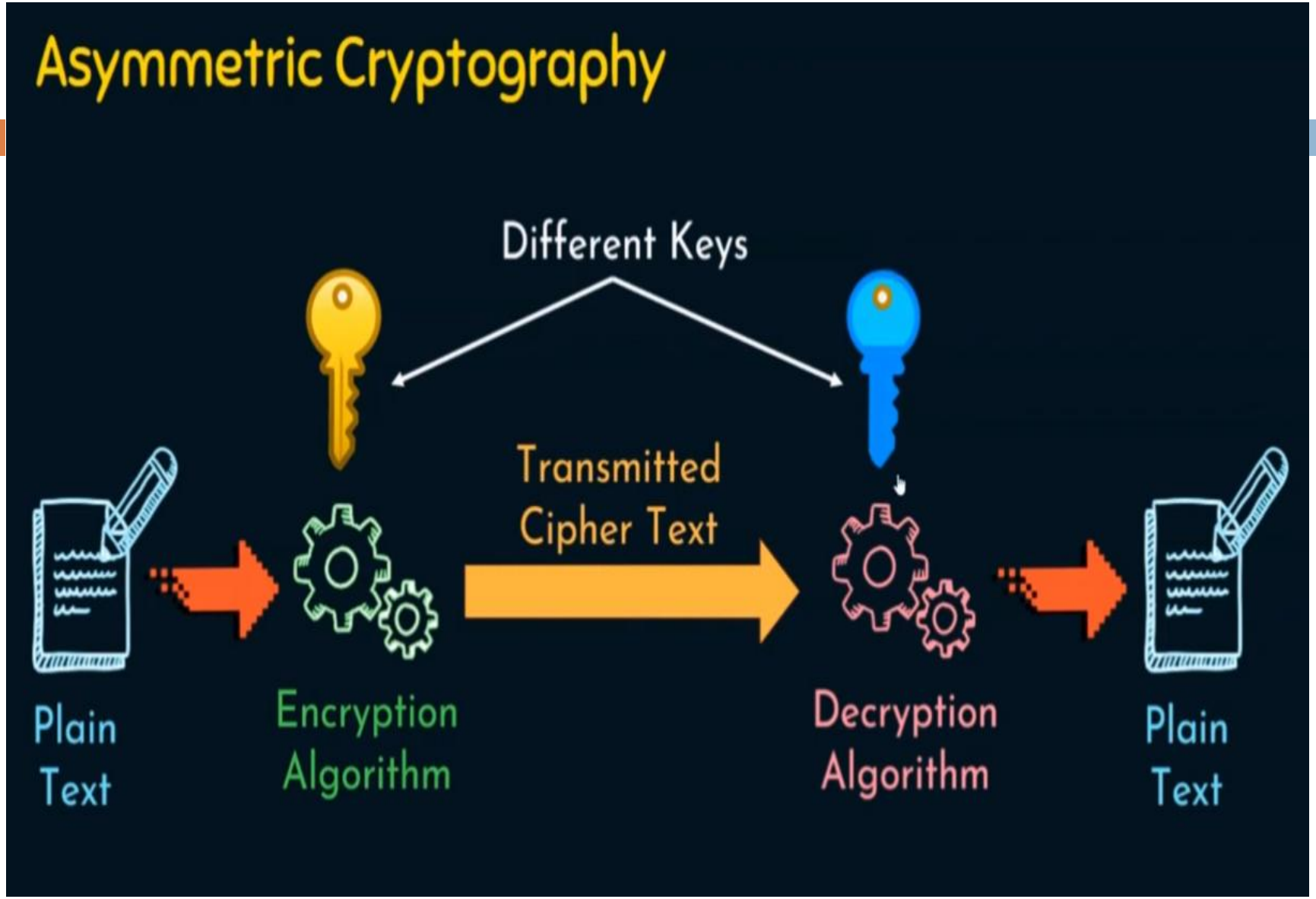
- **Diffusion** is an encryption operation where the influence of one plaintext symbol is spread over many ciphertext symbols with the goal of hiding statistical properties of the plaintext. A simple diffusion element is the bit permutation, which is used frequently within DES.
- Diffusion means that if we change a single bit of the plaintext, then (statistically) half of the bits in the ciphertext should change, and similarly, if we change one bit of the ciphertext, then approximately one half of the plaintext bits should change.
- The idea of diffusion is to hide the relationship between the ciphertext and the plain text. This will make it hard for an attacker who tries to find out the plain text and it increases the redundancy of plain text by spreading it across the rows and columns; it is achieved through transposition of algorithm and it is used by block ciphers only.

