MODULE 3

INFORMATION SECURITY [3 0 0 3]

ICT 3172:

The conceptual differences between the two systems are based on how these systems keep a secret.

In symmetric-key cryptography, the secret must be shared between two persons.

In asymmetric-key cryptography, the secret is personal (unshared); each person creates and keeps his or her own secret.

In a community of n people, $n(n - 1)/2$ shared secrets are needed for symmetric-key cryptography;

Only n personal secrets are needed in asymmetric-key cryptography.

Symmetric-key cryptography is based on sharing secrecy;

asymmetric-key cryptography is based on personal secrecy.

In symmetric-key cryptography, the plaintext and ciphertext are thought of as a combination of symbols.

Encryption and decryption permute these symbols or substitute a symbol for another.

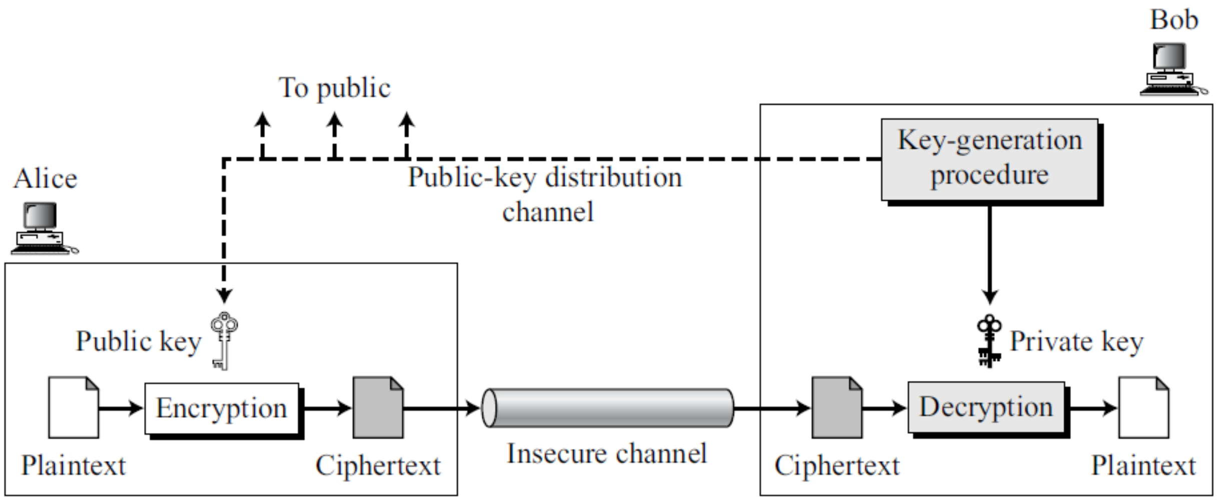
In asymmetric-key cryptography, the plaintext and ciphertext are numbers;

encryption and decryption are mathematical functions that are applied to numbers to create other numbers.

General Idea

General idea of asymmetric-key cryptography as used for encipherment.

Figure 10.2 *General idea of asymmetric-key cryptosystem*



There are distinctive keys in asymmetric-key cryptography: a private key and a public key.

Figure 10.2 shows several important facts.

First, it emphasizes the asymmetric nature of the cryptosystem. Bob needs to create two keys: one private and one public. Bob is responsible for distributing the public key to the community. This can be done through a public-key distribution channel.

Second, asymmetric-key cryptography means that Bob and Alice cannot use the same set of keys for two-way communication.
Each entity in the community should create its own private and public keys.

Third, asymmetric-key cryptography means that Bob needs only one private key to receive all correspondence from anyone in the community, but Alice needs n public keys to communicate with n entities in the community, one public key for each entity.

Plaintext/Ciphertext

Unlike in symmetric-key cryptography, plaintext and ciphertext are treated as integers in asymmetric-key cryptography.

The message must be encoded as an integer (or a set of integers) before encryption;

the integer (or the set of integers) must be decoded into the message after decryption.

Encryption/Decryption

Encryption and decryption in asymmetric-key cryptography are mathematical functions applied over the numbers representing the plaintext and ciphertext.

ciphertext C = f (Kpublic, P);

plaintext P = g(Kprivate, C).

function f is used only for encryption;

function g is used only for decryption.

function f needs to be a trapdoor one-way function to allow Bob to decrypt the message but to prevent Eve from doing so.

Need for Both

The advent of asymmetric key (public-key) cryptography does not eliminate the need for symmetric-key (secretkey) cryptography.

Asymmetric-key cryptography, which uses mathematical functions for encryption and decryption, is much slower than symmetric-key cryptography.

For encipherment of large messages, symmetric-key cryptography is still needed.

Speed of symmetric-key cryptography does not eliminate the need for asymmetric-key cryptography.

Asymmetric-key cryptography is still needed for authentication, digital signatures, and secret-key exchanges.

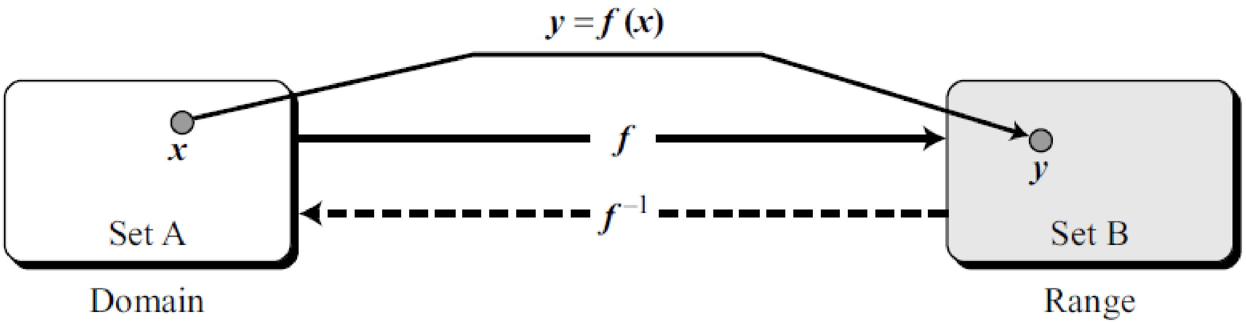
Trapdoor One-Way Function

Main idea behind asymmetric-key cryptography is the concept of the trapdoor oneway function.

Functions

A function is a rule that associates (maps) one element in set A, called the domain, to one element in set B.

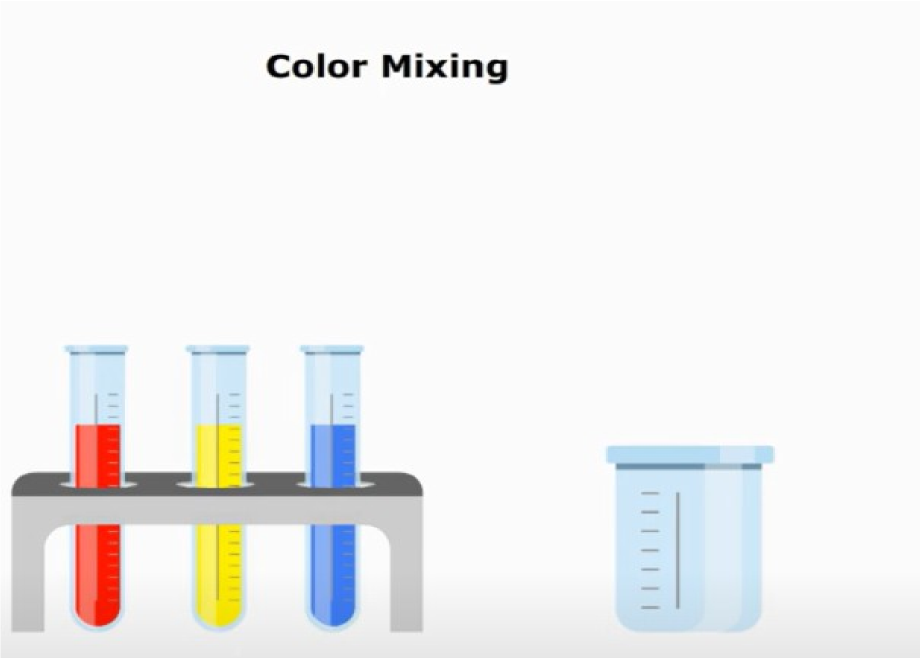
Figure 10.3 *A function as rule mapping a domain to a range*



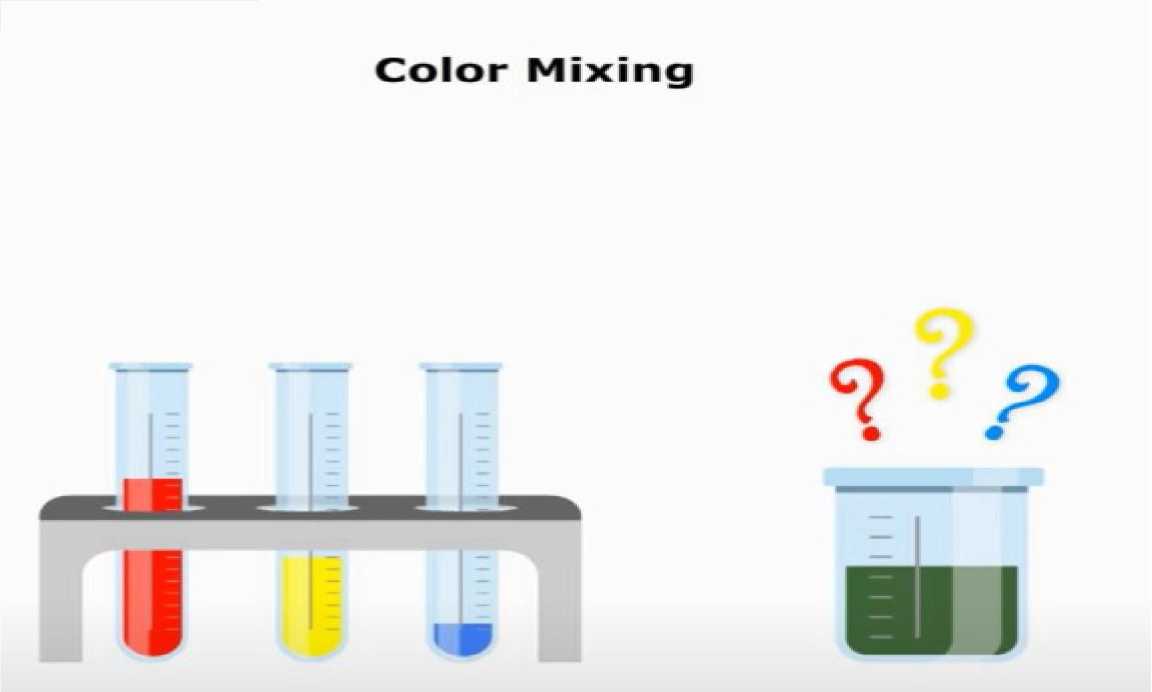
An invertible function is a function that associates each element in the range with exactly one element in the domain.

