PUBLIC KEY CRYPTOSYSTEMS

ASYMMETRIC KEY ENCRYPTION

- The concept of using different keys at the encryption and decryption ends.
- Depends on different mathematical principles than symmetric encryption.
- Usually done through a combination of hardware and software.
- Can be used for several different applications, other than just encryption.

MISCONCEPTIONS CONCERNING PUBLIC-KEY ENCRYPTION

- Public-key encryption is more secure from cryptanalysis than symmetric encryption
  - Not true – they depend on different principles, but can be equally secure
- Public-key encryption has made symmetric encryption obsolete
  - Not true – symmetric encryption is still used in several areas, quite successfully.

ASYMMETRIC ENCRYPTION TERMINOLOGY

- Asymmetric Keys
  - Two related keys – a public key and a private key, that are used to perform complemetatry operations, such as encryption and decryption or signature generation and signature verification
- Public Key Certificate
  - A digital document issued and digitally signed by the private key of the certification authority that binds the name of a subscriver to a public key. The certificate indicates that the subscriber identified in the certificate has sole control and access to the corresponding private key
- Public Key Algorithm
  - A cryptographic algorithm that uses the related keys, a public key and a private key. The two keys have the property that deriving the private key from the publick key is computationally infeasible.
- Public Key Infrastructure
  - A set of policies, processes, server platforms, software and workstations used for the purpose of administering certificates and public-private key pairs, including the ability to issue and revoke public key certificates.