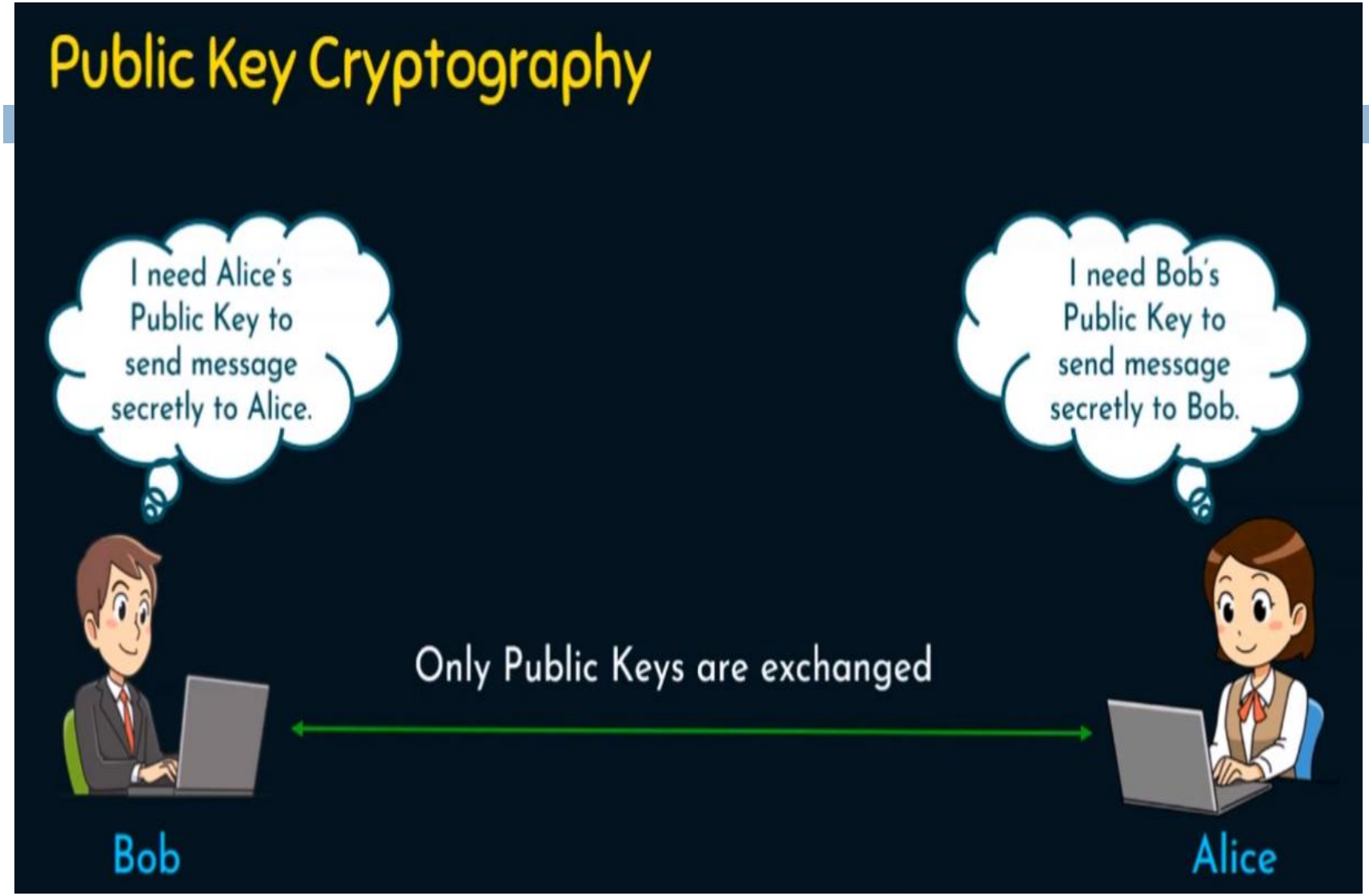
S.NO	CONFUSION	DIFFUSION
1.	Confusion is a cryptographic technique which is used to create faint cipher texts.	While diffusion is used to create cryptic plain texts.
2.	This technique is possible through substitution algorithm.	While it is possible through transportation algorithm.
3.	In confusion, if one bit within the secret's modified, most or all bits within the cipher text also will be modified.	While in diffusion, if one image within the plain text is modified, many or all image within the cipher text also will be modified
4.	In confusion, vagueness is increased in resultant.	While in diffusion, redundancy is increased in resultant.
5.	Both stream cipher and block cipher uses confusion.	Only block cipher uses diffusion.
6.	The relation between the cipher text and the key is masked by confusion.	While The relation between the cipher text and the plain text is masked by diffusion.

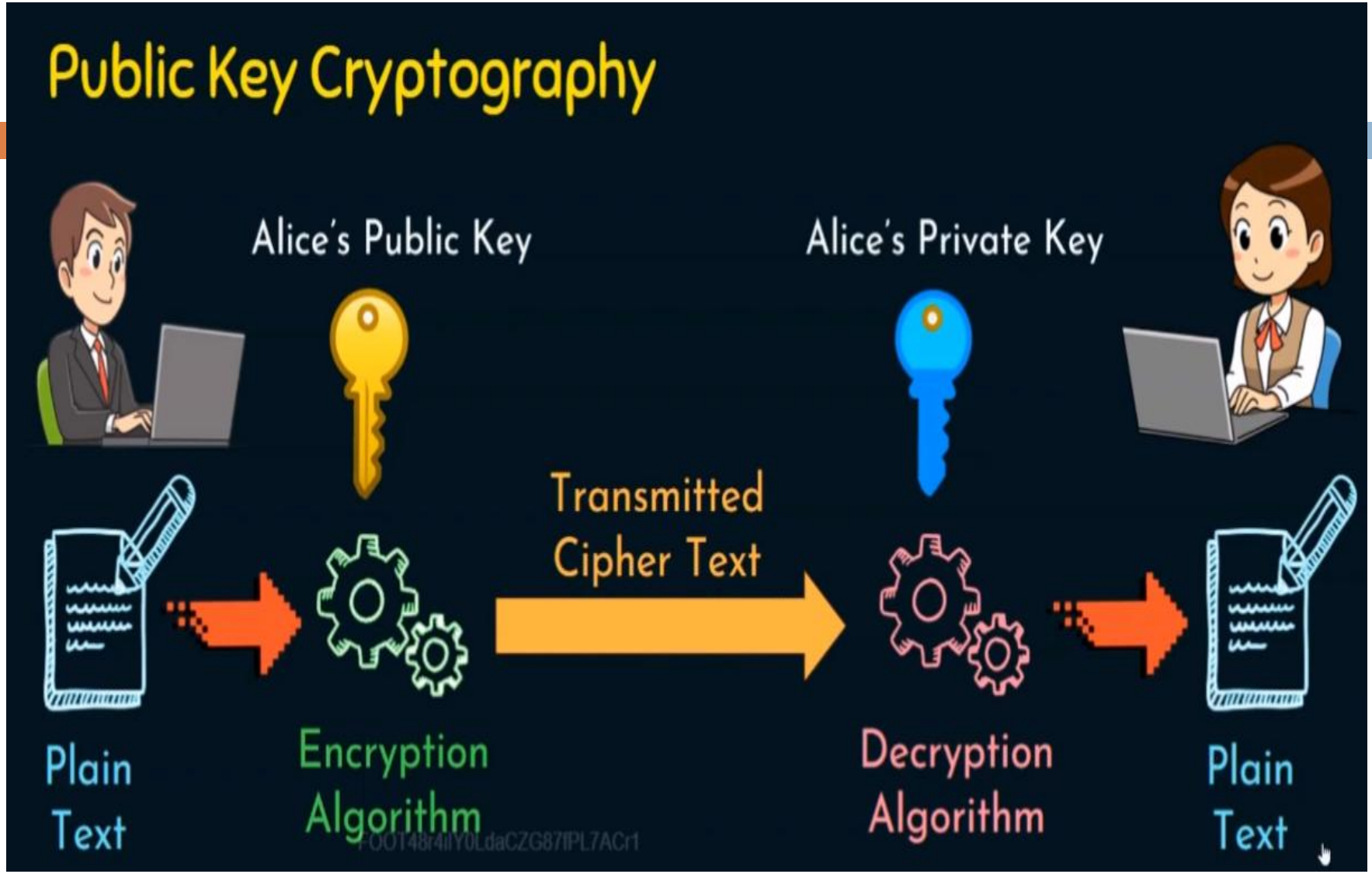

Symmetric Cryptography
Same Key
Transmitted
Cipher Text
Plain
Text
Encryption
Algorithm
Decryption
Algorithm
Plain
Text

