Asymmetric Key Cryptography

Introduction

- Symmetric and asymmetric-key cryptography will exist in parallel and continue to serve the community.
- They are complements of each other;
- the advantages of one can compensate for the disadvantages of the other.
- 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

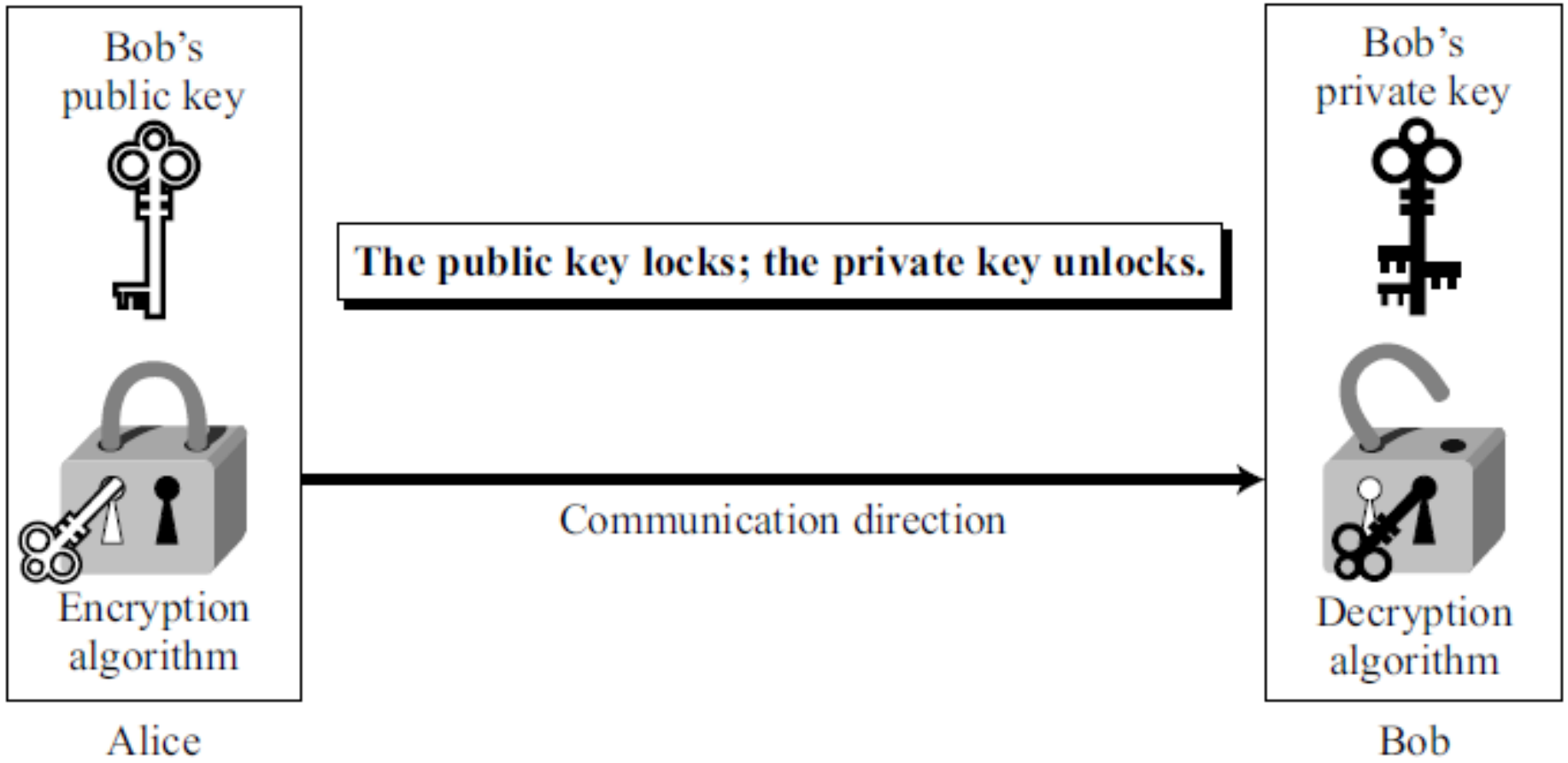
Introduction Contd..

- 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.
- For a community with a population of 1 million, symmetric-key cryptography would require half a billion shared secrets;
- asymmetric-key cryptography would require 1 million personal secrets.
- Besides encipherment, authentication and digital signatures that need asymmetric key cryptography

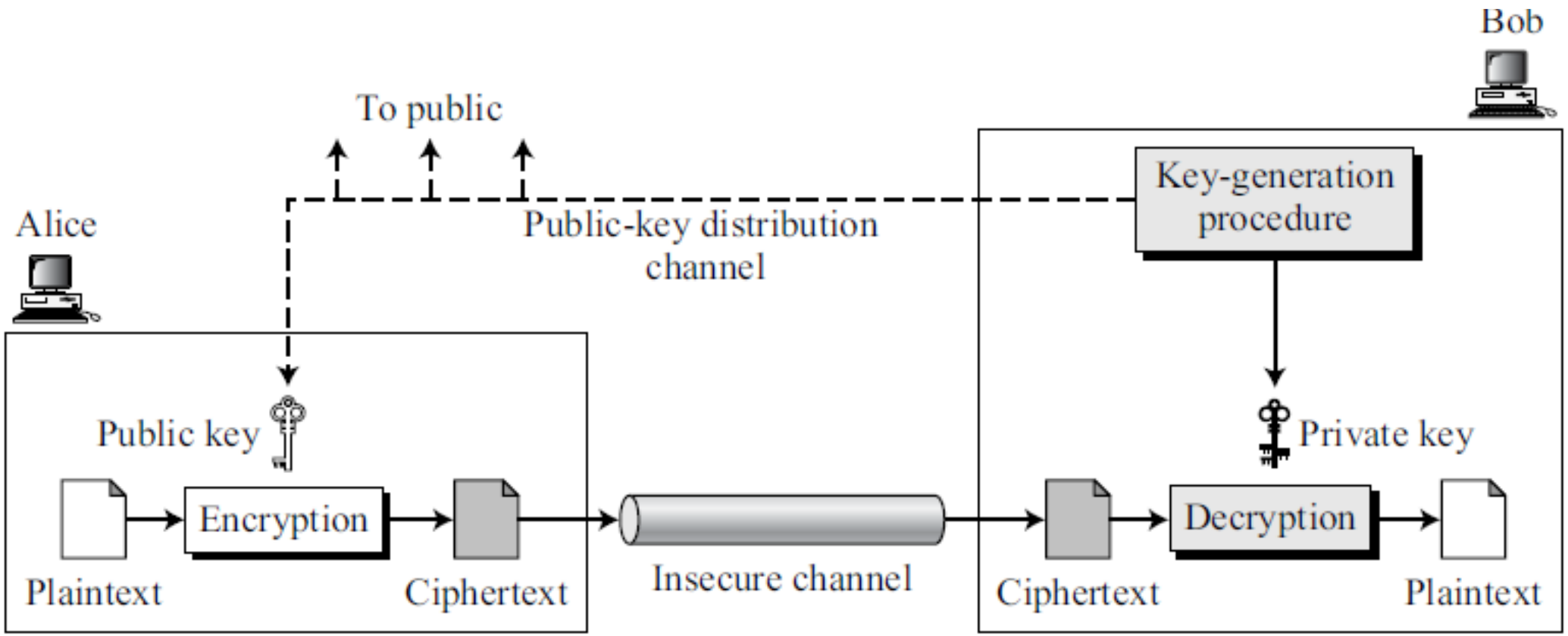
Introduction Contd..