Smoothie Making

Smoothie Making

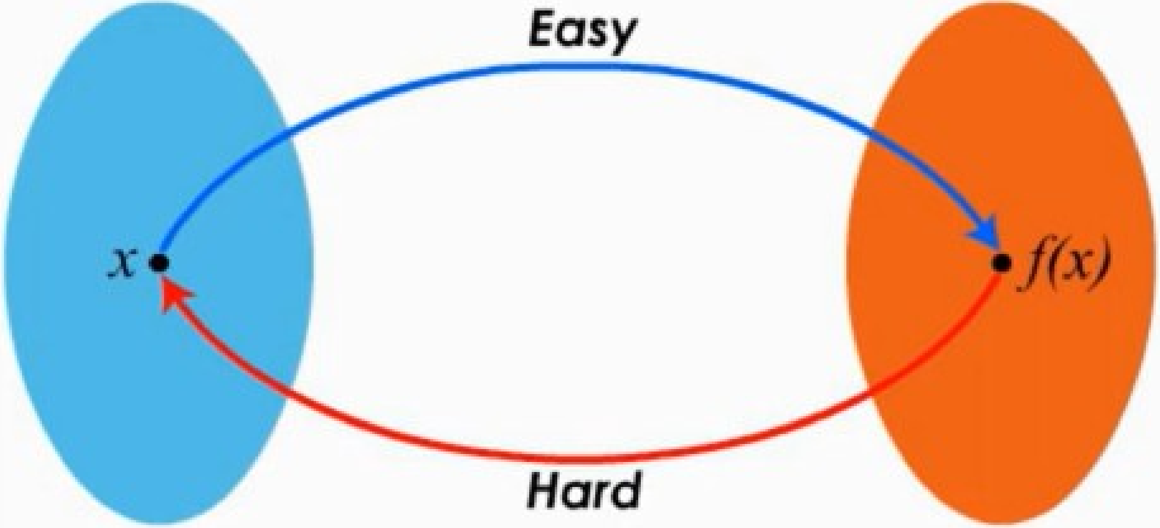


Color Mixing



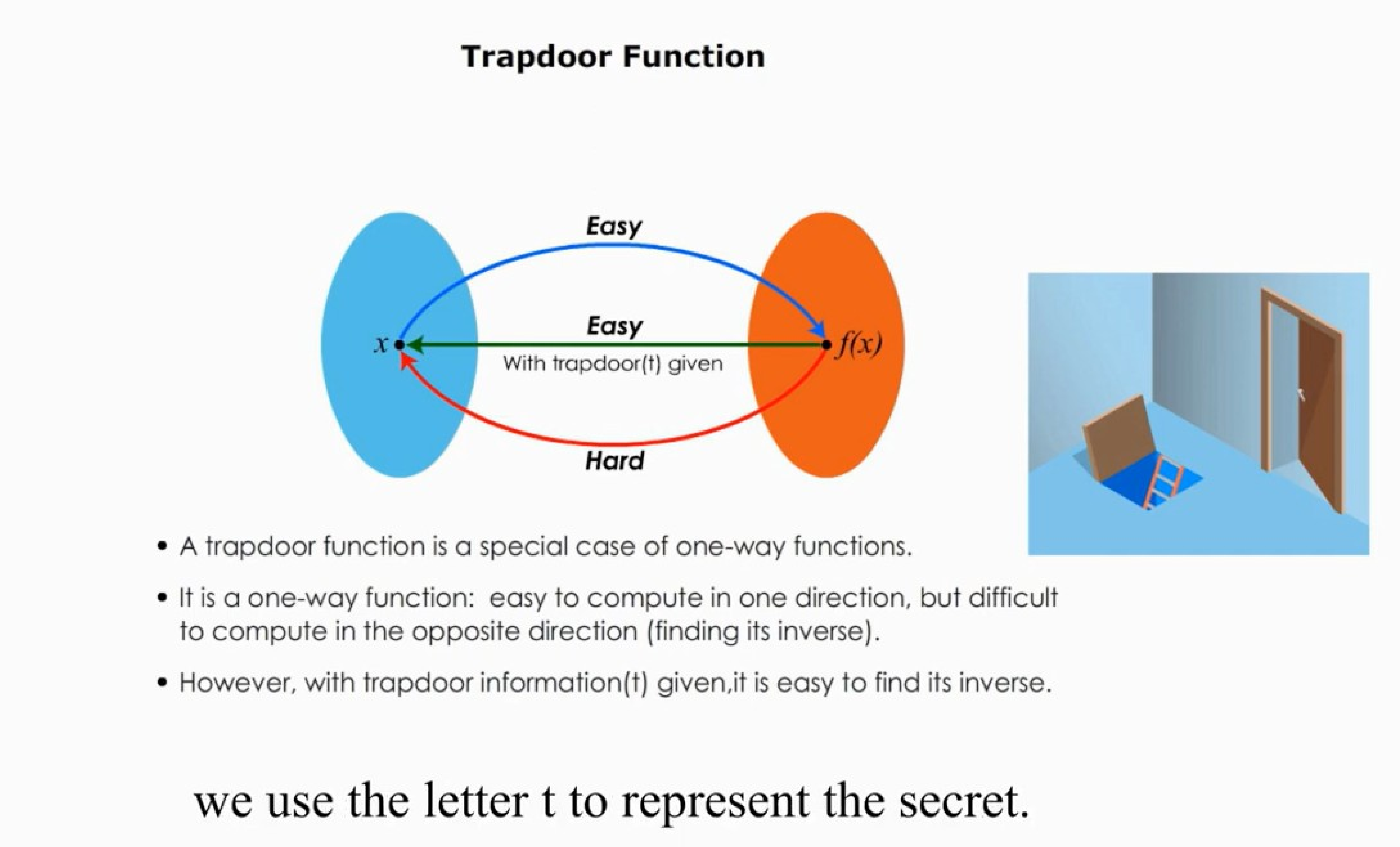
Color Mixing

One-way Function



Easy: Given x, it is easy to compute f(x).

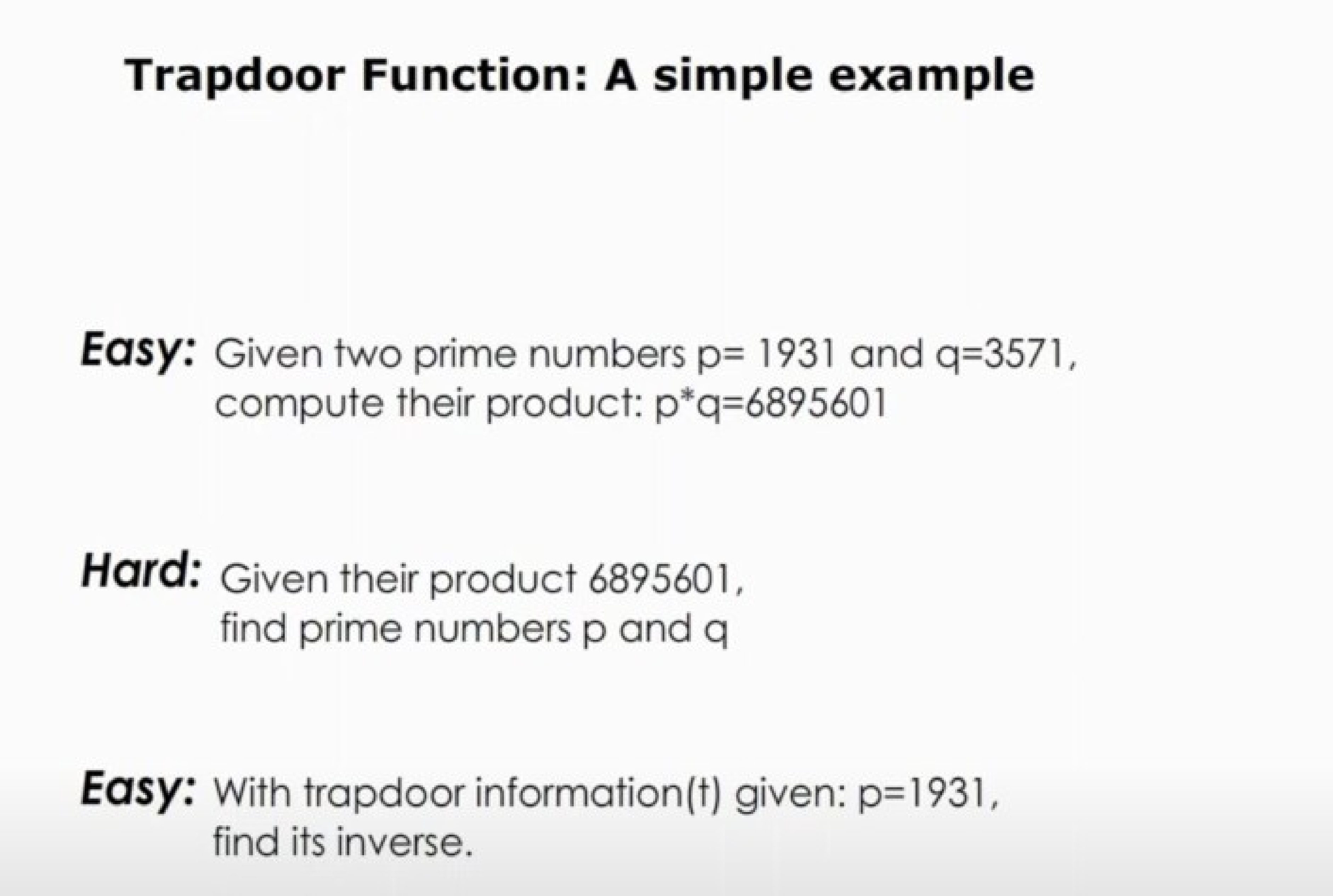
Hard: Practically impossible to find its inverse.



Trapdoor Function

- A trapdoor function is a special case of one-way functions.
- It is a one-way function: easy to compute in one direction, but difficult to compute in the opposite direction (finding its inverse).
- However, with trapdoor information(t) given,it is easy to find its inverse.

we use the letter t to represent the secret.



Trapdoor Function: A simple example

Easy: Given two prime numbers p= 1931 and q=3571, compute their product: p*q=6895601

Hard: Given their product 6895601, find prime numbers p and q

Easy: With trapdoor information(t) given: p=1931, find its inverse.

One-Way Function

A one-way function (OWF) is a function that satisfies the following two properties:

1. f is easy to compute. In other words, given x, y = f (x) can be easily computed.

2. f^{-1} is difficult to compute. In other words, given y, it is computationally infeasible to calculate $x = f^{-1}(y)$.

One-Way Function

Example 10.1

When n is large, n = p × q is a one-way function.

function x is a tuple (p, q) of two primes and y is n.

Given p and q, it is always easy to calculate n; given n, it is very difficult to compute p and q.

This is the factorization problem. There is not a polynomial time solution to the f^{-1} function in this case.

One-Way Function

Example 10.2

When n is large, the function $y = x^k \bmod n$ is a trapdoor one-way function.

Given x, k, and n, it is easy to calculate y using the fast exponential algorithm.

Given y, k, and n, it is very difficult to calculate x. This is the discrete logarithm problem.

There is not a polynomial time solution to the f^{-1} function in this case.

However, if we know the trapdoor, k′ such that $k \times k' = 1 \bmod \varphi(n)$, we can use $x = y^{k'} \bmod n$ to find x. This is the famous RSA.

10.2 RSA CRYPTOSYSTEM

The most common public-key algorithm is the RSA cryptosystem, named for its inventors (Rivest, Shamir, and Adleman).

RSA uses two exponents, e and d, where e is public and d is private.

Suppose P is the plaintext and C is the ciphertext.

Alice uses $C = P^e \bmod n$ to create ciphertext C from plaintext P;

Bob uses $P = C^d \bmod n$ to retrieve the plaintext sent by Alice.

n is very large number, is created during the key generation process.