Review of Secret Key (Symmetric) Cryptography

- Confidentiality
  - stream ciphers
  - block ciphers with encryption modes
- Integrity
  - Cryptographic hash functions
  - Message authentication code (keyed hash functions)
- Limitation: sender and receiver must share the same key
  - Needs secure channel for key distribution
  - Impossible for two parties having no prior relationship
  - Needs many keys for n parties to communicate


Asymmetric Cryptography
Different Keys
Transmitted
Cipher Text
Plain
Text
Encryption
Algorithm
Decryption
Algorithm
Plain
Text


Public Key Cryptography
I need Alice's Public Key to send message secretly to Alice.
I need Bob's Public Key to send message secretly to Bob.
Only Public Keys are exchanged
Bob
Alice


Public Key Cryptography
Alice's Public Key
Alice's Private Key
Transmitted
Cipher Text
Plain
Text
Encryption
Algorithm
Decryption
Algorithm
Plain
Text

Symmetric Vs Asymmetric Cryptography

Conventional Encryption	Public-Key Encryption
Same Algorithm	Different Algorithm
Same Key	Different Keys
Key is kept secret	One of the keys is kept secret
Faster	Slower
Classical Cryptosystem	RSA, Diffie-Hellman, ECC, Rabin Cryptosystem

Applications of Public-Key Cryptosystems

Algorithm	Encryption/ Decryption	Digital Signature	Key Exchange
RSA	✓	✓	✓
ECC	✓	✓	✓
Diffie-Hellman	✗	✗	✓
DSS	✗	✓	✗

Public Key Encryption Overview

- Each party has a PAIR (K, K^{-1}) of keys:
  - K is the **public** key, and used for encryption
  - K^{-1} is the **private** key, and used for decryption
  - Satisfies $\mathbf{D}_{K^{-1}}[\mathbf{E}_K[M]] = M$
- Knowing the public-key K, it is computationally infeasible to compute the private key K^{-1}
- The public-key K may be made publicly available, e.g., in a publicly available directory
  - Many can encrypt, only one can decrypt
- Public-key systems aka *asymmetric* crypto systems.

Public Key Encryption

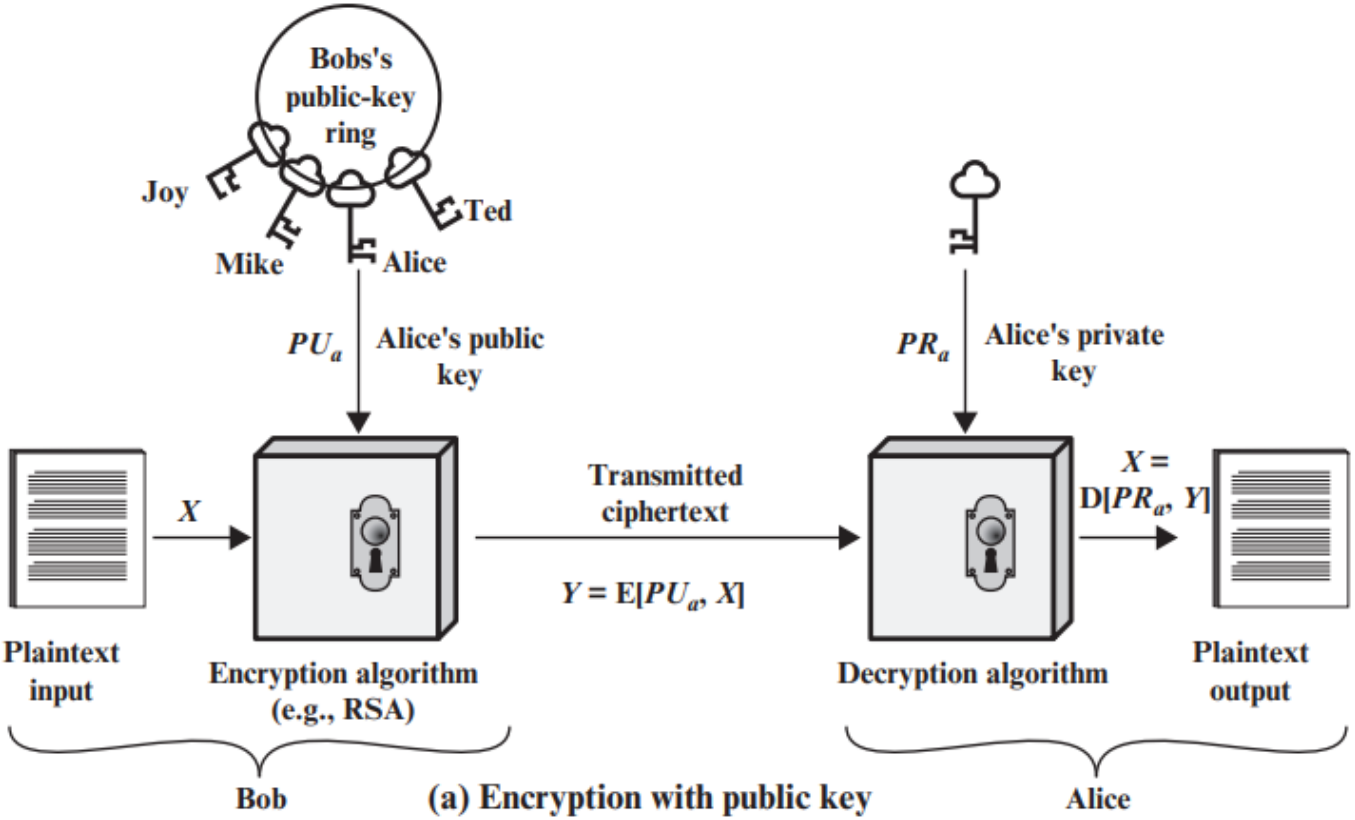
Bob has N-Bit message to send to Alice.