- symmetric-key cryptography is based on substitution and permutation of symbols (characters or bits)
- asymmetric-key cryptography is based on applying mathematical functions to numbers.
- 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.
- Asymmetric key cryptography uses two separate keys: one private and one public.

Asymmetric key cryptography



General idea of asymmetric-key cryptosystem



Points to be noted w.r.t. Asymmetric key cryptography

- The burden of providing security is responsibility of the receiver
  - Bob needs to create two keys: one private and one public.
  - Bob is responsible for distributing the public key to the community-through a public-key distribution channel.
  - Although this channel is not required to provide secrecy, it must provide authentication and integrity.
  - Eve should not be able to advertise her public key to the community pretending that it is Bob's public key.
- 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.
  - Alice can use Bob's public key to send encrypted messages to Bob. If Bob wants to respond, Alice needs to establish her own private and public keys.
- 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.
  - In other words, Alice needs a ring of public keys.

Plaintext/Ciphertext

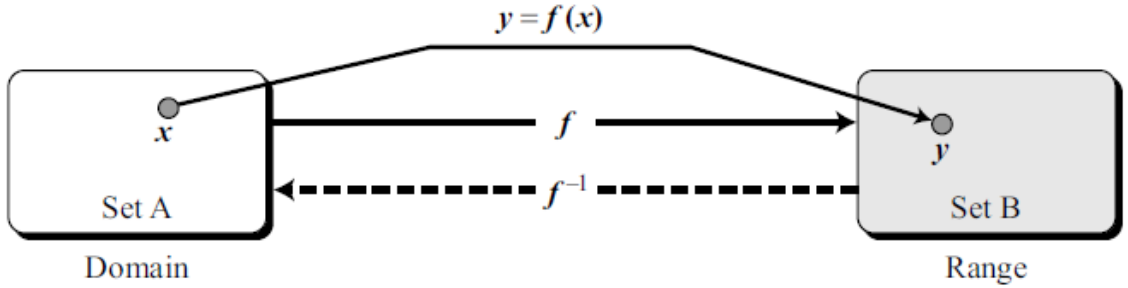
- plaintext and ciphertext are treated as integers in asymmetric-key cryptography.
- The message must be encoded as an integer (or a set of integers) before encryption; the integer (or the set of integers) must be decoded into the message after decryption.
- Asymmetric-key cryptography is normally used to encrypt or decrypt small pieces of information, such as the cipher key for a symmetrickey cryptography.
- In other words, asymmetric-key cryptography normally is used for ancillary goals instead of message encipherment.
- However, these ancillary goals play a very important role in cryptography today.

- C = f (Kpublic, P);
- P = g(Kprivate, C).
- The encryption function f is used only for encryption;
- the decryption function g is used only for decryption.
- the 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 asymmetrickey (public-key) cryptography does not eliminate the need for symmetric-key (secretkey) cryptography.
- The reason is that 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
- On the other hand, the 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.
- This means that, we need both symmetric-key and asymmetric-key cryptography. One complements the other.

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



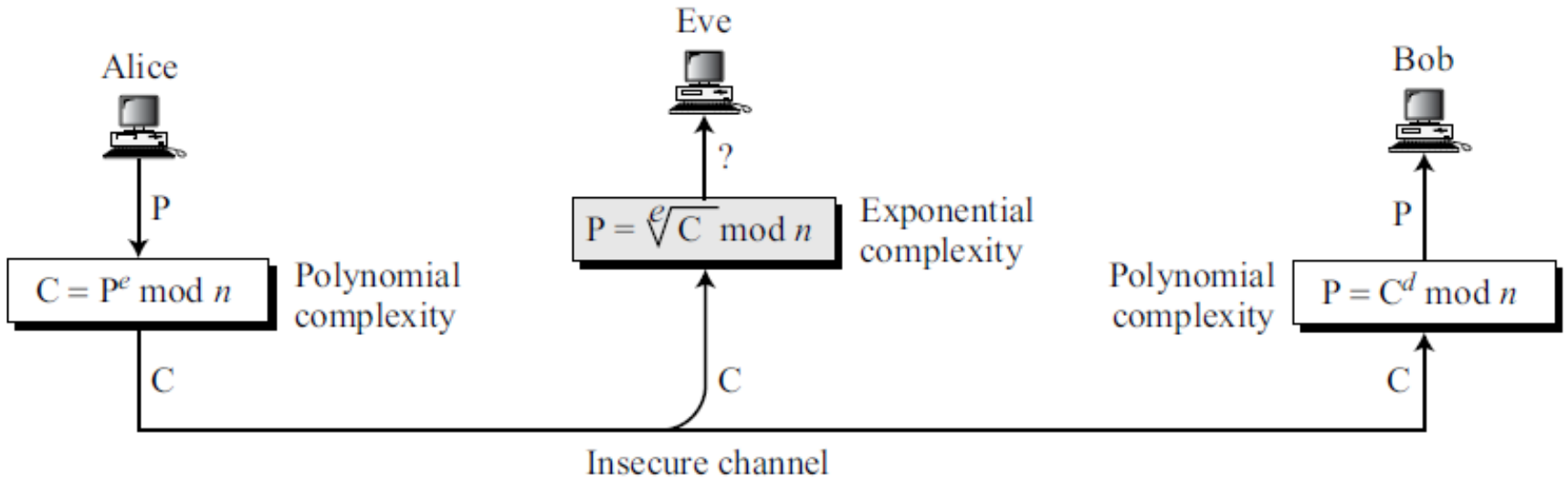
- An invertible function is a function that associates each element in the range with exactly one element in the domain
- one-way function (OWF) is a function that satisfies the following two properties
  1. f is easy to compute. In other words, given x, y = f (x) can be easily computed.
  2. f^{-1} is difficult to compute. In other words, given y, it is computationally infeasible to calculate $x = f^{-1}(y)$.
- A trapdoor one-way function (TOWF) is a one-way function with a third property:
  3. Given y and a trapdoor (secret), x can be computed easily.

- When n is large, $n = p \times q$ 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***
- 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***
- 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*

RSA CRYPTOSYSTEM (Rivest, Shamir, and Adleman)