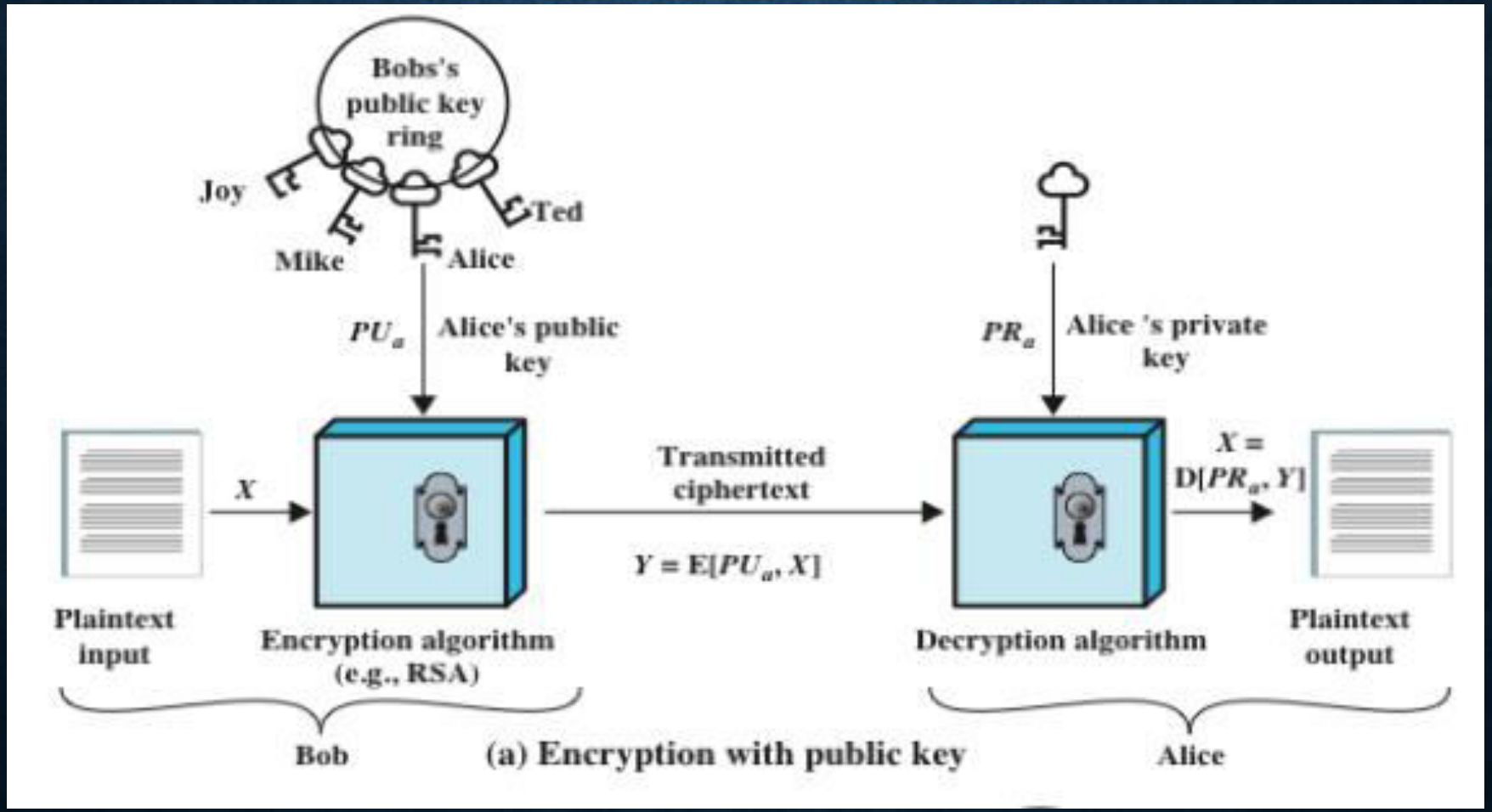
PRINCIPLES OF PUBLIC-KEY CRYPTOSYSTEMS

- Public-key cryptography evolved from an attempt to address the two basic limitations of symmetric encryption:
- Key Distribution - How to have secure communication without having to trust a KDC with your key
- Digital Signatures - How to verify that a message comes intact from the claimed sender
- Whit Diffie and Martin Hellman proposed a method that addressed both problems and was radically different from all previous approaches to cryptography

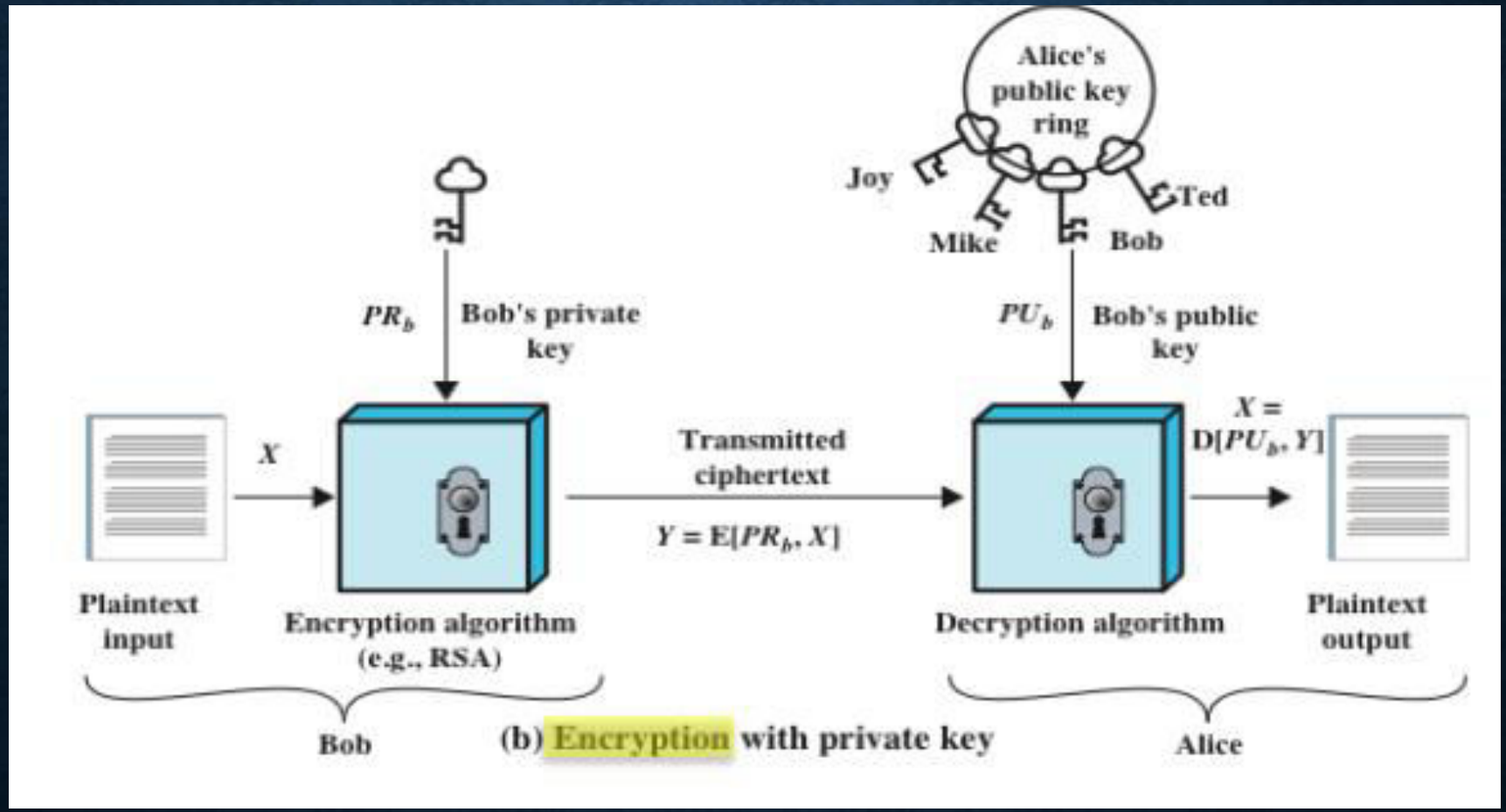
PUBLIC-KEY CRYPTOSYSTEMS - TERMINOLOGY

- Plaintext – the input data
- Encryption Algorithm – Performs various transformations on the plaintext
- Public Key – Used for encryption
- Private Key – Used for decryption
- Ciphertext – The output data
- Decryption Algorithm – Used for decryption