- Alice has public and secret key.
  - PUBLIC key = published on Web in digital phonebook (VERISIGN).
  - PRIVATE key = known only by Alice.
- Bob encrypts message using Alice's public key.
- Alice decrypts message using her private key.

To achieve security, need following properties:

- Can encrypt message efficiently with public key.
- Can decrypt message efficiently with private key.
- CANNOT decrypt message efficiently with public key alone.

Public-Key Cryptosystem: Confidentiality



(a) Encryption with public key

Public-Key Cryptosystem: Confidentiality

- Let us take a closer look at the essential elements of a public-key encryption scheme. There is some source A that produces a message in plaintext, X = [X1, X2, …. , XM]. The M elements of X are letters in some finite alphabet. The message is intended for destination B.
- B generates a related pair of keys: a public key, PUb, and a private key, PRb. PRb is known only to B, whereas PUb is publicly available and therefore accessible by A. With the message X and the encryption key PUb as input, A forms the ciphertext Y = [Y1, Y2, …. , YN]

$$Y = E(PUb, X)$$

- The intended receiver, in possession of the matching private key, is able to invert the transformation:

$$X = D(PRb, Y)$$

Public-Key Cryptosystem: Confidentiality

- An adversary, observing Y and having access to PUb, but not having access to PRb or X, must attempt to recover X and/or PRb. It is assumed that the adversary does have knowledge of the encryption (E) and decryption (D) algorithms.
- If the adversary is interested only in this particular message, then the focus of effort is to recover X by generating a plaintext estimate X .
- Often, however, the adversary is interested in being able to read future messages as well, in which case an attempt is made to recover PRb by generating an estimate PRb.

Public-Key Cryptosystem: Confidentiality

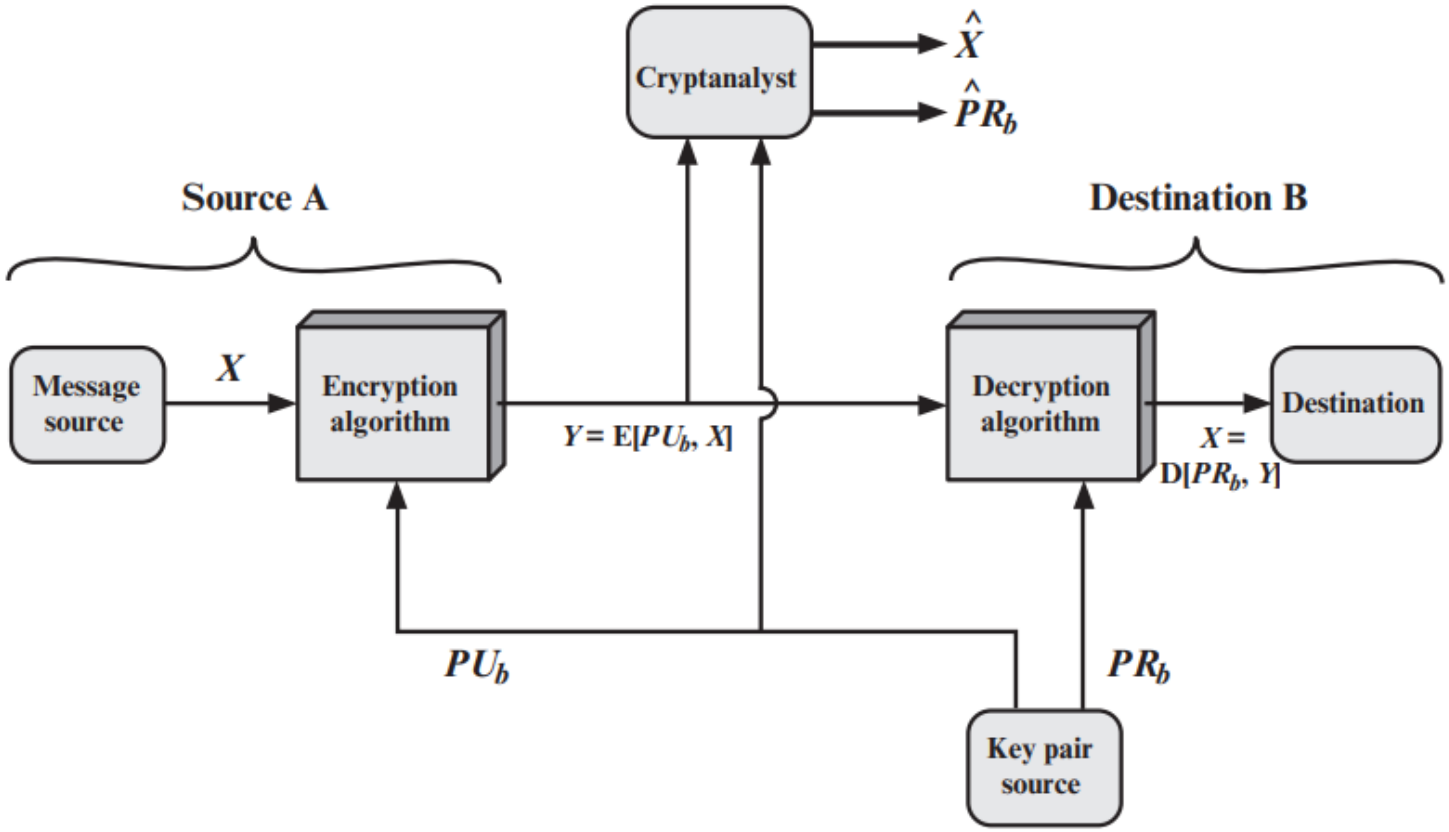
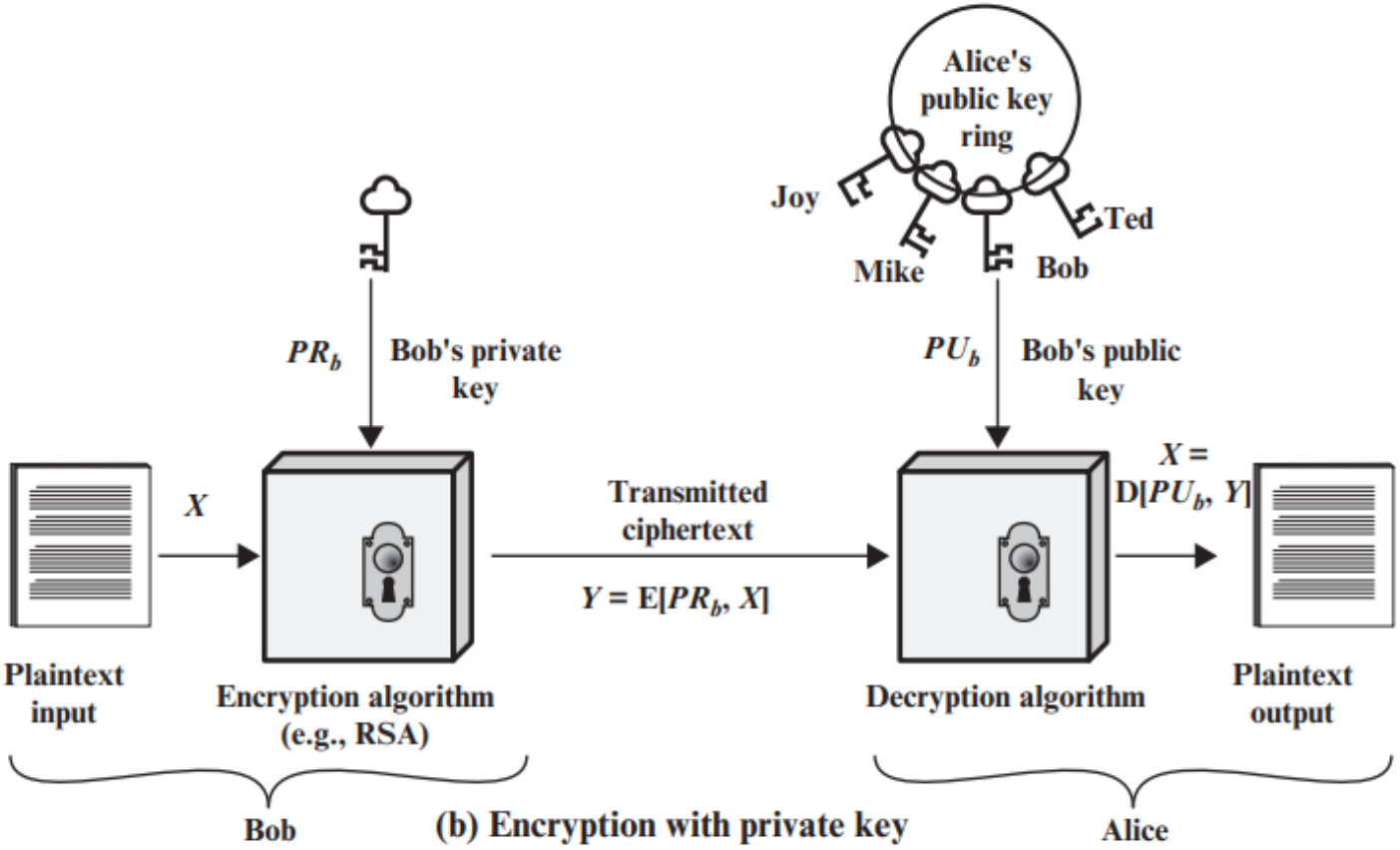