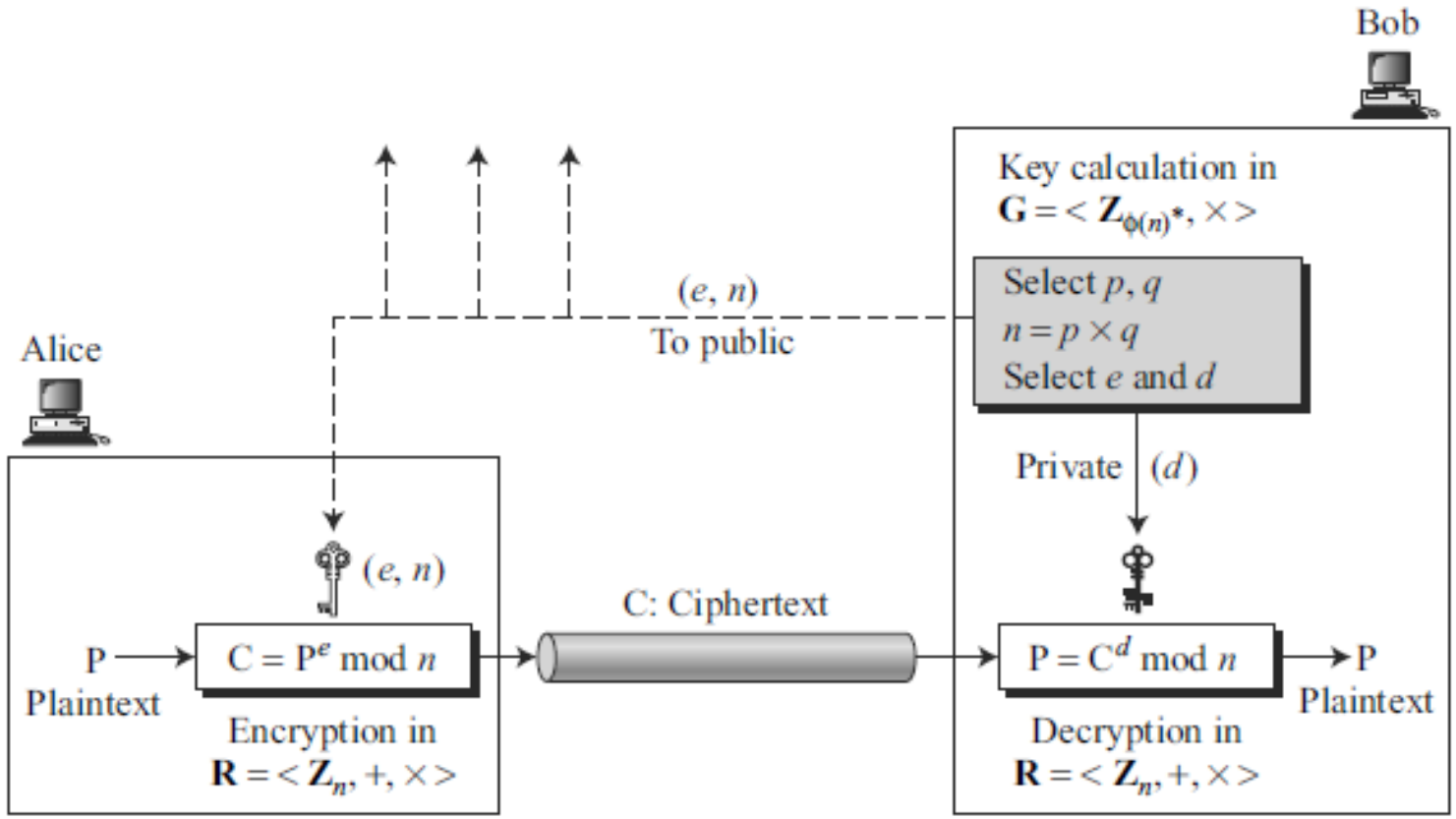
- RSA uses two exponents, e and d, where e is public and d is private.
- Suppose P is the plaintext and C is the ciphertext. Alice uses
- $C = P^e \bmod n$ to create ciphertext C fromplaintext P;
- Bob uses $P = C^d \bmod n$ to retrieve the plaintext sent by Alice.
- The modulus n, a very large number, is created during the key generation process
- 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.
- 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.

RSA



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 $\sqrt[e]{C} \bmod n$.


Bob
Key calculation in
$\mathbf{G} = < \mathbf{Z}_{\phi(n)^*}, \times >$
Select p, q
$n = p \times q$
Select e and d
Private (d)
(e, n)
To public
Alice
(e, n)
C: Ciphertext
P
Plaintext
$C = P^e \bmod n$
Encryption in
$\mathbf{R} = < \mathbf{Z}_n, +, \times >$
$P = C^d \bmod n$
Decryption in
$\mathbf{R} = < \mathbf{Z}_n, +, \times >$
P
Plaintext

RSA - Algebraic Structures Used

- RSA uses two algebraic structures: a ring and a group
- **Encryption/Decryption Ring :** Encryption and decryption are done using the commutative ring $R = <Zn, +, \times >$ with two arithmetic operations: addition and multiplication.
- In RSA, this ring is public because the modulus n is public.
- Anyone can send a message to Bob using this ring to do encryption.
- **Key-Generation Group :** RSA uses a multiplicative group $G = <Z\phi(n)*, \times >$ for key generation.
- This group supports only multiplication and division (using multiplicative inverses), which are needed for generating public and private keys.
- This group is hidden from the public because its modulus, $\phi(n)$, is hidden from the public.
- We will see shortly that if Eve can find this modulus, she can easily attack the cryptosystem.

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

}

- Bob create his public and private key using the steps shown
- 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
- To be secure, the 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).

- Anyone can send a message to Bob using his public key. Encryption in RSA can be done using an algorithm with polynomial time complexity,

```
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
}
```

```
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
}
```

- 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.

Proof of RSA

If $n = p \times q$, $a < n$, and k is an integer, then $a^{k\times\phi(n)+1} = a \pmod{n}$

$$P_1 = C^d \bmod n = (P^e \bmod n)^d \bmod n = P^{ed} \bmod n$$

$ed = k\phi(n) + 1$ // d and e are inverses modulo $\phi(n)$

$P_1 = P^{ed} \bmod n \rightarrow P_1 = P^{k\phi(n)+1} \bmod n$

$P_1 = P^{k\phi(n)+1} \bmod n = P \bmod n$ // Euler's theorem (second version)

Examples

- Bob chooses 7 and 11 as p and q and calculates
- $n = 7 \times 11 = 77$. The value of $\phi(n) = (7 - 1)(11 - 1) = 60$
- Now he chooses two exponents, e and d, from Z_{60}^*.
- 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).
- Alice wants to send the plaintext 5 to Bob. She uses the public exponent 13 to encrypt 5. $C = 5^{13} = 26 \bmod 77$
- Bob receives the ciphertext 26 and uses the private key 37 to decipher the ciphertext
- Ciphertext: 26 $P = 26^{37} = 5 \bmod 77$

plaintext = 63

Calculate ciphertext

Quadratic congruence

- Quadratic congruence– equations of the form $a_2x^2 + a_1x + a_0 \equiv 0 \pmod{n}$
- a2 = 1 and a1 = 0, that is equations of the form $x^2 \equiv a \pmod{n}$

Quadratic Congruence Modulo a Prime

- solutions for an equation of the form $x^2 \equiv a \pmod{p}$, in which p is a prime, a is an integer such that p does not divide a.
- this type of equation has either no solution or exactly two incongruent solutions
- The equation $x^2 \equiv 3 \pmod{11}$ has two solutions, $x \equiv 5 \pmod{11}$ and $x \equiv -5 \pmod{11}$.
- But $-5 \equiv 6 \pmod{11}$, so the solutions are actually 5 and 6 and these two solutions are incongruent
- The equation $x^2 \equiv 2 \pmod{11}$ has no solution.
- No integer x can be found such that its square is 2 mod 11.