10.2 RSA CRYPTOSYSTEM

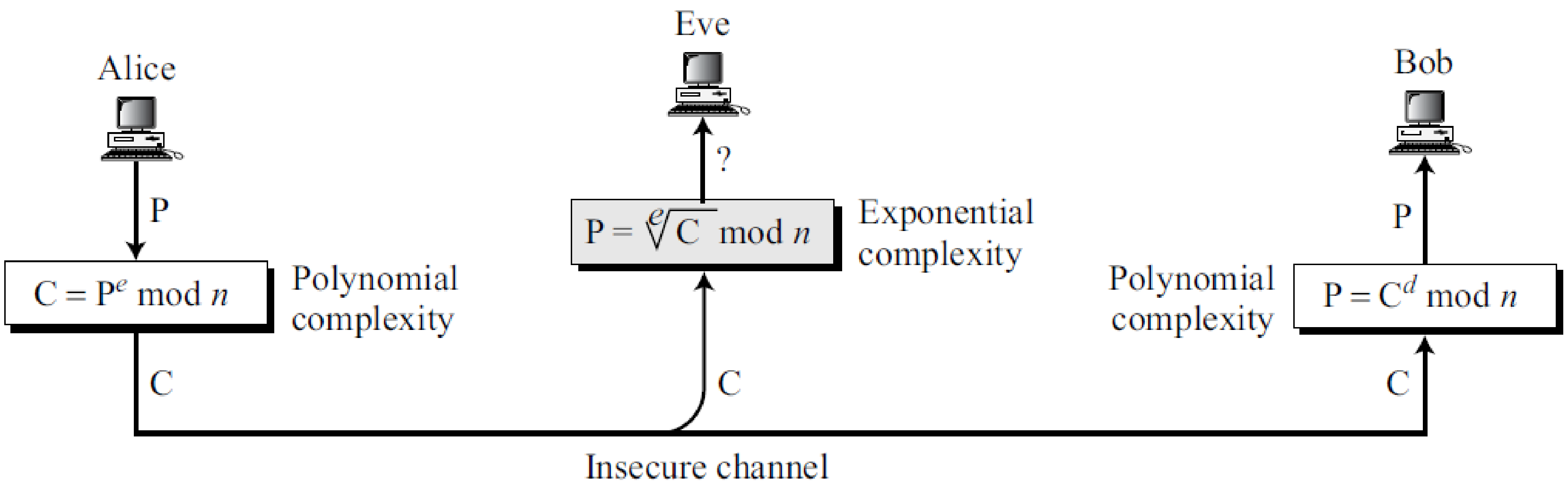
Encryption and decryption use modular exponentiation.

Modular exponentiation is feasible in polynomial time using the fast exponentiation algorithm.

However, modular logarithm is as hard as factoring the modulus, for which there is no polynomial algorithm yet.

10.2 RSA CRYPTOSYSTEM

Figure 10.5 *Complexity of operations in RSA*



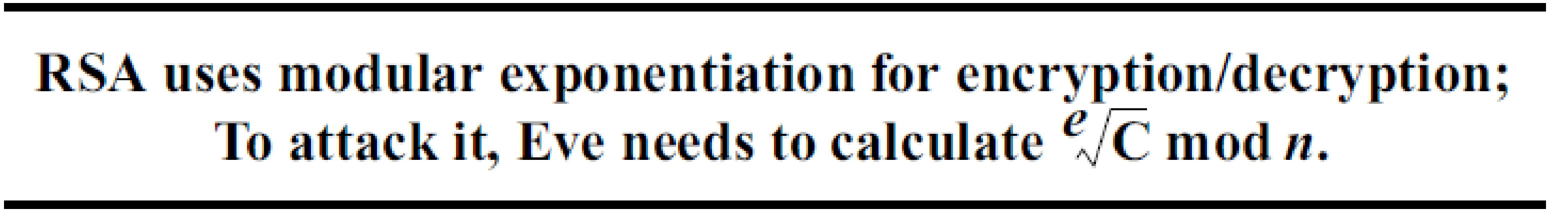
This means that Alice can encrypt in polynomial time (e is public), Bob also can decrypt in polynomial time (because he knows d), but Eve cannot decrypt because she would have to calculate the eth root of C using modular arithmetic.

10.2 RSA CRYPTOSYSTEM

In other words, Alice uses a one-way function (modular exponentiation) with a trapdoor known only to Bob.

Eve, who does not know the trapdoor, cannot decrypt the message.

If some day, a polynomial algorithm for eth root modulo n calculation is found, modular exponentiation is not a one-way function any more.

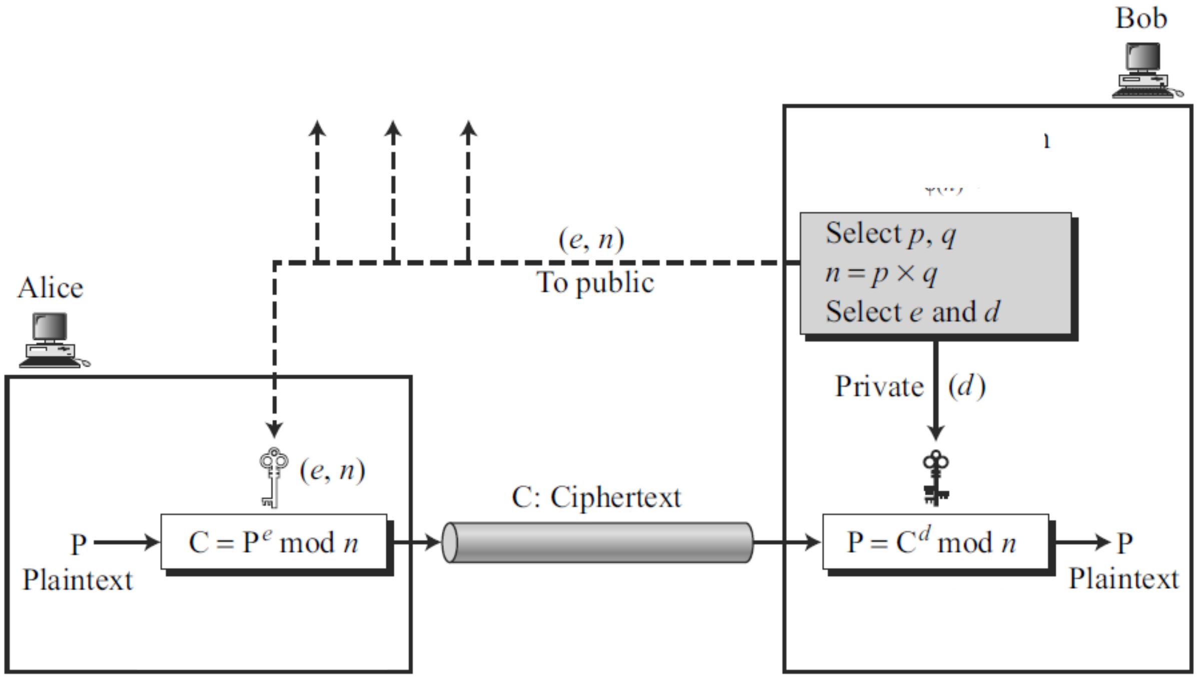


RSA uses modular exponentiation for encryption/decryption; To attack it, Eve needs to calculate $\sqrt[e]{C} \bmod n$.

Procedure

general idea behind the procedure used in RSA.

Figure 10.6 *Encryption, decryption, and key generation in RSA*



Procedure

Key Generation

Bob uses the steps shown in Algorithm 10.2 to create his public and private key.

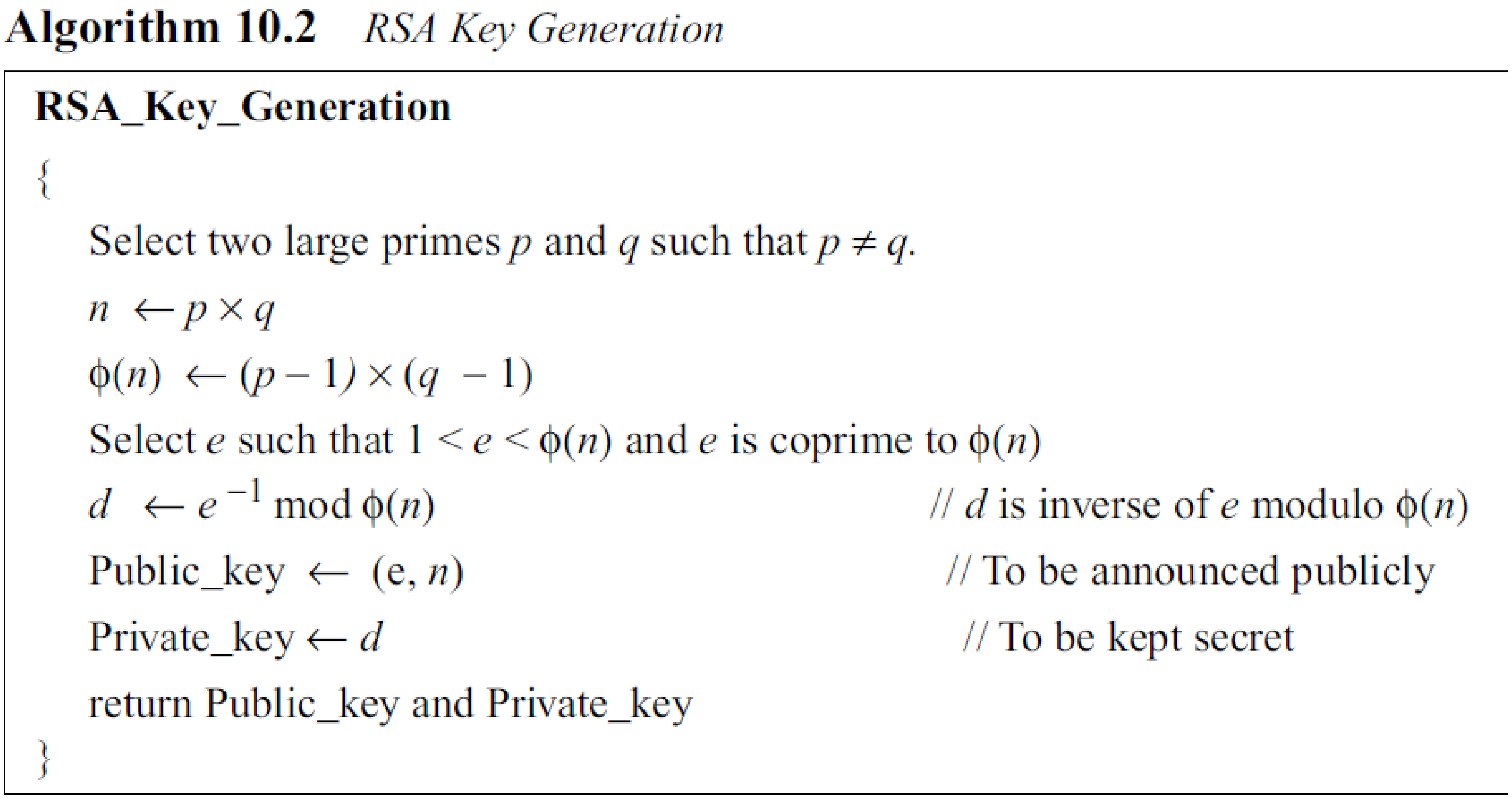
After key generation, Bob announces the tuple (e, n) as his public key; Bob keeps the integer d as his private key.