Figure 9.2 Public-Key Cryptosystem: Confidentiality

Public-Key Cryptosystem: Authentication



(b) Encryption with private key

Public-Key Cryptosystem: Authentication

- The figure show the use of public-key encryption to provide authentication:

$$Y = E(PRa, X)$$

$$X = D(PUa, Y)$$

- In this case, A prepares a message to B and encrypts it using A's private key before transmitting it. B can decrypt the message using A's public key. Because the message was encrypted using A's private key, only A could have prepared the message.
- Therefore, the entire encrypted message serves as a digital signature. In addition, it is impossible to alter the message without access to A's private key, so the message is authenticated both in terms of source and in terms of data integrity.

Public-Key Cryptosystem: Authentication

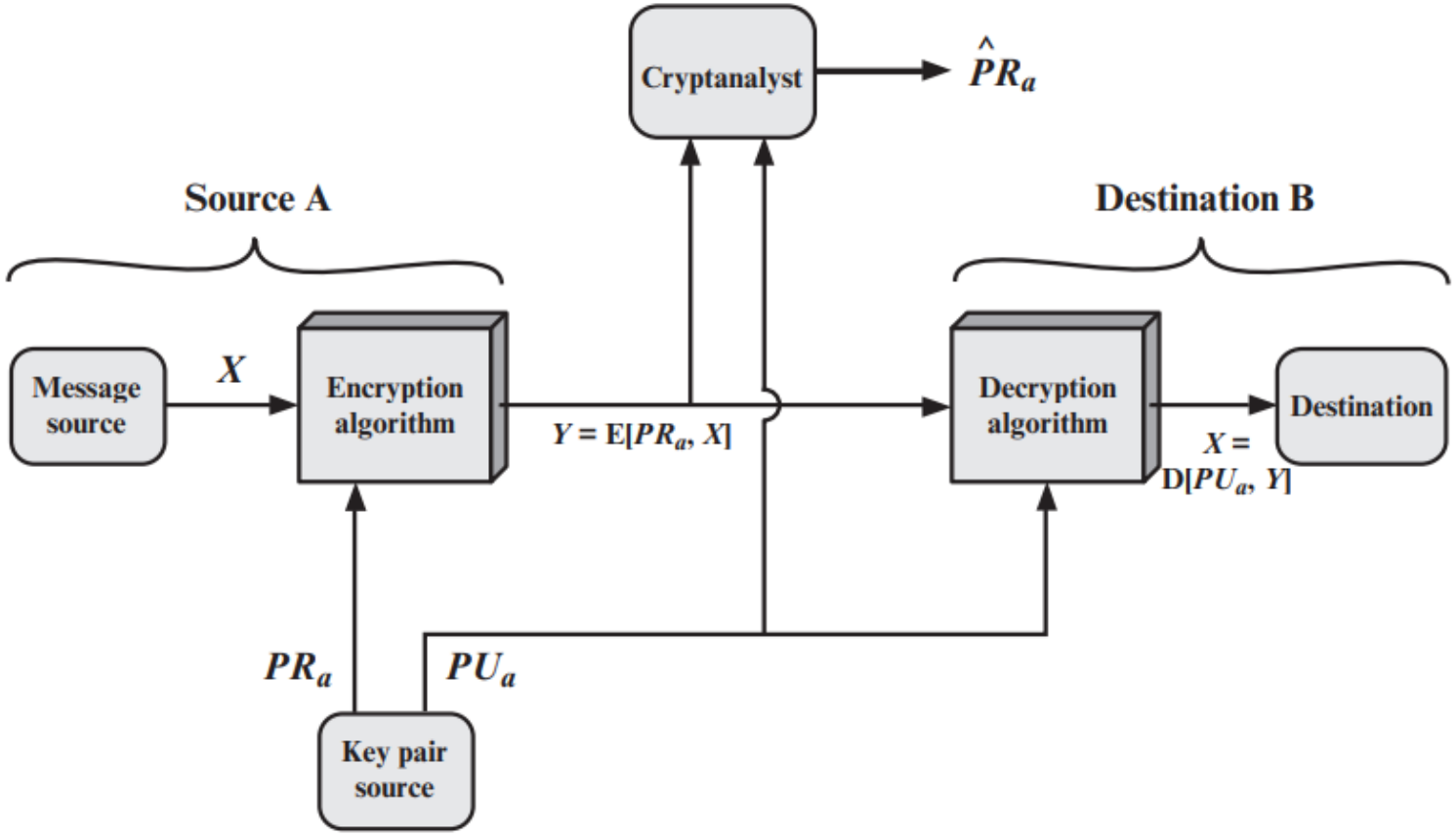


Figure 9.3 Public-Key Cryptosystem: Authentication

Public-Key Cryptosystem: Authentication

- It is important to emphasize that the encryption process depicted in Figures, does not provide confidentiality. That is, the message being sent is safe from alteration but not from eavesdropping.