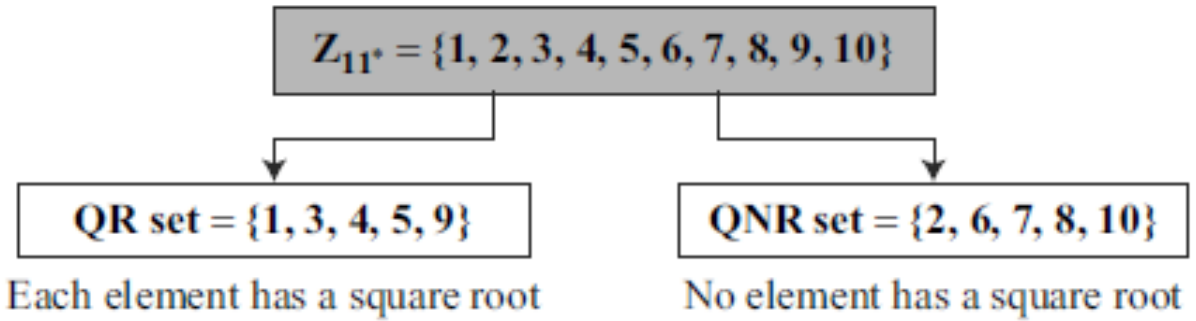
Quadratic Residues and Nonresidue

- In the equation $x^2 \equiv a \pmod{p}$,

  a is called a quadratic residue (QR) if the equation has two solutions;

  a is called quadratic nonresidue (QNR) if the equation has no solutions.

- It can be proved that in Z_p^* , with $p - 1$ elements, exactly $(p - 1)/2$ elements are quadratic residues and $(p - 1)/2$ are quadratic nonresidues
- There are 10 elements in Z_{11}^*. Exactly five of them are quadratic residues and five of them are nonresidues. In other words, Z_{11}^* is divided into two separate sets, QR and QNR



Euler's Criterion

How can we check to see if an integer is a QR modulo p?

Euler's criterion gives a very specific condition:

- If $a^{(p-1)/2} \equiv 1 \pmod{p}$, a is a quadratic residue modulo p.
- b. If $a^{(p-1)/2} \equiv -1 \pmod{p}$, a is a quadratic nonresidue modulo p
- To find out if 14 or 16 is a QR in Z_{23}^*, we calculate:

$14^{(23-1)/2} \bmod 23 \rightarrow 14^{11} \bmod 23 \rightarrow 22 \bmod 23 \rightarrow -1 \bmod 23$ **nonresidue**

$16^{(23-1)/2} \bmod 23 \rightarrow 16^{11} \bmod 23 \rightarrow 1 \bmod 23$ **residue**

Solving Quadratic Equation Modulo a Prime

- To find the solution to this quadratic equation, a prime can be either p = 4k + 1 or p = 4k + 3, in which k is a positive integer.
- The solution to a quadratic equation is very involved in the first case; it is easier in the second
- Special Case: p = 4k + 3 If p is in the form 4k + 3 (that is, p ≡ 3 mod 4) and a is a QR in Zp*, then

$$x \equiv a^{(p+1)/4} \pmod{p} \qquad \text{and} \qquad x \equiv -a^{(p+1)/4} \pmod{p}$$

- Solve the following quadratic equations:

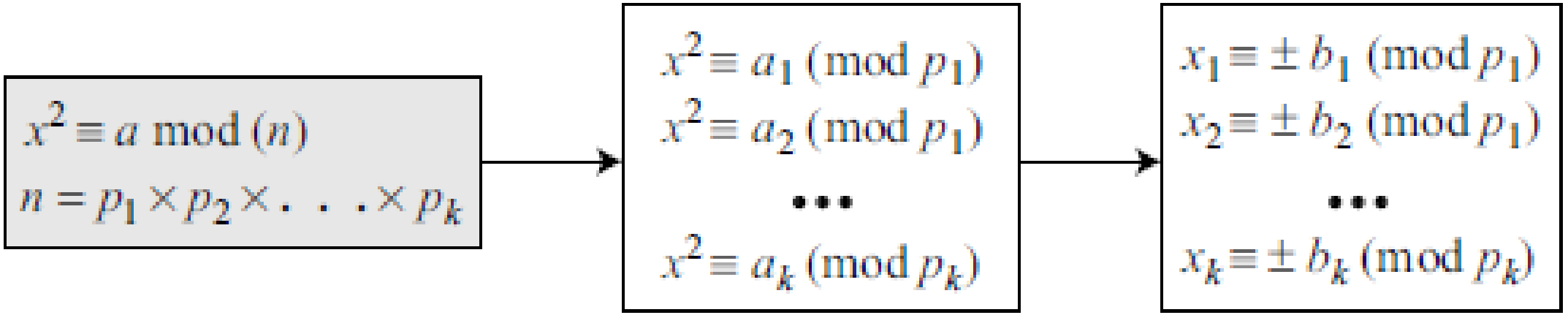
a. $x^2 \equiv 3 \pmod{23}$

b. $x^2 \equiv 2 \pmod{11}$

c. $x^2 \equiv 7 \pmod{19}$

- In the first equation, 3 is a QR in Z_{23}. The solution is $x \equiv \pm 16 \pmod{23}$. In other words, $\sqrt{3} \equiv \pm 16 \pmod{23}$.
- In the second equation, 2 is a QNR in Z_{11}. There is no solution for $\sqrt{2}$ in Z_{11}.
- In the third equation, 7 is a QR in Z_{19}. The solution is $x \equiv \pm 11 \pmod{19}$. In other words, $\sqrt{7} \equiv \pm 11 \pmod{19}$.

Quadratic Congruence Modulo a Composite



- Assume that $x^2 \equiv 36 \pmod{77}$. We know that $77 = 7 \times 11$. We can write

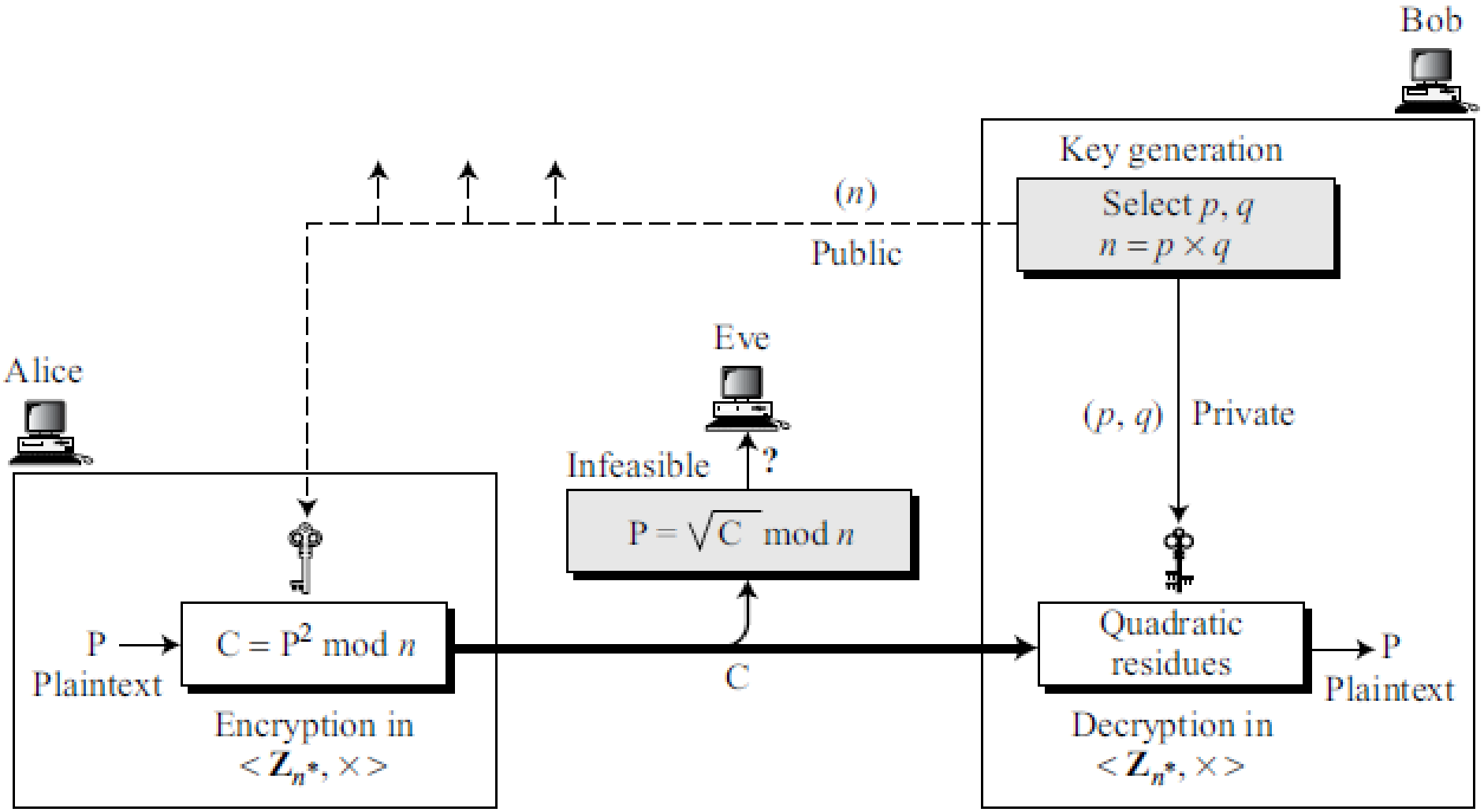
$$x^2 \equiv 36 \pmod 7 \equiv 1 \pmod 7 \qquad \text{and} \qquad x^2 \equiv 36 \pmod{11} \equiv 3 \pmod{11}$$