Bob can discard p, q, and φ(n); they will not be needed unless Bob needs to change his private key without changing the modulus (not recommended).

Recommended size for each prime, p or q, is 512 bits (almost 154 decimal digits). This makes the size of n, the modulus, 1024 bits (309 digits).

Procedure

Key Generation



Algorithm 10.2 *RSA Key Generation*

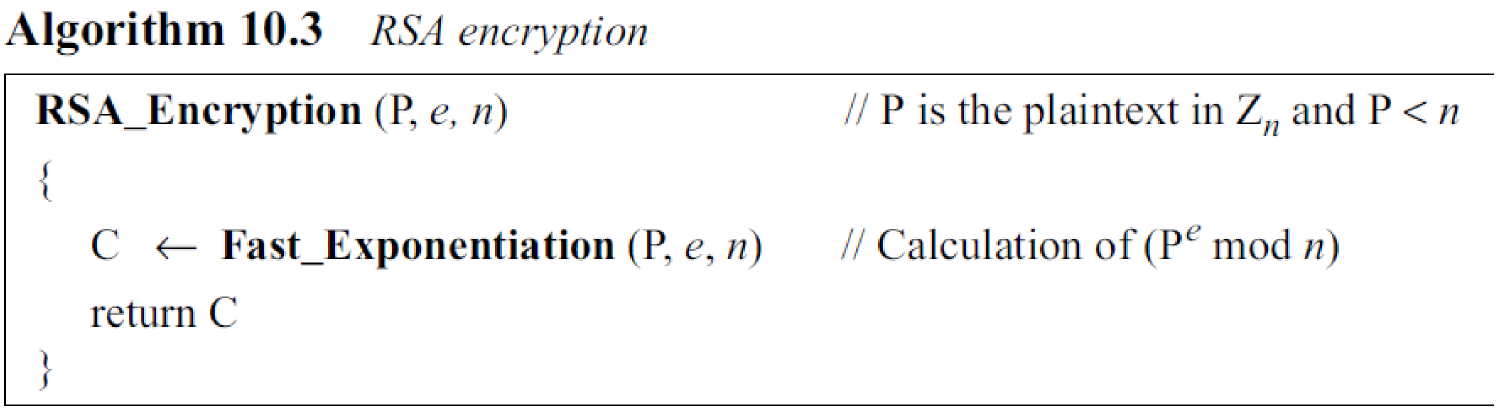
```
RSA_Key_Generation
{
    Select two large primes $p$ and $q$ such that $p \neq q$.
    $n \leftarrow p \times q$
    $\phi(n) \leftarrow (p - 1) \times (q - 1)$
    Select $e$ such that $1 < e < \phi(n)$ and $e$ is coprime to $\phi(n)$
    $d \leftarrow e^{-1} \bmod \phi(n)$                    // $d$ is inverse of $e$ modulo $\phi(n)$
    Public_key $\leftarrow$ (e, $n$)                       // To be announced publicly
    Private_key $\leftarrow d$                             // To be kept secret
    return Public_key and Private_key
}
```

Procedure

Encryption

Anyone can send a message to Bob using his public key.

The size of the plaintext must be less than n, which means that if the size of the plaintext is larger than n, it should be divided into blocks.



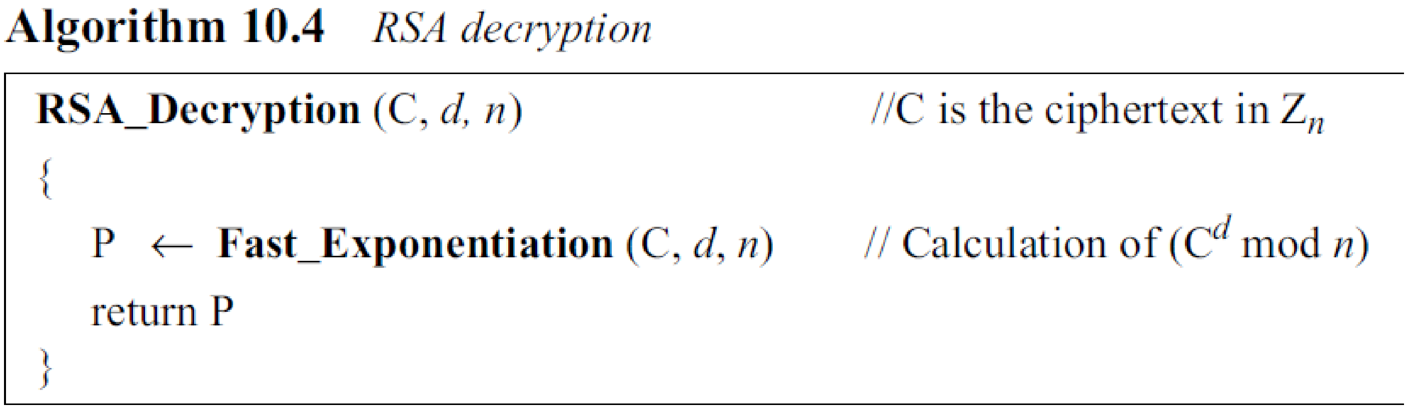
Algorithm 10.3 *RSA encryption*

```
RSA_Encryption (P, $e$, $n$)                        // P is the plaintext in $Z_n$ and $P < n$
{
    C ← Fast_Exponentiation (P, $e$, $n$)           // Calculation of ($P^e$ mod $n$)
    return C
}
```

Procedure

Decryption

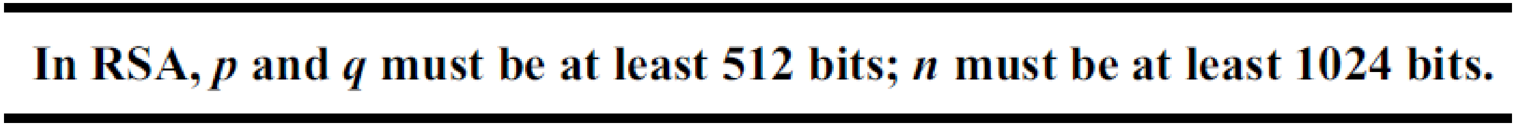
The size of the ciphertext is less than n.



Algorithm 10.4 *RSA decryption*

```
RSA_Decryption (C, $d$, $n$)                            //C is the ciphertext in $Z_n$
{
    P ← Fast_Exponentiation (C, $d$, $n$)         // Calculation of ($C^d$ mod $n$)
    return P
}
```

Procedure



In RSA, p and q must be at least 512 bits; n must be at least 1024 bits.

Some Trivial Examples

Example 10.5

Bob chooses 7 and 11 as p and q and calculates n = 7×11 = 77.

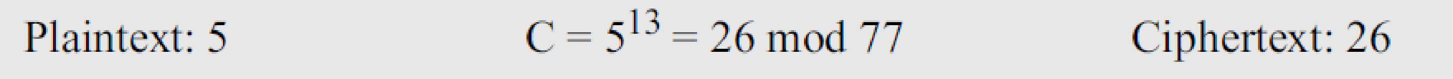
The value of $\varphi(n) = (7 - 1)(11 - 1)$ or 60.

Now he chooses two exponents, e and d

If he chooses e to be 13, then d is 37.

Note that $e \times d \bmod 60 = 1$ (they are inverses of each other).

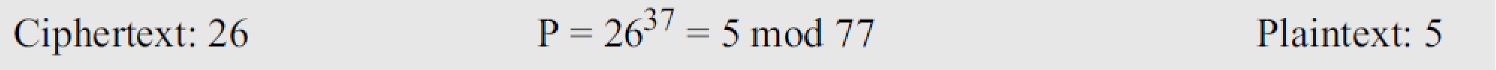
Now imagine that Alice wants to send the plaintext 5 to Bob. She uses the public exponent 13 to encrypt 5.



Plaintext: 5 $C = 5^{13} = 26 \bmod 77$ Ciphertext: 26

Some Trivial Examples

Bob receives the ciphertext 26 and uses the private key 37 to decipher the ciphertext:



Ciphertext: 26 $P = 26^{37} = 5 \bmod 77$ Plaintext: 5

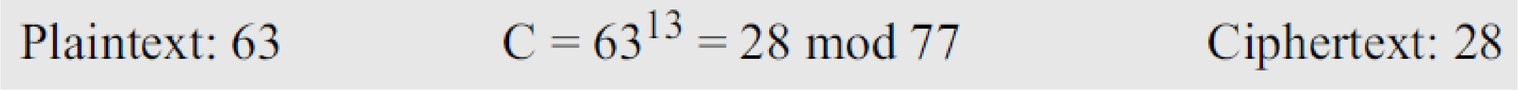
The plaintext 5 sent by Alice is received as plaintext 5 by Bob.

Some Trivial Examples

Example 10.6

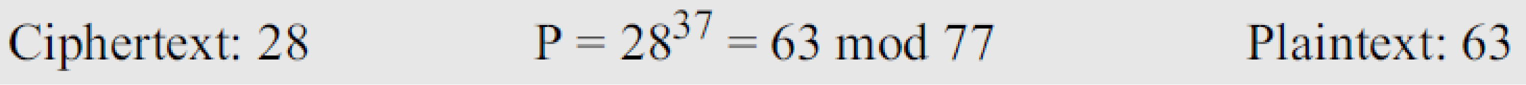
Now assume that another person, John, wants to send a message to Bob.

John can use the same public key announced by Bob (probably on his website), 13; John’s plaintext is 63. John calculates the following:



Plaintext: 63 $C = 63^{13} = 28 \bmod 77$ Ciphertext: 28

Bob receives the ciphertext 28 and uses his private key 37 to decipher the ciphertext



Ciphertext: 28 $P = 28^{37} = 63 \bmod 77$ Plaintext: 63

Some Trivial Examples

Example 10.7

Jennifer creates a pair of keys for herself. She chooses p = 397 and q = 401.

She calculates n = 397 × 401= 159197.

She then calculates φ(n) = 396 × 400 = 158400.

She then chooses e = 343 and d = 12007.

Show how Ted can send a message to Jennifer if he knows e and n.

Some Trivial Examples

Solution

Suppose Ted wants to send the message “NO” to Jennifer.

He changes each character to a number (from 00 to 25), with each character coded as two digits.

He then concatenates the two coded characters and gets a four-digit number.

The plaintext is 1314.

The ciphertext is $1314^{343} = 33677 \bmod 159197$.

Jennifer receives the message 33677 and uses the decryption key d to decipher it as $33677^{12007} = 1314 \bmod 159197$.