PUBLIC-KEY CRYPTOGRAPHY - ENCRYPTION



PUBLIC KEY CRYPTOGRAPHY - SIGNATURES

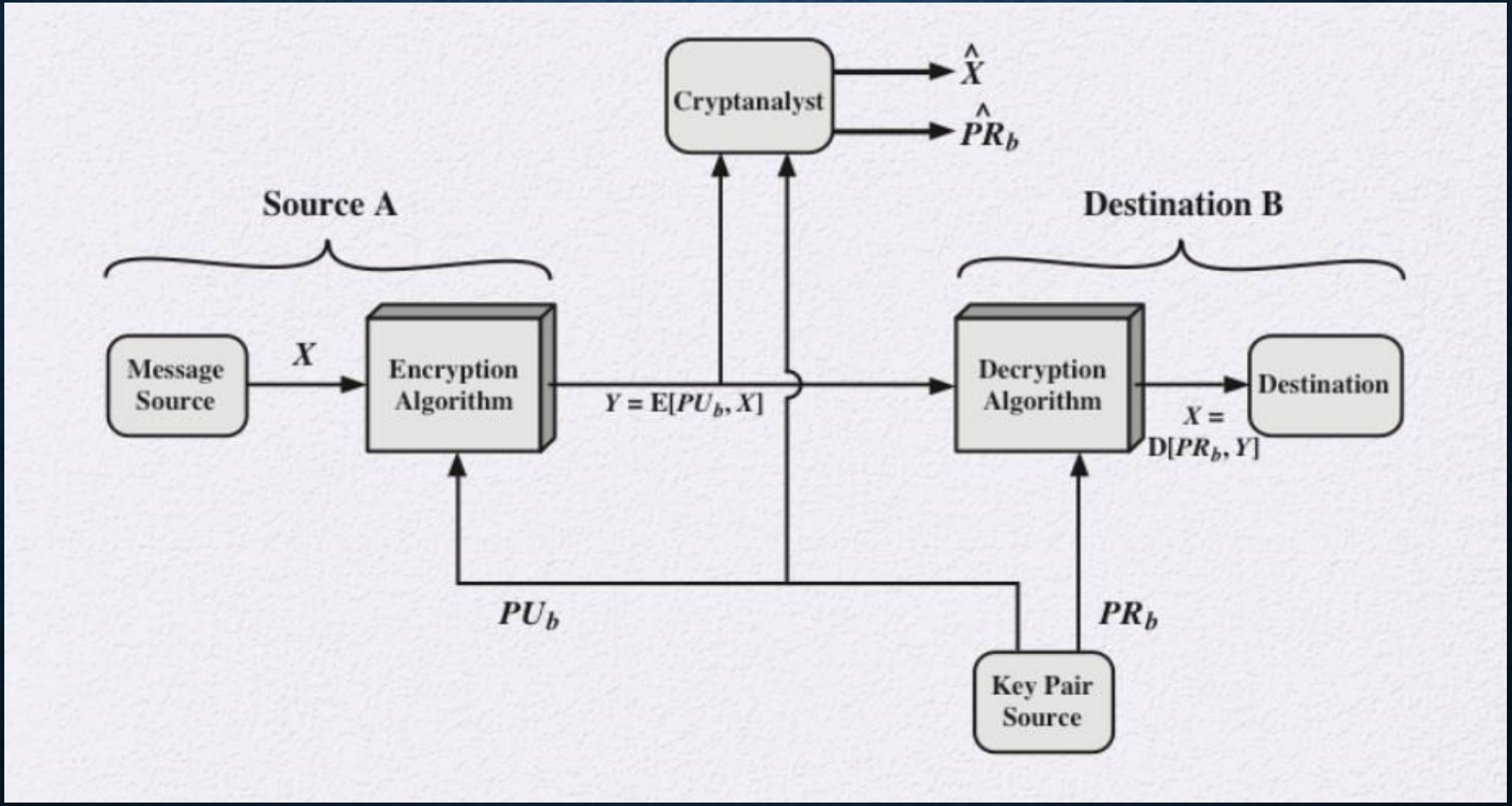


(b) Encryption with private key

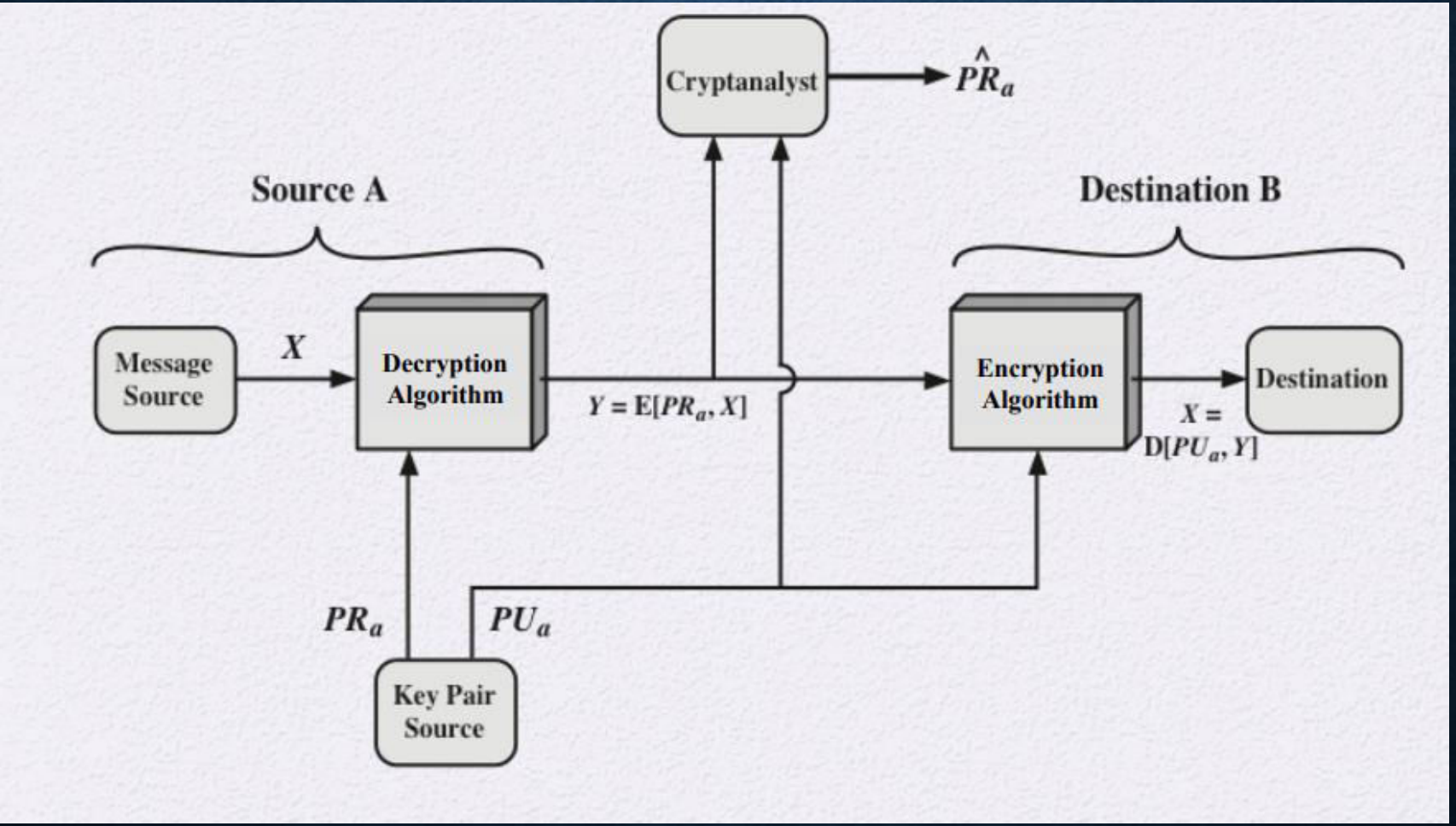
CONVENTIONAL AND PUBLIC-KEY ENCRYPTION

Conventional Encryption	Public-Key Encryption
Needed to Work: 1. The same algorithm with the same key is used for encryption and decryption. 2. The sender and receiver must share the algorithm and the key. *Needed for Security:* 1. The key must be kept secret. 2. It must be impossible or at least impractical to decipher a message if the key is kept secret. 3. Knowledge of the algorithm plus samples of ciphertext must be insufficient to determine the key.	*Needed to Work:* 1. One algorithm is used for encryption and a related algorithm for decryption with a pair of keys, one for encryption and one for decryption. 2. The sender and receiver must each have one of the matched pair of keys (not the same one). *Needed for Security:* 1. One of the two keys must be kept secret. 2. It must be impossible or at least impractical to decipher a message if one of the keys is kept secret. 3. Knowledge of the algorithm plus one of the keys plus samples of ciphertext must be insufficient to determine the other key.