$$\begin{aligned}
&\textbf{Set 1: } x \equiv +1 \pmod 7 && x \equiv +5 \pmod{11} \\
&\textbf{Set 2: } x \equiv +1 \pmod 7 && x \equiv -5 \pmod{11} \\
&\textbf{Set 3: } x \equiv -1 \pmod 7 && x \equiv +5 \pmod{11} \\
&\textbf{Set 4: } x \equiv -1 \pmod 7 && x \equiv -5 \pmod{11}
\end{aligned}$$

- The answers are x = ± 6 and ± 27.
- the complexity of solving a quadratic congruence modulo a composite is the same as factorizing a composite integer.

RABIN CRYPTOSYSTEM

- The Rabin cryptosystem can be thought of as an RSA cryptosystem in which the value of e and d are fixed; e = 2
- $C \equiv P^{2} (\text{mod } n)$ and the decryption is $P \equiv C^{1/2} (\text{mod } n)$.
- The public key in the Rabin cryptosystem is n; the private key is the tuple (p, q)
- Everyone can encrypt a message using n; only Bob can decrypt the message using p and q.
- Decryption of the message is infeasible for Eve because she does not know the values of p and q.
- If Bob is using RSA, he can keep d and n and discard p, q, and $\phi(n)$ after key generation. If Bob is using Rabin cryptosystem, he needs to keep p and q.


Bob
Key generation
Select p, q
$n = p \times q$
(n)
Public
Alice
Eve
(p, q) Private
Infeasible
?
$P = \sqrt{C} \bmod n$
P
Plaintext
$C = P^2 \bmod n$
C
Quadratic residues
P
Plaintext
Encryption in
$< \mathbf{Z}_n^*, \times >$
Decryption in
$< \mathbf{Z}_n^*, \times >$

Key generation for Rabin cryptosystem

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

Although the two primes, p and q, can be in the form 4k + 1 or 4k + 3, the decryption process becomes more difficult if the first form is used. It is recommended to use the second form, 4k + 3, to make the decryption for Alice much easier.

Encryption in Rabin cryptosystem

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

Although the plaintext P can be chosen from the set Zn, we have defined the set to be in Zn* to make the decryption easier.
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.

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

}

- Rabin system is that it is not deterministic.
- 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.

Example

p = 23 and q = 7

n = p × q = 161

P = 24

$C = 24^2 = 93 \bmod 161$

Receiver receives 93 and calculates four values:

$a1 = +(93^{(23+1)/4}) \bmod 23 = 1 \bmod 23$

$a2 = -(93^{(23+1)/4}) \bmod 23 = 22 \bmod 23$

$b1 = +(93^{(7+1)/4}) \bmod 7 = 4 \bmod 7$

$b2 = -(93^{(7+1)/4}) \bmod 7 = 3 \bmod 7$

Using CRT, find P

- (a1, b1), (a1, b2), (a2, b1), and (a2, b2), and use the Chinese remainder theorem to find four possible plaintexts: 116, 24, 137, and 45

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.

Order of an Element

- The order of an element, a, is the smallest integer i such that $a^i \equiv e \pmod{n}$. The identity element e is 1 in this case.
- Find the order of all elements in G = <Z10∗, ×>
- order of an element divides the order of the group (Lagrange theorem).
- The only integers that divide 4 are 1, 2, and 4
- we need to check only these powers to find the order of the element

$$1^1 \equiv 1 \bmod (10) \rightarrow \text{ord}(1) = 1.$$
$$3^1 \equiv 3 \bmod (10);\ 3^2 \equiv 9 \bmod (10);\ 3^4 \equiv 1 \bmod (10) \rightarrow \text{ord}(3) = 4.$$
$$7^1 \equiv 7 \bmod (10);\ 7^2 \equiv 9 \bmod (10);\ 7^4 \equiv 1 \bmod (10) \rightarrow \text{ord}(7) = 4.$$
$$9^1 \equiv 9 \bmod (10);\ 9^2 \equiv 1 \bmod (10) \rightarrow \text{ord}(9) = 2.$$

- shows the result of ai ≡ x (mod 8) for the group G = <Z8∗, ×>. Note that ϕ(8) = 4. The elements are 1, 3, 5, and 7.

	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$	$i = 6$	$i = 7$
$a = 1$	**x: 1**	x: 1	x: 1	x: 1	x: 1	x: 1	x: 1
$a = 3$	x: 3	**x: 1**	x: 3	x: 1	x: 3	x: 1	x: 3
$a = 5$	x: 5	**x: 1**	x: 5	x: 1	x: 5	x: 1	x: 5
$a = 7$	x: 7	**x: 1**	x: 7	x: 1	x: 7	x: 1	x: 7

- When i = ϕ(8) = 4, the result is x = 1 for every a.
- the value of x can be 1 for many values of i.
- The first time when x is 1, the value of i gives us the order of the element (double-sided boxes).
- The orders of elements are ord(1) = 1, ord(3) = 2, ord(5) = 2,and ord(7) = 2.

Primitive Roots

- In the group G = <Zn∗, ×>, when the order of an element is the same as ϕ(n), that element is called the primitive root of the group
- there are no primitive roots in G = <Z8∗, ×> because no element has the order equal to ϕ(8) = 4. The order of elements are all smaller than 4.

The group $G = \langle Z_n^*, \times \rangle$ has primitive roots only if n is 2, 4, p^t, or $2p^t$.

Example

For which value of n, does the group G = <Zn∗, ×> have primitive roots: 17, 20, 38, and 50?

a. G = <Z17∗, ×> has primitive roots, because 17 is a prime (pt where t is 1).

b. G = <Z20∗, ×> has no primitive roots.

c. G = <Z38∗, ×> has primitive roots, because 38 = 2 × 19 and 19 is a prime.

d. G = <Z50∗, ×> has primitive roots, because 50 = 2 × 52 and 5 is a prime.