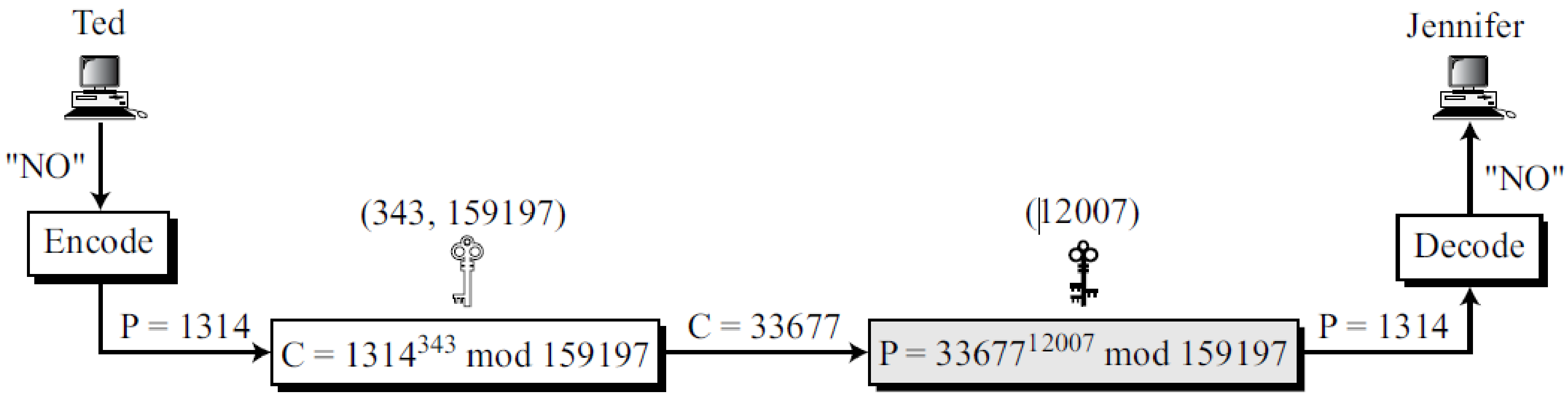
Jennifer then decodes 1314 as the message “NO”.

Some Trivial Examples

Example 10.7

Solution

Figure 10.7 *Encryption and decryption in Example 10.7*



10.3 RABIN CRYPTOSYSTEM

Devised by M. Rabin.

Is a variation of the RSA cryptosystem.

RSA is based on the exponentiation congruence; Rabin is based on quadratic congruence.

Rabin cryptosystem can be thought of as an RSA cryptosystem in which the value of e and d are fixed; $e = 2$ and $d = 1/2$.

The encryption is $C \equiv P^2 \pmod{n}$

The decryption is $P \equiv C^{1/2} \pmod{n}$.

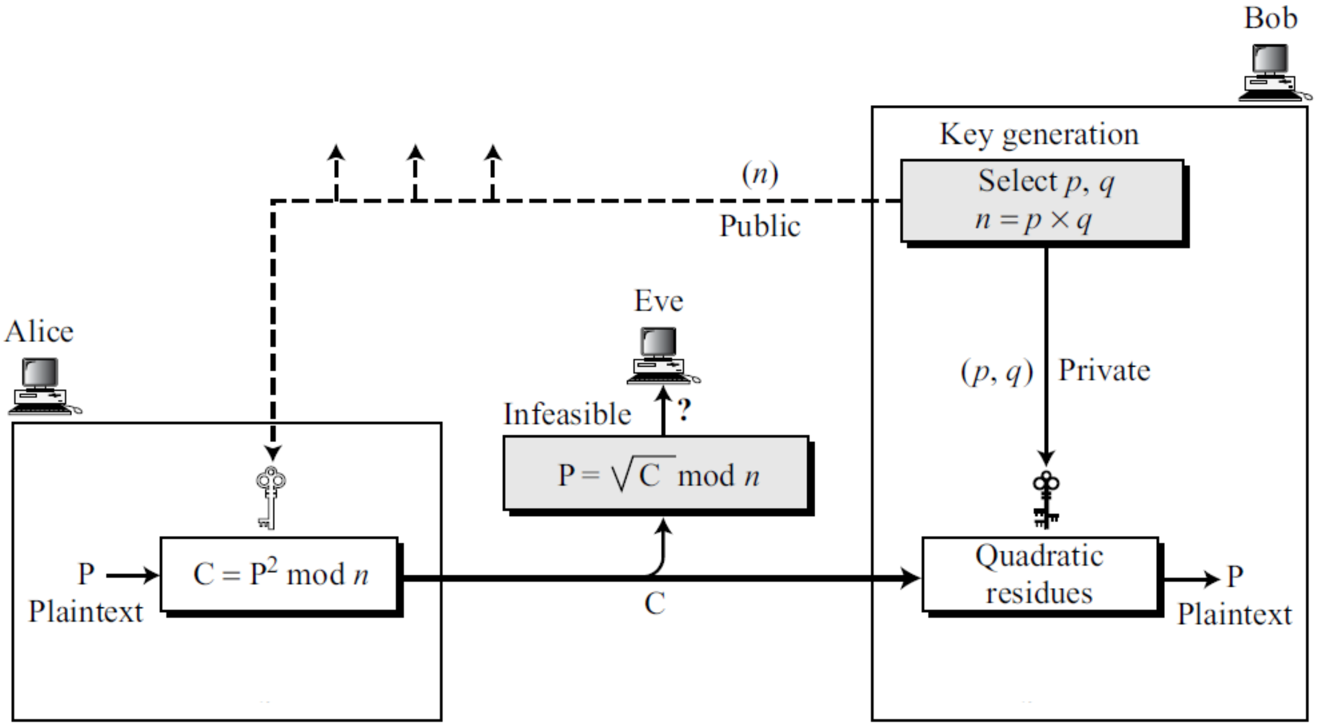
10.3 RABIN CRYPTOSYSTEM

public key in the Rabin cryptosystem is n

private key is the tuple (p, q).

10.3 RABIN CRYPTOSYSTEM

Figure 10.10 *Encryption, decryption, and key generation in the Rabin cryptosystem*



10.3 RABIN CRYPTOSYSTEM

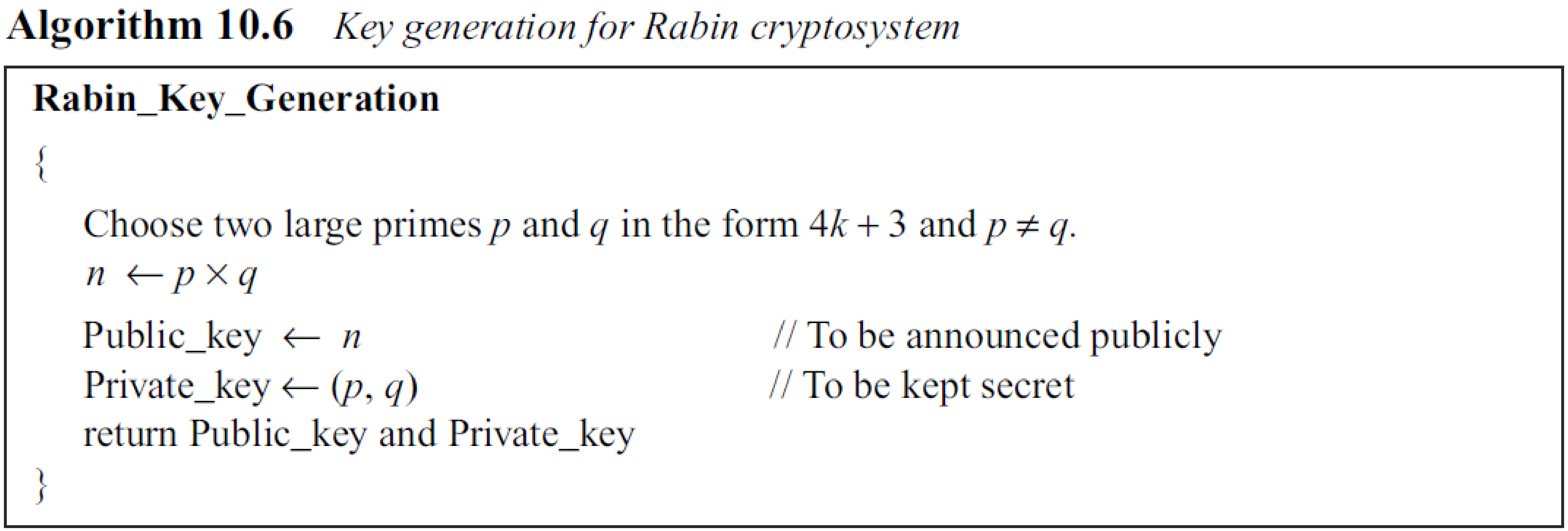
If Bob is using RSA, he can keep d and n and discard p, q, and φ(n) after key generation.

If Bob is using Rabin cryptosystem, he needs to keep p and q.

10.3 RABIN CRYPTOSYSTEM

Key generation, encryption, and decryption are.

Key Generation

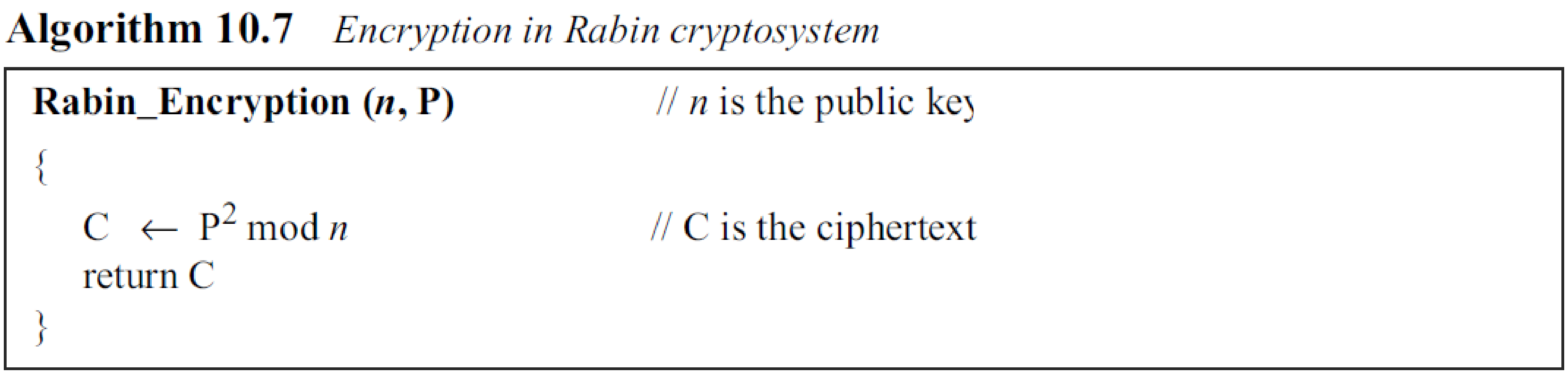


Algorithm 10.6 *Key generation for Rabin cryptosystem*

```
Rabin_Key_Generation
{
    Choose two large primes $p$ and $q$ in the form $4k + 3$ and $p \neq q$.
    $n \leftarrow p \times q$
    Public_key ← $n$                          // To be announced publicly
    Private_key ← $(p, q)$                    // To be kept secret
    return Public_key and Private_key
}
```

10.3 RABIN CRYPTOSYSTEM

Encryption



Algorithm 10.7 *Encryption in Rabin cryptosystem*

```
Rabin_Encryption ($n$, P)                // $n$ is the public key
{
    C ← $P^2$ mod $n$                      // C is the ciphertext
    return C
}
```