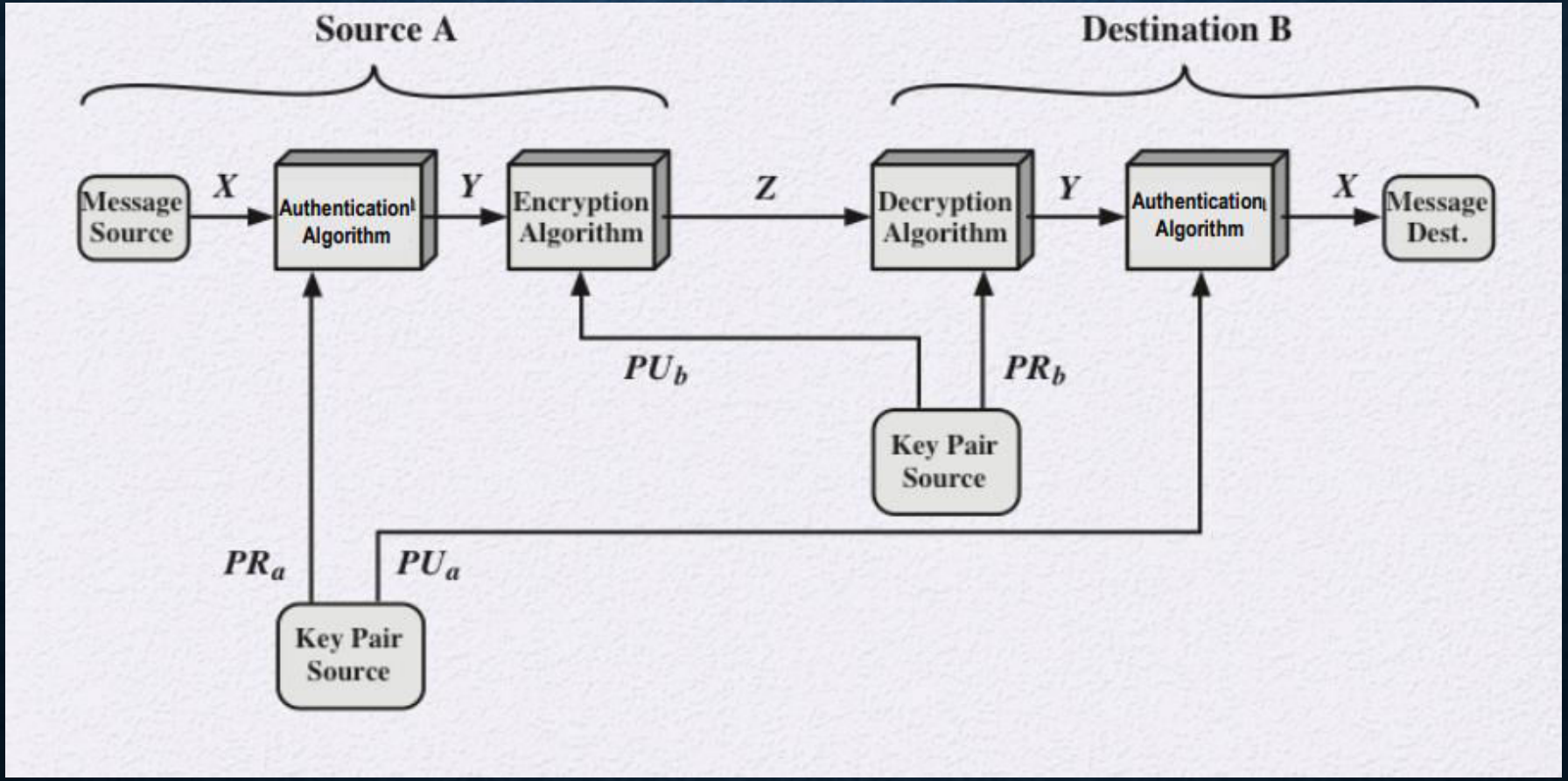
PUBLIC-KEY CRYPTOSYSTEM: SECRECY



PUBLIC-KEY CRYPTOSYSTEM: AUTHENTICATION



PUBLIC-KEY CRYPTOSYSTEM AUTHENTICATION AND SECURITY



APPLICATIONS FOR PUBLIC KEY CRYPTOSYSTEMS

- Public-key cryptosystems can be classified into three categories:
- Encryption/decryption: The sender encrypts a message with the recipient's public key
- Digital Signatures: The sender "signs" a message with its private key
- Key Exchange: Two sides cooperate to exchange a session key
- Some algorithms are suitable for all three applications, whereas others can be used only for one or two

APPLICATIONS FOR PUBLIC KEY CRYPTOSYSTEMS

Algortihm	Exncryption/Decryption	Digital Signature	Key Exchange
RSA	Yes	Yes	Yes
Elliptic Curves	Yes	Yes	YEs
Diffie-Hellman	No	No	Yes
DSS	No	Yes	No