Cyclic Groups

- The number of primitive roots can be calculated as ϕ(ϕ(n)). For example, the number of primitive roots of G = <Z17∗, ×> is ϕ(ϕ(17)) = ϕ(16) = 8
- if the group G = <Zn*, ×> has primitive roots, it is cyclic
- Each primitive root is a generator and can be used to create the whole set.
- In other words, if g is a primitive root in the group, we can generate the set Zn* as

$$\mathbf{Z}_n* = \{g^1, g^2, g^3, \ldots, g^{\phi(n)}\}$$

The group $G = <Z_n*, \times>$ is a cyclic group if it has primitive roots.
The group $G = <Z_p*, \times>$ is always cyclic.

Discrete Logarithm

The group G = <Zp*, ×> has several interesting properties:

1. Its elements include all integers from 1 to p − 1.

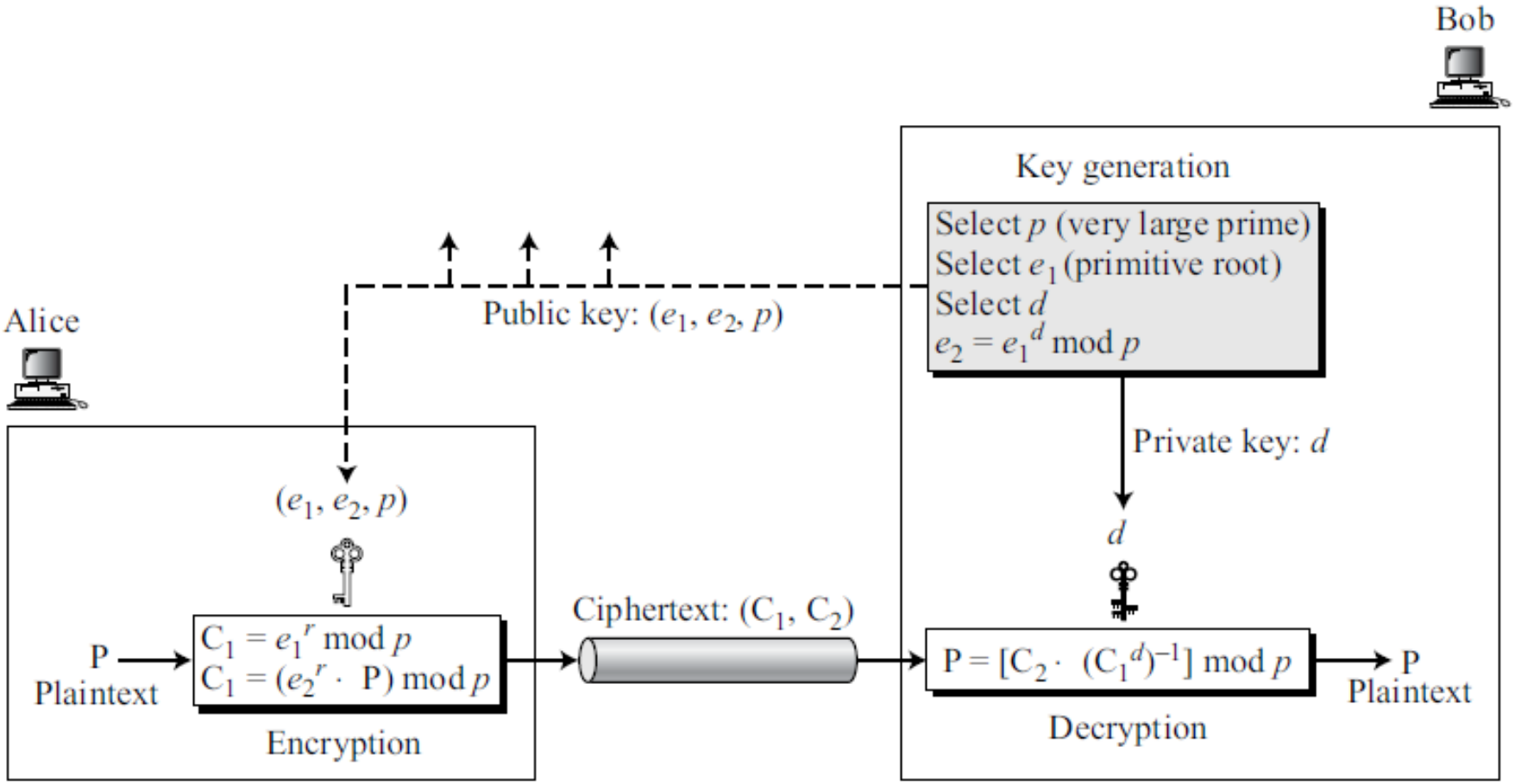
2. It always has primitive roots.

3. It is cyclic. The elements can be created using g^x where x is an integer from 1 to $\phi(n) = p - 1$.

4. The primitive roots can be thought as the base of logarithm. If the group has k primitive roots, calculations can be done in k different bases.

- Given $x = \log_g y$ for any element y in the set, there is another element x that is the log of y in base g. This type of logarithm is **called discrete logarithm.**
- problems of type $y = a^x \pmod n$ when y is given and we need to find x **discrete logarithm problem**

ELGAMAL CRYPTOSYSTEM



ElGamal_Key_Generation

{

Select a large prime p

Select d to be a member of the group $\mathbf{G} = < \mathbf{Z}_p^*, \times >$ such that $1 \leq d \leq p - 2$

Select e_1 to be a primitive root in the group $\mathbf{G} = < \mathbf{Z}_p^*, \times >$

$e_2 \leftarrow e_1^{\,d} \bmod p$

Public_key $\leftarrow$ (e_1, e_2, p) // To be announced publicly

Private_key $\leftarrow d$ // To be kept secret

return Public_key and Private_key

}

```
ElGamal_Encryption (e1, e2, p, P)                    // P is the plaintext
{
    Select a random integer r in the group G = < Zp*, × >
    C1  ←  e1^r mod p
    C2  ←  (P × e2^r) mod p                          // C1 and C2 are the ciphertexts
    return C1 and C2
}
```

```
ElGamal_Decryption (d, p, C1, C2)                    // C1 and C2 are the ciphertexts
{
    P  ←  [C2 (C1^d)^-1] mod p                       // P is the plaintext
    return P
}
```

The ElGamal decryption expression $C_2 \times (C_1{}^d)^{-1}$ can be verified to be P through substitution:

$$[C_2 \times (C_1{}^d)^{-1}] \bmod p = [(e_2{}^r \times P) \times (e_1{}^{rd})^{-1}] \bmod p = (e_1{}^{dr}) \times P \times (e_1{}^{rd})^{-1} = P$$

Example

- Bob chooses 11 as p.
- He then chooses e1 = 2. Note that 2 is a primitive root in Z11*
- Bob then chooses d = 3 and calculates e2 = $e1^d$ = 8.
- So the public keys are (2, 8, 11) and the private key is 3.
- Alice chooses r = 4 and calculates C1 and C2 for the plaintext 7.