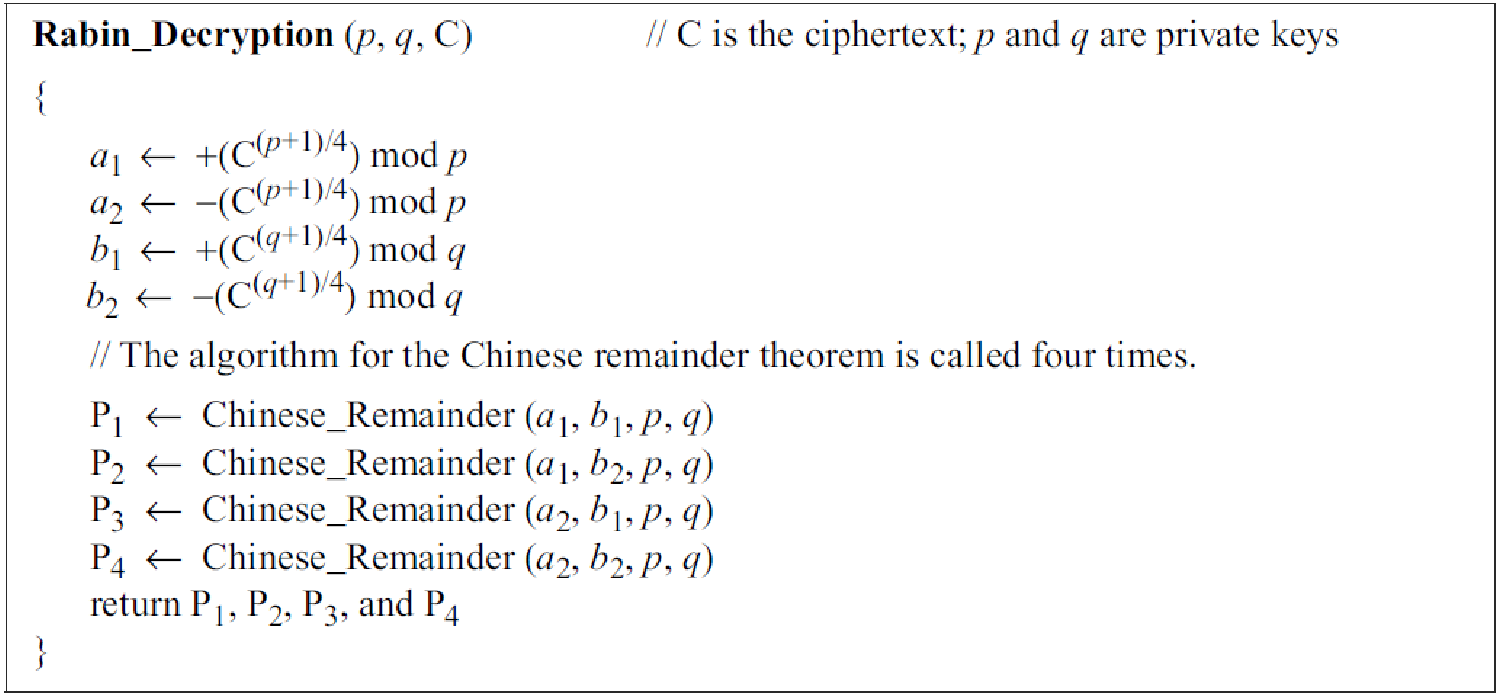
- Encryption in the Rabin cryptosystem is very simple. The operation needs only one multiplication, which can be done quickly. This is beneficial when resources are limited.

- For example, smart cards have limited memory and need to use short CPU time.

10.3 RABIN CRYPTOSYSTEM

Decryption

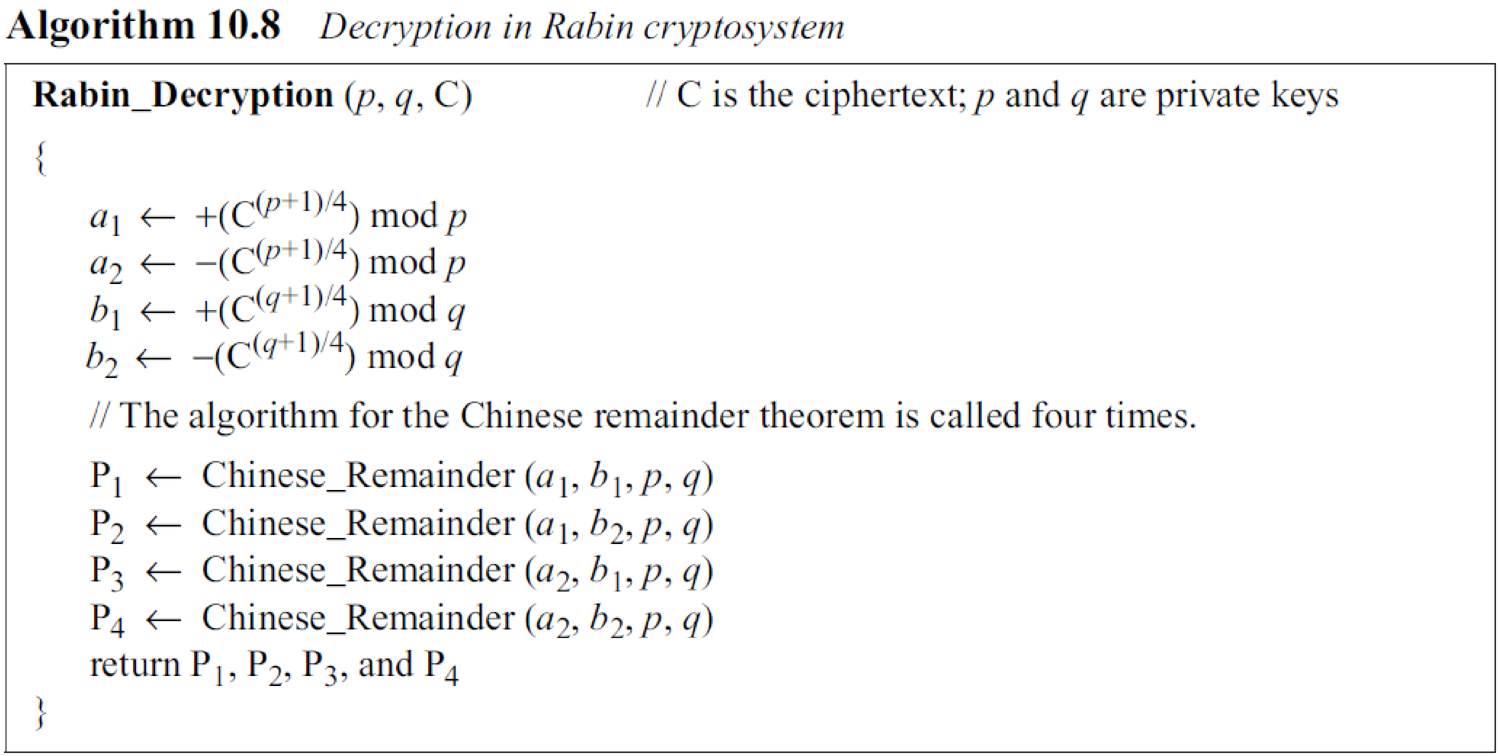
Algorithm 10.8 *Decryption in Rabin cryptosystem*



```
Rabin_Decryption ($p$, $q$, C)          // C is the ciphertext; $p$ and $q$ are private keys
{
    $a_1 \leftarrow +(C^{(p+1)/4}) \bmod p$
    $a_2 \leftarrow -(C^{(p+1)/4}) \bmod p$
    $b_1 \leftarrow +(C^{(q+1)/4}) \bmod q$
    $b_2 \leftarrow -(C^{(q+1)/4}) \bmod q$
    // The algorithm for the Chinese remainder theorem is called four times.
    $P_1 \leftarrow$ Chinese_Remainder ($a_1$, $b_1$, $p$, $q$)
    $P_2 \leftarrow$ Chinese_Remainder ($a_1$, $b_2$, $p$, $q$)
    $P_3 \leftarrow$ Chinese_Remainder ($a_2$, $b_1$, $p$, $q$)
    $P_4 \leftarrow$ Chinese_Remainder ($a_2$, $b_2$, $p$, $q$)
    return $P_1$, $P_2$, $P_3$, and $P_4$
}
```

10.3 RABIN CRYPTOSYSTEM

Decryption



Algorithm 10.8 *Decryption in Rabin cryptosystem*

```
Rabin_Decryption ($p$, $q$, C)          // C is the ciphertext; $p$ and $q$ are private keys
{
    $a_1 \leftarrow +(C^{(p+1)/4}) \bmod p$
    $a_2 \leftarrow -(C^{(p+1)/4}) \bmod p$
    $b_1 \leftarrow +(C^{(q+1)/4}) \bmod q$
    $b_2 \leftarrow -(C^{(q+1)/4}) \bmod q$
    // The algorithm for the Chinese remainder theorem is called four times.
    $P_1 \leftarrow$ Chinese_Remainder ($a_1, b_1, p, q$)
    $P_2 \leftarrow$ Chinese_Remainder ($a_1, b_2, p, q$)
    $P_3 \leftarrow$ Chinese_Remainder ($a_2, b_1, p, q$)
    $P_4 \leftarrow$ Chinese_Remainder ($a_2, b_2, p, q$)
    return $P_1$, $P_2$, $P_3$, and $P_4$
}
```

10.3 RABIN CRYPTOSYSTEM

The decryption has four answers.

It is up to the receiver of the message to choose one of the four as the final answer.

However, in many situations, the receiver can easily pick up the right answer.

10.3 RABIN CRYPTOSYSTEM

Example

1. Bob selects $p = 23$ and $q = 7$. Note that both are congruent to 3 mod 4.

2. Bob calculates $n = p \times q = 161$.

3. Bob announces n publicly; he keeps p and q private.

4. Alice wants to send the plaintext $P = 24$. Note that 161 and 24 are relatively prime; 24 is in $Z_{161}{}^*$.

She calculates $C = 24^2 = 93 \bmod 161$, and sends the ciphertext 93 to Bob.

10.3 RABIN CRYPTOSYSTEM

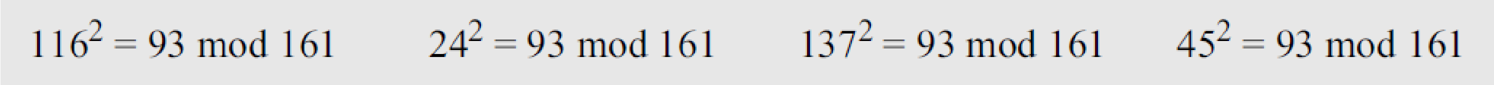
5. Bob receives 93 and calculates four values:

a. $a1 = +(93^{(23+1)/4}) \bmod 23 = 1 \bmod 23$
b. $a2 = -(93^{(23+1)/4}) \bmod 23 = 22 \bmod 23$
c. $b1 = +(93^{(7+1)/4}) \bmod 7 = 4 \bmod 7$
d. $b2 = -(93^{(7+1)/4}) \bmod 7 = 3 \bmod 7$

10.3 RABIN CRYPTOSYSTEM

6. Bob takes four possible answers, (a1, b1), (a1, b2), (a2, b1), and (a2, b2), and uses the Chinese remainder theorem to find four possible plaintexts: 116, 24, 137, and 45 (all of them relatively prime to 161).

Note that only the second answer is Alice’s plaintext. Bob needs to make a decision based on the situation. Note also that all four of these answers, when squared modulo n, give the ciphertext 93 sent by Alice.