PUBLIC KEY REQUIREMENTS

- Computationally easy
  - for party B to generate a pair (public-key PUb , private key PRb)
  - for sender A, knowing the public key and the message, to generate the corresponding ciphertext
  - for receiver B to decrypt the resulting ciphertext using the private key to recover the original message
- Computationally infeasible for an adversary
  - knowing the public key, to determine the private key
  - knowing the public key and a ciphertext, to recover the original message

PUBLIC KEY REQUIREMENTS

- Need a trap-door one-way function
  - $f : \{0,1\}^n \rightarrow \{0,1\}^n$ is a one-way function if
    - $Y = f(X)$ can easily be computed for X in $\{0,1\}^n$
    - $X = f^{-1}(Y)$ infeasible for Y in $\{0,1\}^n$
- A trap-door one-way function is a family of invertible functions fk , such that computing
  - $Y = f_k(X)$ is easy, if ~~k and~~ X ~~are~~ is known
  - $X = f_k^{-1}(Y)$ is easy, if k and Y are known
  - $X = f_k^{-1}(Y)$ infeasible, if Y is known but k not known
- A practical public-key scheme depends on a suitable trapdoor one-way function

RIVEST-SHAMIR-ADLEMAN (RSA) SCHEME

- Developed in 1977 by Ron Rivest, Adi Shamir & Len Adleman
- Most widely used general-purpose public-key encryption
- A cipher for which the plaintext and ciphertext are integers between 0 and n – 1 for some n
  - A typical size for n is 1024 bits, or 309 decimal digits

RSA ALGORITHM

- Plaintext is encrypted in blocks with whose value less than some number n
- Encryption and decryption are of the following form, for plaintext block M and ciphertext block C
- $C = M^e \bmod n$
- $M = C^d \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n$
- Both sender and receiver must know the value of n •
- The sender knows the value of e, and only the receiver knows the value of d
- This is a public-key encryption algorithm with a public key of PU={e,n} and a private key of PR={d,n}

ALGORITHM REQUIREMENTS

- It should be possible to find values of e, d, n such that

  $M^{ed} \bmod n = M$ for all $M < n$

- It should be relatively easy to calculate

  $M^{e} \bmod n$ and $C^{d} \bmod n$ for all values of $M < n$

- It should be infeasible to determine d given e and n

RSA ALGORITHM

Key Generation by Alice

Select p, q	p and q both prime, $p \neq q$
Calculate $n = p \times q$	
Calculate $\phi(n) = (p-1)(q-1)$	
Select integer e	$\gcd(\phi(n), e) = 1;\ 1 < e < \phi(n)$
Calculate d	$d \equiv e^{-1} \pmod{\phi(n)}$
Public key	$PU = \{e, n\}$
Private key	$PR = \{d, n\}$

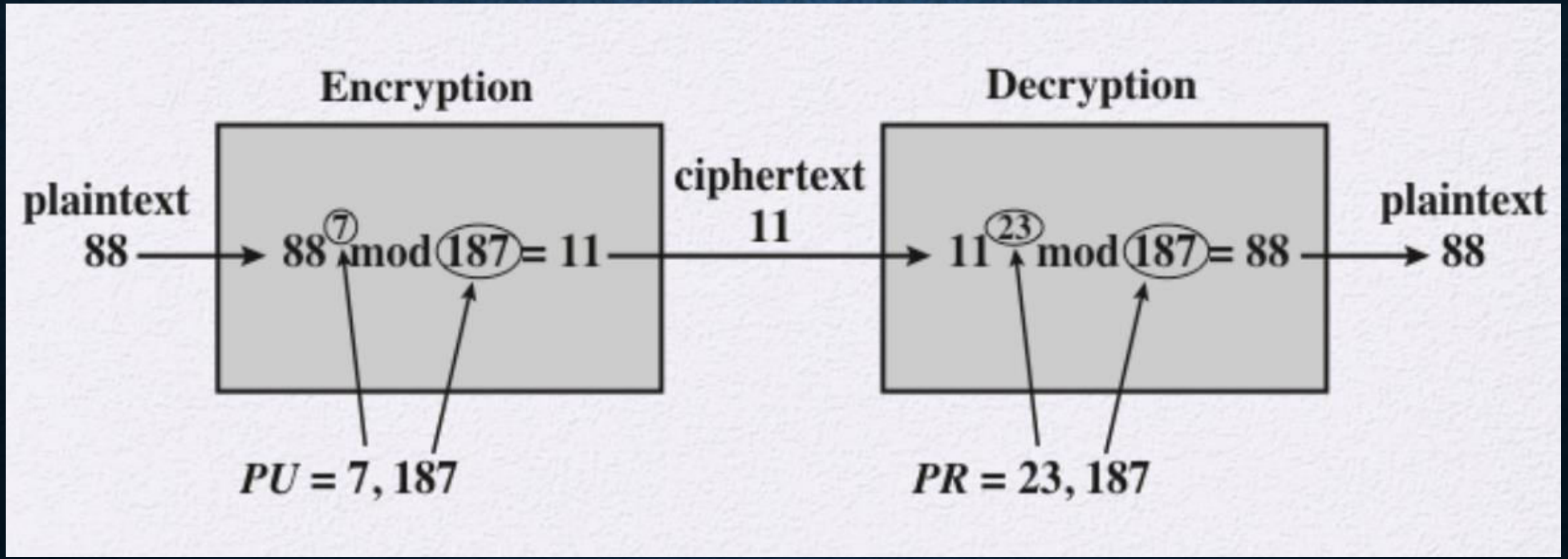
Encryption by Bob with Alice's Public Key