Plaintext: 7

$$C_1 = e_1{}^r \bmod 11 = 16 \bmod 11 = 5 \bmod 11$$

$$C_2 = (P \times e_2{}^r) \bmod 11 = (7 \times 4096) \bmod 11 = 6 \bmod 11$$

Ciphertext: (5, 6)

Bob receives the ciphertexts (5 and 6) and calculates the plaintext

$$[C_2 \times (C_1{}^d)^{-1}] \bmod 11 = 6 \times (5^3)^{-1} \bmod 11 = 6 \times 3 \bmod 11 = 7 \bmod 11$$

Plaintext: 7

Elliptic Curve Cryptosystems

- RSA and ElGamal need large keys
- Elliptic curve cryptosystem (ECC) system is based on the theory of elliptic curves
- The general equation for an elliptic curve is

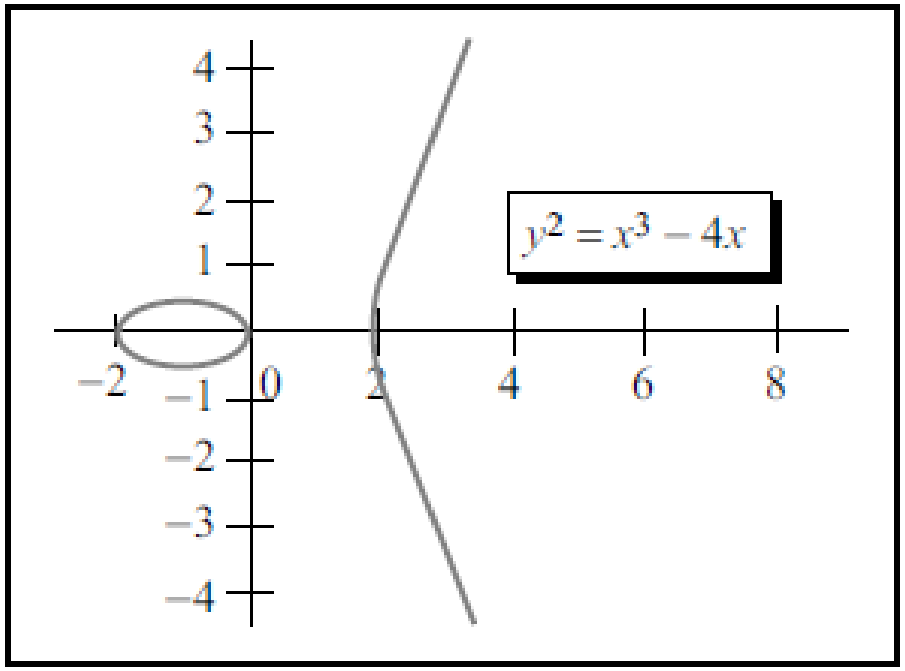
$$y^2 + b_1xy + b_2y = x^3 + a_1x^2 + a_2x + a_3$$

- Elliptic curves over real numbers use a special class of elliptic curves of the form

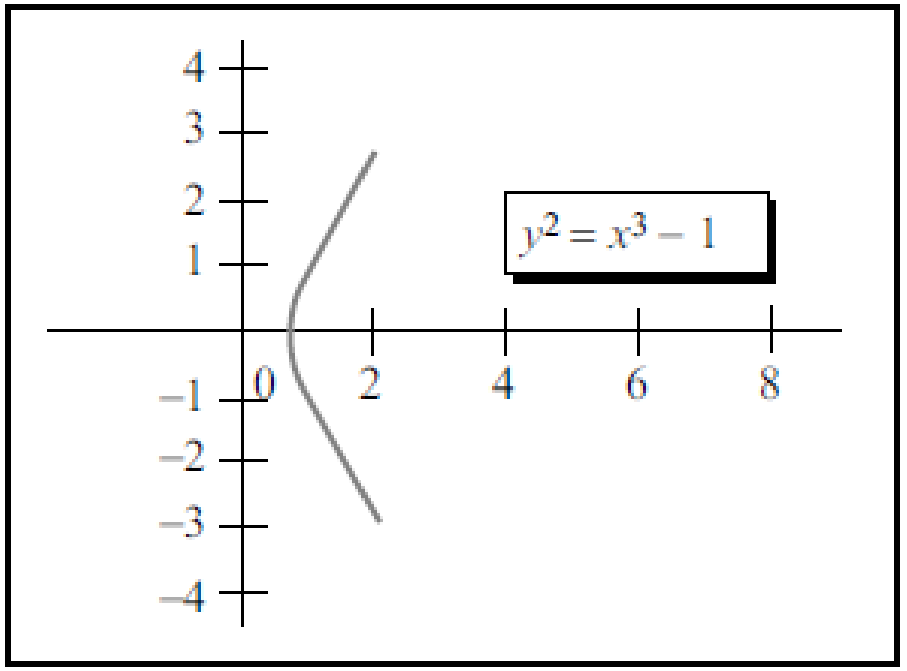
$$y^2 = x^3 + ax + b$$

- if $4a^3 + 27b^2 \neq 0$, the equation represents a nonsingular elliptic curve, otherwise, the equation represents a singular elliptic curve.
- In a nonsingular elliptic curve, the equation x3 + ax + b = 0 has three distinct roots (real or complex);
- in a singular elliptic curve the equation x3 + ax + b = 0 does not have three distinct roots.

- two elliptic curves with equations $y^2 = x^3 - 4x$ and $y^2 = x^3 - 1$.



a. Three real roots



b. One real and two imaginary roots

An Abelian Group

- A tuple P = (x1, y1) represents a point on the curve if x1 and y1 are the coordinates of a point on the curve that satisfy the equation of the curve
- The points P = (2.0, 0.0), Q = (0.0, 0.0), R = (−2.0, 0.0), S = (10.0, 30.98), and T = (10.0, −30.98) are all points on the curve $y^2 = x^3 - 4x$
- Each point is represented by two real numbers
- To create an abelian group we need a set, an operation on the set, and five properties that are satisfied by the operation.
- The group in this case is G=<E, +>.
- **Set** Points on the curve, where each point is a pair of real numbers
- Set E for the elliptic curve $y^2 = x^3 - 4x$ is shown as

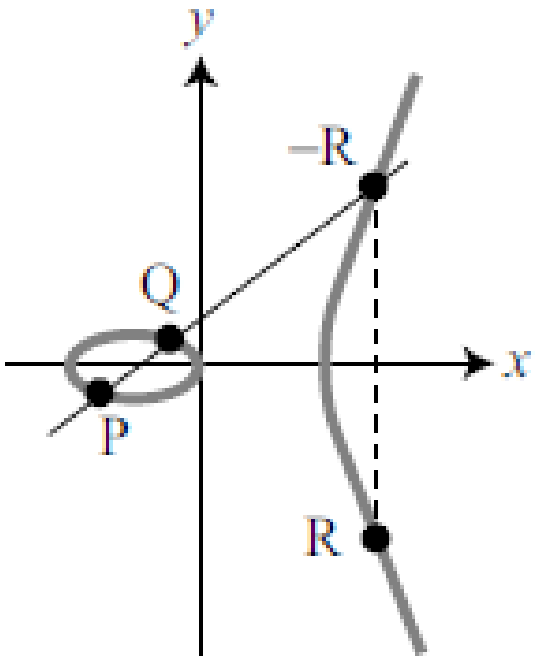
E = {(2.0, 0.0), (0.0, 0.0), (−2.0, 0.0), (10.0, 30.98), (10.0, −30.98), …}