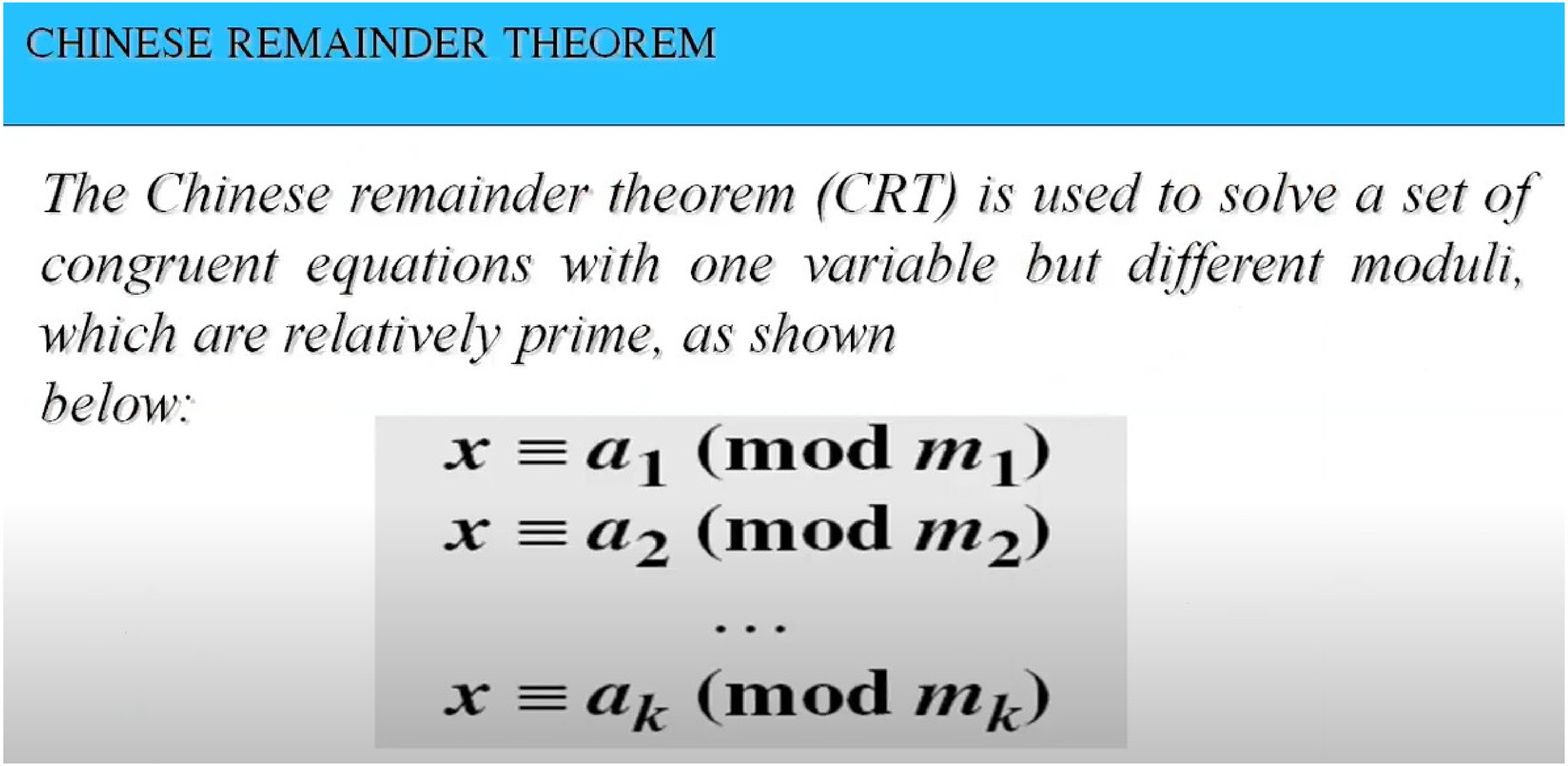
$116^2 = 93 \bmod 161$ $\quad$ $24^2 = 93 \bmod 161$ $\quad$ $137^2 = 93 \bmod 161$ $\quad$ $45^2 = 93 \bmod 161$

Security of the Rabin System

The Rabin system is secure as long as p and q are large numbers.

The complexity of the Rabin system is at the same level as factoring a large number n into its two prime factors p and q.

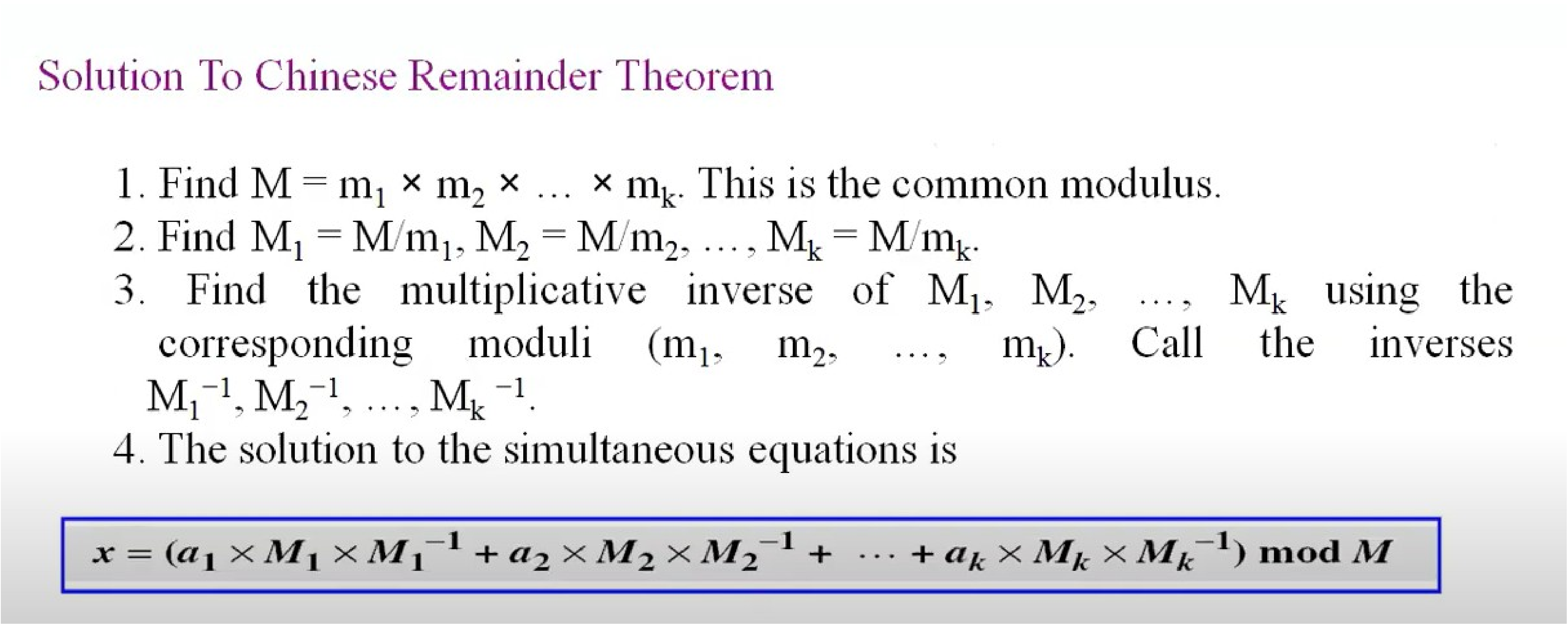
In other words, the Rabin system is as secure as RSA.



CHINESE REMAINDER THEOREM

The Chinese remainder theorem (CRT) is used to solve a set of congruent equations with one variable but different moduli, which are relatively prime, as shown below:

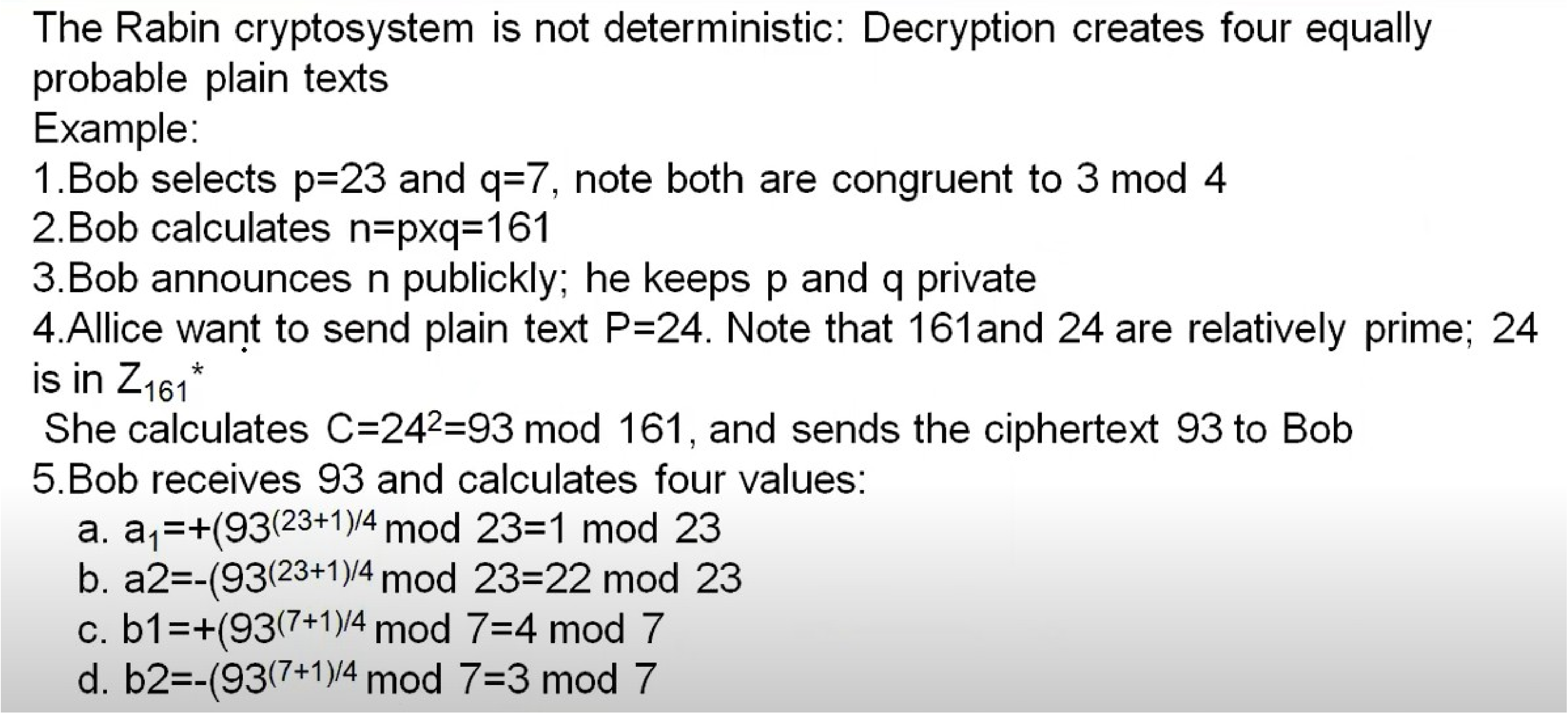
$$
\begin{gathered}
x \equiv a_1 \pmod{m_1} \\
x \equiv a_2 \pmod{m_2} \\
\cdots \\
x \equiv a_k \pmod{m_k}
\end{gathered}
$$