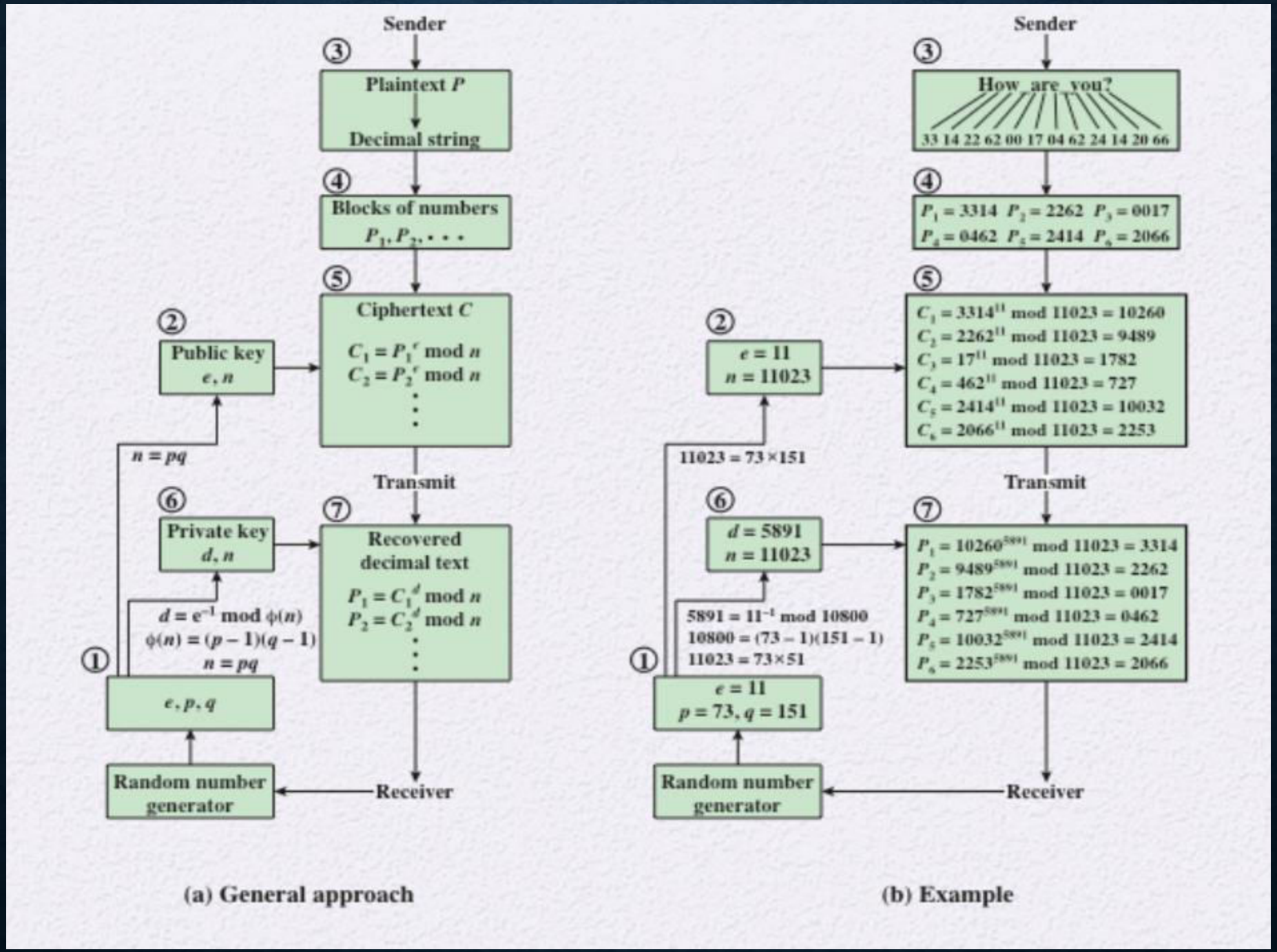
Plaintext:	$M < n$
Ciphertext:	$C = M^e \bmod n$