- This is obvious in the case of a signature based on a portion of the message, because the rest of the message is transmitted in the clear. Even in the case of complete encryption, as shown in Figure, there is no protection of confidentiality because any observer can decrypt the message by using the sender's public key.

Public-Key Cryptosystem: Authentication and Secrecy

- It is, however, possible to provide both the authentication function and confidentiality by a double use of the public-key scheme :

  $Z = E(PUb, E(PRa,X))$

  $X = D(PUa, D(PRb,Z))$

- In this case, we begin as before by encrypting a message, using the sender's private key. This provides the digital signature. Next, we encrypt again, using the receiver's public key. The final ciphertext can be decrypted only by the intended receiver, who alone has the matching private key. Thus, confidentiality is provided. The disadvantage of this approach is that the public-key algorithm, which is complex, must be exercised four times rather than two in each communication.

Public-Key Cryptosystem: Authentication and Secrecy

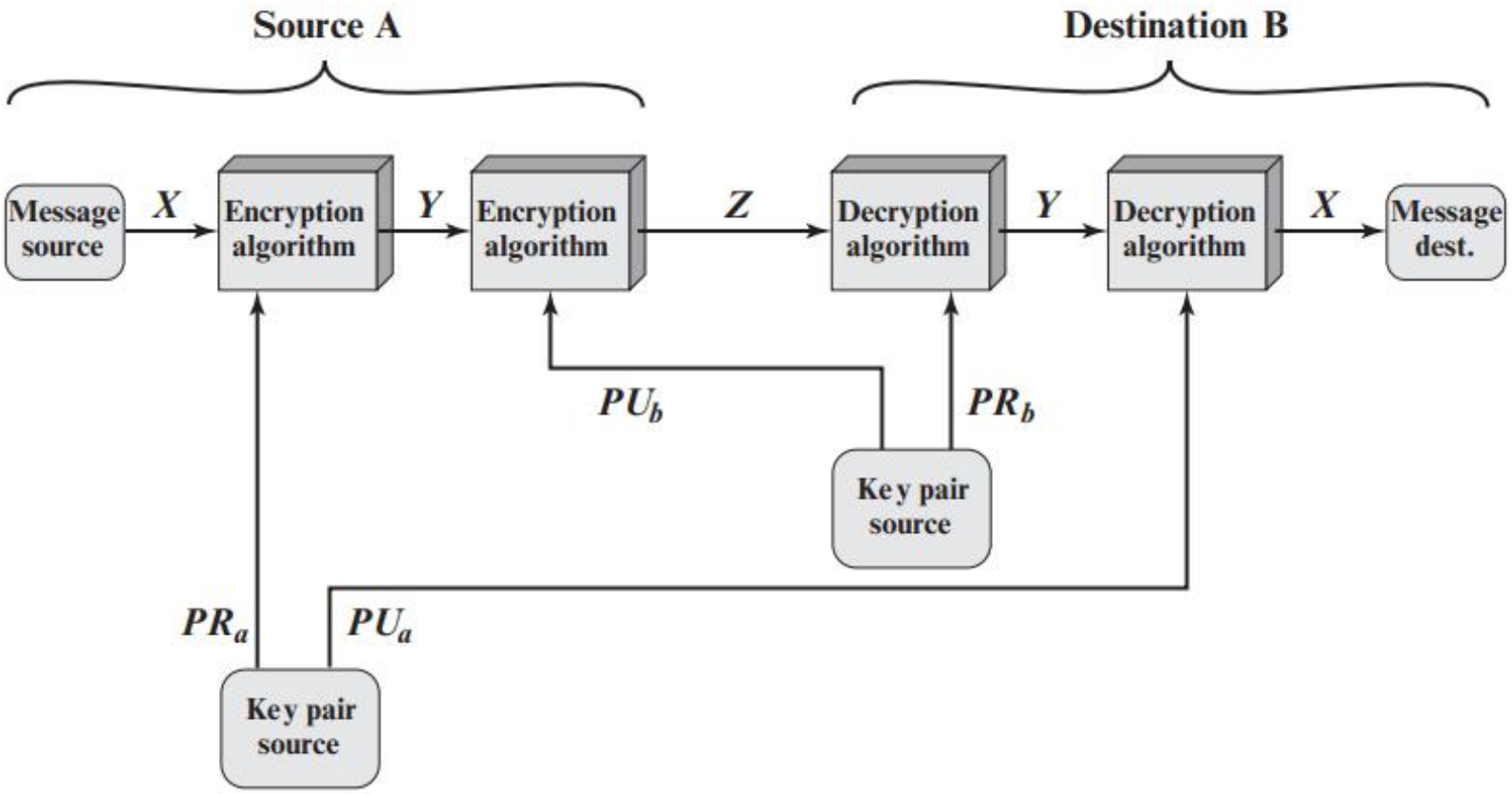


Figure 9.4 Public-Key Cryptosystem: Authentication and Secrecy

Applications of PKC.