Solution To Chinese Remainder Theorem

1. Find $M = m_1 \times m_2 \times \ldots \times m_k$. This is the common modulus.
2. Find $M_1 = M/m_1$, $M_2 = M/m_2$, …, $M_k = M/m_k$.
3. Find the multiplicative inverse of M_1, M_2, …, M_k using the corresponding moduli (m_1, m_2, …, m_k). Call the inverses M_1^{-1}, M_2^{-1}, …, M_k^{-1}.
4. The solution to the simultaneous equations is

$$\boxed{x = (a_1 \times M_1 \times M_1^{-1} + a_2 \times M_2 \times M_2^{-1} + \cdots + a_k \times M_k \times M_k^{-1}) \bmod M}$$



The Rabin cryptosystem is not deterministic: Decryption creates four equally probable plain texts

Example:

1.Bob selects p=23 and q=7, note both are congruent to 3 mod 4

2.Bob calculates n=pxq=161

3.Bob announces n publickly; he keeps p and q private

4.Allice want to send plain text P=24. Note that 161and 24 are relatively prime; 24 is in Z_{161}^{*}

She calculates $C=24^{2}=93$ mod 161, and sends the ciphertext 93 to Bob

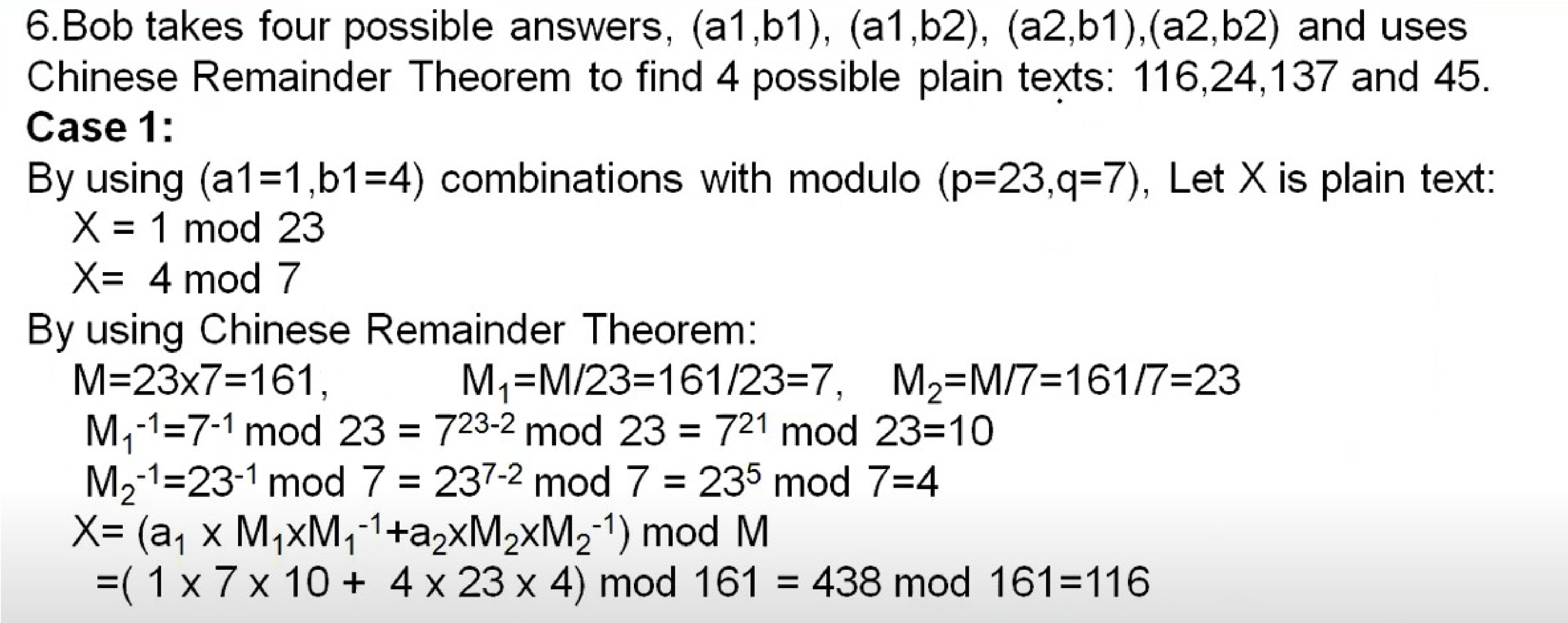
5.Bob receives 93 and calculates four values:

a. $a_1=+(93^{(23+1)/4}$ mod 23=1 mod 23

b. a2=-(93$^{(23+1)/4}$ mod 23=22 mod 23

c. b1=+(93$^{(7+1)/4}$ mod 7=4 mod 7

d. b2=-(93$^{(7+1)/4}$ mod 7=3 mod 7



6.Bob takes four possible answers, (a1,b1), (a1,b2), (a2,b1),(a2,b2) and uses Chinese Remainder Theorem to find 4 possible plain texts: 116,24,137 and 45.

Case 1:

By using (a1=1,b1=4) combinations with modulo (p=23,q=7), Let X is plain text:

$X = 1 \bmod 23$

$X = 4 \bmod 7$

By using Chinese Remainder Theorem:

$M=23x7=161,$ $M_1=M/23=161/23=7,$ $M_2=M/7=161/7=23$

$M_1^{-1}=7^{-1} \bmod 23 = 7^{23-2} \bmod 23 = 7^{21} \bmod 23=10$

$M_2^{-1}=23^{-1} \bmod 7 = 23^{7-2} \bmod 7 = 23^{5} \bmod 7=4$

$X= (a_1 \text{ x } M_1 x M_1^{-1}+a_2 x M_2 x M_2^{-1}) \bmod M$

$=(1 \text{ x } 7 \text{ x } 10 + 4 \text{ x } 23 \text{ x } 4) \bmod 161 = 438 \bmod 161=116$

Case 2:

By using (a1=1,b2=3) combinations with modulo (p=23,q=7), Let X is plain text:

$X = 1 \bmod 23$

$X = 3 \bmod 7$

By using Chinese Remainder Theorem:

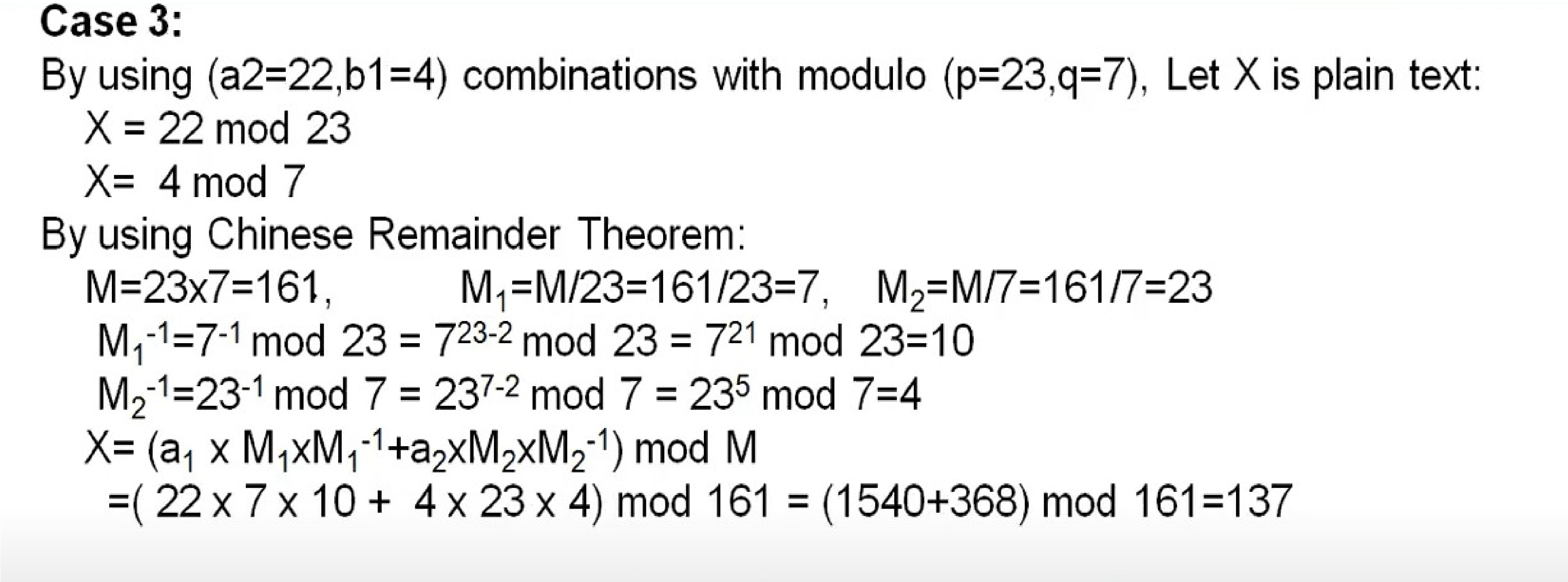
$M=23x7=161,$ $M_1=M/23=161/23=7,$ $M_2=M/7=161/7=23$

$M_1^{-1}=7^{-1} \bmod 23 = 7^{23-2} \bmod 23 = 7^{21} \bmod 23=10$

$M_2^{-1}=23^{-1} \bmod 7 = 23^{7-2} \bmod 7 = 23^{5} \bmod 7=4$

$X= (a_1 \ x \ M_1 x M_1^{-1} + a_2 x M_2 x M_2^{-1}) \bmod M$

$=(1 \ x \ 7 \ x \ 10 + \ 3 \ x \ 23 \ x \ 4) \bmod 161 = 346 \bmod 161=24$



Case 3:

By using (a2=22,b1=4) combinations with modulo (p=23,q=7), Let X is plain text:

$X = 22 \bmod 23$

$X= 4 \bmod 7$

By using Chinese Remainder Theorem:

$M=23x7=161,\qquad M_1=M/23=161/23=7,\quad M_2=M/7=161/7=23$

$M_1^{-1}=7^{-1} \bmod 23 = 7^{23-2} \bmod 23 = 7^{21} \bmod 23=10$

$M_2^{-1}=23^{-1} \bmod 7 = 23^{7-2} \bmod 7 = 23^{5} \bmod 7=4$

$X= (a_1 \text{ x } M_1 x M_1^{-1}+a_2 x M_2 x M_2^{-1}) \bmod M$

$=(22 \text{ x } 7 \text{ x } 10 + 4 \text{ x } 23 \text{ x } 4) \bmod 161 = (1540+368) \bmod 161=137$

Case 4:

By using (a2=22,b2=3) combinations with modulo (p=23,q=7), Let X is plain text:

$X = 22 \bmod 23$

$X = 4 \bmod 7$

By using Chinese Remainder Theorem:

$M=23x7=161$, $M_1=M/23=161/23=7$, $M_2=M/7=161/7=23$

$M_1^{-1}=7^{-1} \bmod 23 = 7^{23-2} \bmod 23 = 7^{21} \bmod 23=10$

$M_2^{-1}=23^{-1} \bmod 7 = 23^{7-2} \bmod 7 = 23^{5} \bmod 7=4$

$X= (a_1 \text{ x } M_1 x M_1^{-1}+a_2 x M_2 x M_2^{-1}) \bmod M$

$=(22 \text{ x } 7 \text{ x } 10 + 3 \text{ x } 23 \text{ x } 4) \bmod 161 = (1540+276) \bmod 161=45$

So, Finally from four cases: we got four plain text messages

case 1: 116 Case 2:24 Case 3: 137 Case 4: 45.

Only second answer 24 is Alice plain text, Bob needs to make a decision based on the situation