Decryption by Alice with Alice's Private Key

Ciphertext:	C
Plaintext:	$M = C^d \bmod n$

EXAMPLE OF RSA ALGORITHM




Sender
Plaintext P
Decimal string
Blocks of numbers
P1, P2, . . .
Ciphertext C
C1 = P1^e mod n
C2 = P2^e mod n
Public key
e, n
n = pq
Transmit
Private key
d, n
Recovered decimal text
P1 = C1^d mod n
P2 = C2^d mod n
d = e^-1 mod φ(n)
φ(n) = (p − 1)(q − 1)
n = pq
e, p, q
Random number generator
Receiver
(a) General approach
Sender
How are you?
33 14 22 62 00 17 04 62 24 14 20 66
P1 = 3314 P2 = 2262 P3 = 0017
P4 = 0462 P5 = 2414 P6 = 2066
C1 = 3314^11 mod 11023 = 10260
C2 = 2262^11 mod 11023 = 9489
C3 = 17^11 mod 11023 = 1782
C4 = 462^11 mod 11023 = 727
C5 = 2414^11 mod 11023 = 10032
C6 = 2066^11 mod 11023 = 2253
e = 11
n = 11023
11023 = 73 × 151
Transmit
d = 5891
n = 11023
P1 = 10260^5891 mod 11023 = 3314
P2 = 9489^5891 mod 11023 = 2262
P3 = 1782^5891 mod 11023 = 0017
P4 = 727^5891 mod 11023 = 0462
P5 = 10032^5891 mod 11023 = 2414
P6 = 2253^5891 mod 11023 = 2066
5891 = 11^-1 mod 10800
10800 = (73 − 1)(151 − 1)
11023 = 73 × 51
e = 11
p = 73, q = 151
Random number generator
Receiver
(b) Example

EXPONENTIATION IN MODULAR ARITHMETIC

- Both encryption and decryption in RSA involve raising an integer to an integer power, mod n
- Can make use of a property of modular arithmetic:

$$[(a \bmod n) \times (b \bmod n)] \bmod n = (a \times b) \bmod n$$

- With RSA you are dealing with potentially large exponents so efficiency of exponentiation is a consideration

EFFICIENT OPERATION USING THE PUBLIC KEY

- To speed up the operation of the RSA algorithm using the public key, a specific choice of e is usually made
- The most common choice is 65537 ($2^{16} + 1$)
- Two other popular choices are e=3 and e=17
- Each of these choices has only two 1 bits, so the number of multiplications required to perform exponentiation is minimized

EFFICIENT OPERATION USING THE PRIVATE KEY

- Decryption uses exponentiation to power d
- A small value of d is vulnerable to a brute-force attack and to other forms of cryptanalysis
- Can use the Chinese Remainder Theorem (CRT) to speed up computation
  - The quantities d mod (p – 1) and d mod (q – 1) can be precalculated
  - End result is that the calculation is approximately four times as fast as evaluating

    $M = C^d \bmod n$ directly

KEY GENERATION

- Before the application of the public-key cryptosystem each participant must generate a pair of keys:
  - Determine two prime numbers p and q
  - Select either e or d and calculate the other
- Because the value of n = pq will be known to any potential adversary, primes must be chosen from a sufficiently large set
- The method used for finding large primes must be reasonably efficient

PROCEDURE FOR PICKING A PRIME NUMBER

- Pick an odd integer n at random
- Pick an integer a < n at random
- Perform the probabilistic primality test with a as a parameter. If n fails the test, reject the value n and go to step 1
- If n has passed a sufficient number of tests, accept n; otherwise, go to step 2

THE SECURITY OF RSA

Five possible approaches to attacking RSA

- Brute force
  - Try all possible private keys
- Mathematical attacks
  - Several approaches, all equivalent in effort to factoring the product of two primes
- Timing attacks
  - Depend on the run time of the decryption algorithm
- Hardware fault-based attack
  - Inducing hardware faults in the processor that is generating digital signatures
- Chosen ciphertext attacks
  - Exploits properties of the RSA algorithm

FACTORING PROBLEM

- We can identify three approaches to attacking RSA mathematically:
- Factor n into its two prime factors. This enables calculation of phi(n) = (p – 1) x (q – 1), which in turn enables determination of $d = e^{-1} \pmod{phi(n)}$
- Determine phi(n) directly without first determining p and q. Again this enables determination of $d = e^{-1} \pmod{phi(n)}$
- Determine d directly without first determining phi(n)

Type of Attack	Known to Cryptanalyst
Ciphertext Only	• Encryption algorithm • Ciphertext
Known Plaintext	• Encryption algorithm • Ciphertext • One or more plaintext-ciphertext pairs formed with the secret key
Chosen Plaintext	• Encryption algorithm • Ciphertext • Plaintext message chosen by cryptanalyst, together with its corresponding ciphertext generated with the secret key
Chosen Ciphertext	• Encryption algorithm • Ciphertext • Ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key