- Depending on the application, the sender uses either the sender's private key or the receiver's public key, or both, to perform some type of cryptographic function. In broad terms, we can classify the use of public-key cryptosystems into three categories

- Encryption/decryption: The sender encrypts a message with the recipient's public key, and the recipient decrypts the message with the recipient's private key.
- Digital signature: The sender "signs" a message with its private key. Signing is achieved by a cryptographic algorithm applied to the message or to a small block of data that is a function of the message.
- Key exchange: Two sides cooperate to exchange a session key, which is a secret key for symmetric encryption generated for use for a particular transaction (or session) and valid for a short period of time. Several different approaches are possible, involving the private key(s) of one or both parties.

Requirement for PKC

1. It is computationally easy for a party B to generate a key pair (public key PUb, private key PRb).
2. It is computationally easy for a sender A, knowing the public key and the message to be encrypted, M, to generate the corresponding ciphertext:

$$C = E(PUb, M)$$

3. It is computationally easy for the receiver B to decrypt the resulting ciphertext using the private key to recover the original message:

$$M = D(PRb, C) = D[PRb, E(PUb, M)]$$

4. It is computationally infeasible for an adversary, knowing the public key, PUb, to determine the private key, PRb.

Requirement for PKC

5. It is computationally infeasible for an adversary, knowing the public key, PUb, and a ciphertext, C, to recover the original message, M.

- We can add a sixth requirement that, although useful, is not necessary for all public-key applications:

6. The two keys can be applied in either order:

$$M = D[PUb, E(PRb, M)] = D[PRb, E(PUb, M)]$$

- These are formidable requirements, as evidenced by the fact that only a few algorithms (RSA, elliptic curve cryptography, Diffie–Hellman, DSS) have received widespread acceptance in the several decades since the concept of public-key cryptography was proposed.

RSA Algorithm

- R. Rivest, A. Shamir, L. Adleman (1977)
  - James Ellis came up with the idea in 1970, and proved that it was theoretically possible. In 1973, Clifford Cocks a British mathematician invented a variant on RSA
- RSA uses Block cipher Techniques
- Algorithm:
  - Encryption: $C = M^e \bmod n$
  - Decryption: $M = C^d \bmod n$
- Both sender and the receiver know *n*

RSA Public Key Crypto System

Key generation:

1. Select 2 large prime numbers of about the same size, p and q

   Typically each p, q has between 512 and 2048 bits

2. Compute $n = pq$, and $\Phi(n) = (q-1)(p-1)$
3. Select e, $1<e<\Phi(n)$, s.t. $\gcd(e, \Phi(n)) = 1$

   Typically e=3 or e=65537

4. Compute d, $1< d<\Phi(n)$ s.t. $ed \equiv 1 \bmod \Phi(n)$

   Knowing $\Phi(n)$, d easy to compute.

Public key: (e, n)

Private key: (d, n)

RSA Public Key Crypto System

Encryption

Given a message M, $0 < M < n$ $\quad M \in Z_n - \{0\}$

use public key (e, n)

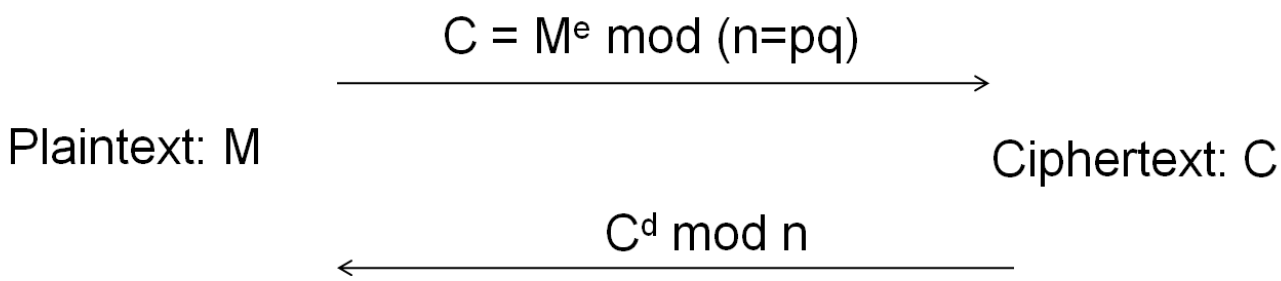
compute $\mathbf{C = M^e \bmod n}$ $\quad C \in Z_n - \{0\}$

Decryption

Given a ciphertext C, use private key (d)

Compute $\mathbf{M = C^d \bmod n}$

RSA Public Key Crypto System



From n, difficult to figure out p,q

From (n,e), difficult to figure d.

From (n,e) and C, difficult to figure out M s.t. $C = M^e$

RSA Example

1. Select primes: $p=17$ & $q=11$
2. Compute $n = pq = 17 \times 11 = 187$
3. Compute $\phi(n) = (p-1)(q-1) = 16 \times 10 = 160$
4. Select e : $\gcd(e,160)=1$; choose $e=7$
5. Determine d: $de = 1 \bmod 160$ and $d < 160$
   Value is $d=23$ since $23 \times 7 = 161 = 10 \times 160 + 1$
6. Publish public key $KU=\{7,187\}$
7. Keep secret private key $KR=\{23,187\}$

RSA Example cont

- sample RSA encryption/decryption is:
- given message $M = 88$ (nb. $88<187$)
- encryption:

  $C = 88^7 \bmod 187 = 11$
- decryption:

  $M = 11^{23} \bmod 187 = 88$

Efficient Encryption

- encryption uses exponentiation to power e
- hence if e small, this will be faster
  - often choose e=65537 (2^{16}-1)
  - also see choices of e=3 or e=17
- but if e too small (eg e=3), attacker can attack easily
  - using Chinese remainder theorem & 3 messages with different modulii
- if e fixed must ensure `gcd(e,ø(n))=1`
  - reject any p or q not relatively prime to e

Efficient Decryption

- decryption uses exponentiation to power d
  - this is likely large, insecure if not
- can use the Chinese Remainder Theorem (CRT) to compute mod p & q separately.
- then combine to get desired answer
  - approx 4 times faster than doing directly
- only owner of private key who knows values of p & q can use this technique

RSA Key Generation

- users of RSA must:
  - determine two primes at random - p, q
  - select either e or d and compute the other
- primes p, q must not be easily derived from modulus $n=p.q$
  - means must be sufficiently large
  - typically guess and use probabilistic test
- exponents e, d are inverses, so use Inverse algorithm to compute the other

RSA Security

- possible approaches to attacking RSA are:
  - brute force key search - infeasible given size of numbers
  - mathematical attacks - based on difficulty of computing ø(n), by factoring modulus n
  - timing attacks - on running of decryption
  - chosen ciphertext attacks - given properties of RSA

Brute force attack

- Brute force attack means trying all possible combinations on private keys.
- If any one case is matching, attack RSA algorithm.
- To overcome this problem, choose key size large. Whenever the key size is as large as possible, then to identify all possible combination is difficult.

Mathematical Attack: Factoring Problem

- mathematical approach takes 3 forms:
  - factor $n=p.q$, hence compute $\varnothing(n)$ and then d
  - determine $\varnothing(n)$ directly and compute d
  - find d directly

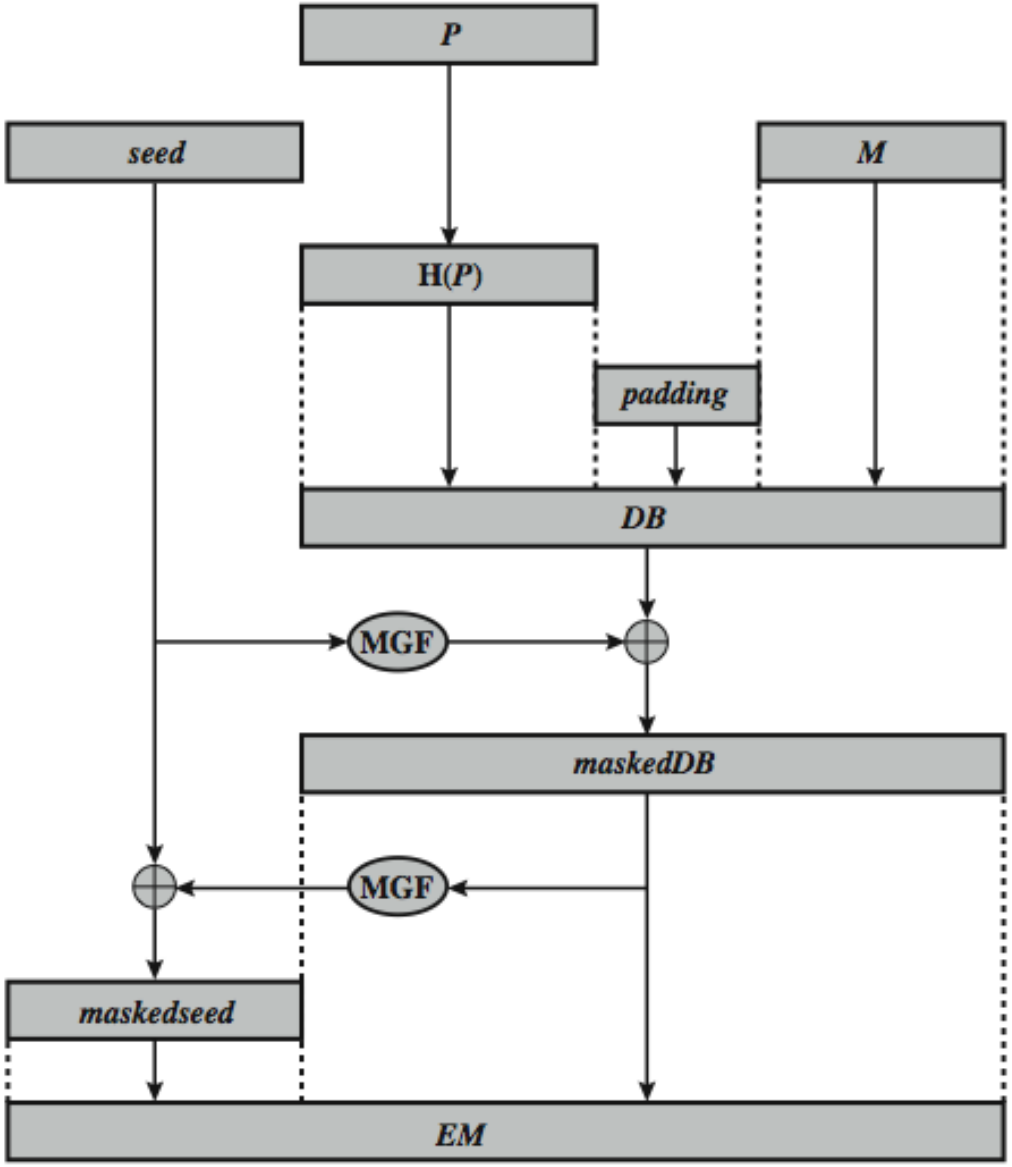
Timing Attacks

- developed by Paul Kocher in mid-1990's
- Keep track of how long a computer takes to decrypt a message!
- Attacker is identifies the running time of the decryption algorithm.
- countermeasures
  - use constant exponentiation time: add some fixed delay in every algorithm.
  - add random delays:
  - blind values used in calculations: multiply cipher text with some random number.

Chosen Ciphertext Attacks

- RSA is vulnerable to a Chosen Ciphertext Attack (CCA)
- attackers chooses ciphertexts & gets decrypted plaintext back
- choose ciphertext to exploit properties of RSA to provide info to help cryptanalysis
- can counter with random pad of plaintext
- or use Optimal Asymmetric Encryption Padding (OASP)

Optimal Asymmetric Encryption Padding (OASP)



A trapdoor one-way function

- A one-way function is one that maps a domain into a range such that every function value has a unique inverse, with the condition that the calculation of the function is easy, whereas the calculation of the inverse is infeasible:

  $Y = f(X)$ easy

  $X = f^{-1}(Y)$ infeasible

- The definition of a trap-door one-way function, which is easy to calculate in one direction and infeasible to calculate in the other direction unless certain additional information is known.

A trapdoor one-way function

- We can summarize as follows: A trapdoor one-way function is a family of invertible functions f_k, such that

  $Y = f_k(X)$ easy, if k and X are known

  $X = f_k^{-1}(Y)$ easy, if k and Y are known

  $X = f_k^{-1}(Y)$ infeasible, if Y is known but k is not known

- Thus, the development of a practical public-key scheme depends on discovery of a suitable trap-door one-way function.