Operation

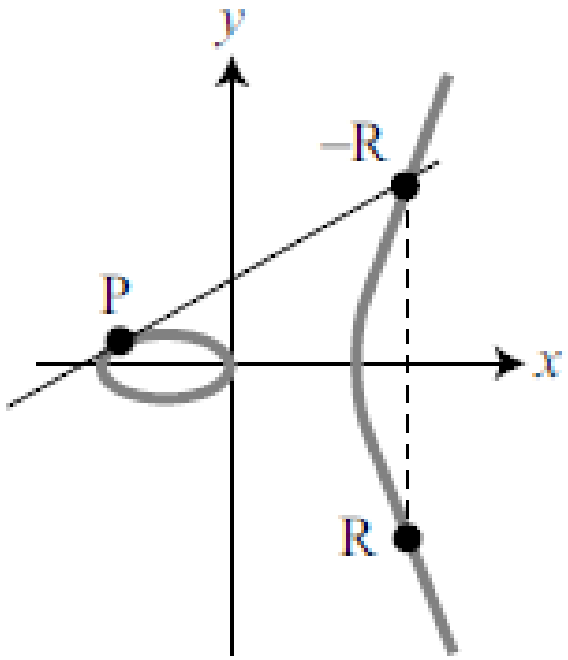
- The specific properties of a nonsingular elliptic curve allows us to define an addition operation on the points of the curve.
- the addition operation here is different from the operation that has been defined for integers.
- The operation is the addition of two points on the curve to get another point on the curve

$$R = P + Q, \text{ where } P = (x_1, y_1), Q = (x_2, y_2), \text{ and } R = (x_3, y_3)$$

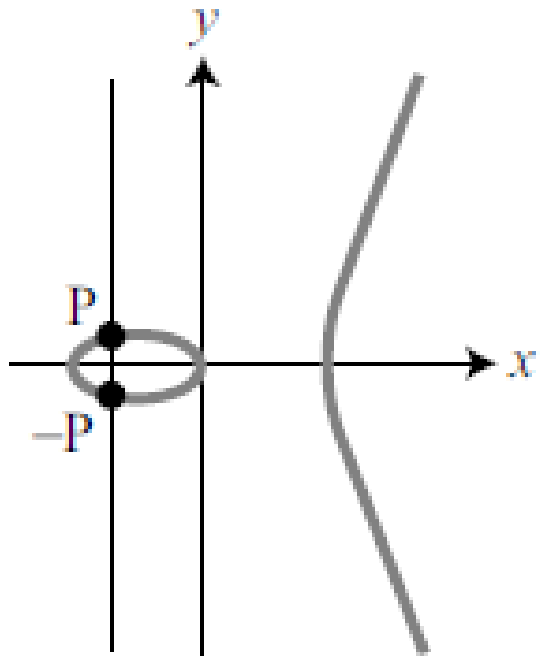
Three adding cases in an elliptic curve



a. (R = P + Q)



b. (R = P + P)



c. (*O* = P + (− P))

Case1: P = (x1, y1) and Q = (x2, y2)

$$\lambda = (y_2 - y_1) / (x_2 - x_1)$$

$$x_3 = \lambda^2 - x_1 - x_2 \qquad y_3 = \lambda (x_1 - x_3) - y_1$$

Case2: R=P+P

$$\lambda = (3x_1^2 + a)/(2y_1)$$

$$x_3 = \lambda^2 - x_1 - x_2 \qquad y_3 = \lambda (x_1 - x_3) - y_1$$

Case 3: P= (x1, y1), Q = (x1, −y1) - points are additive inverses of each other
The line connecting the two points does not intercept the curve at a third point
define a point O as the point at infinity or zero point, which is the additive identity of the group

Properties of the Operation

1. Closure: Adding two points, using the addition operation creates another point on the curve.

2. Associativity: It can be proven that $(P + Q) + R = P + (Q + R)$.

3. Commutativity: The group made from the points on a non-singular elliptic curve is an abelian group. It can be proven that $P + Q = Q + P$.

4. Existence of identity: The additive identity in this case is the zero point, O. In other words $P = P + O = O + P$.

5. Existence of inverse: Each point on the curve has an inverse. The inverse of a point is its reflection with respect to the x-axis. In other words, the point $P = (x1, y1)$ and $Q = (x1, -y1)$ are inverses of each other, which means that $P + Q = O$.

Elliptic Curves over GF(p)

- **Elliptic curve** Ep(a, b), where p defines the modulus and a and b are the coefficient of the equation y2 = x3 + ax + b.
- Value of x in this case ranges from 0 to p
- normally not all points are on the curve
- ***Finding an Inverse***
- The inverse of a point (x, y) is (x, −y), where −y is the additive inverse of y. For example,
- if p = 13, the inverse of (4, 2) is (4, 11).

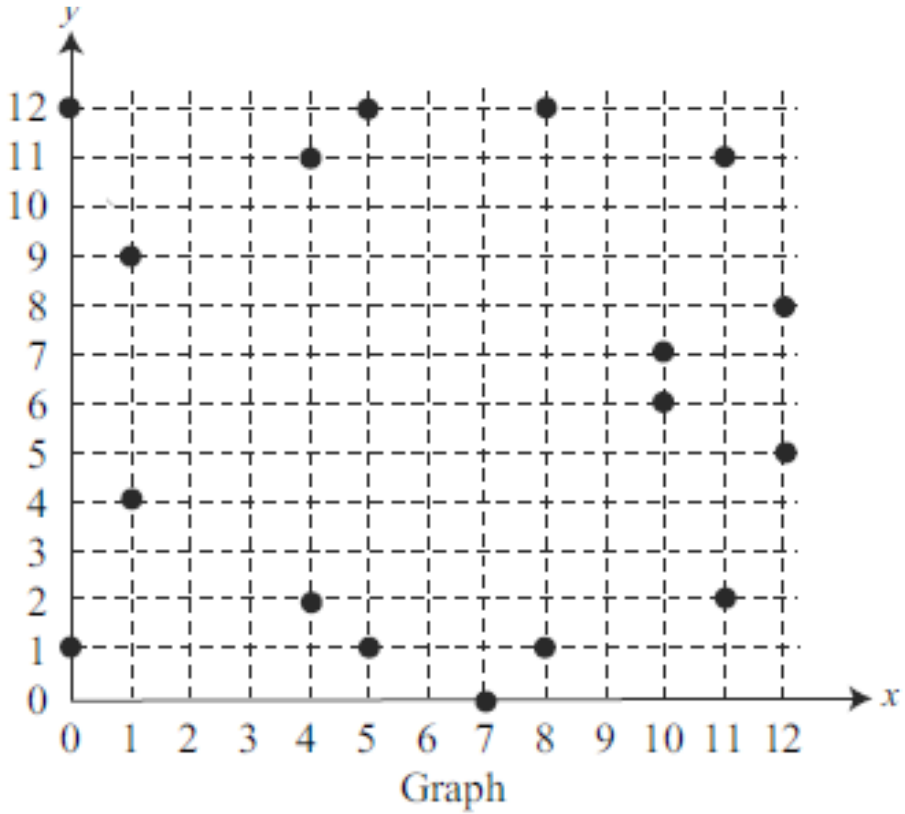
Pseudocode for finding points on an elliptic curve

```
ellipticCurve_points (p, a, b)                          // p is the modulus
{
    x ← 0
    while (x < p)
    {
        w ← (x^3 + ax + b) mod p                        // w is y^2
        if (w is a perfect square in Z_p) output (x, √w) (x, − √w)
        x ← x + 1
    {
}
```

- Define an elliptic curve $E_{13}(1, 1)$.The equation is y2 = x3 + x + 1. Find the points on the curve

(0, 1)	(0, 12)
(1, 4)	(1, 9)
(4, 2)	(4, 11)
(5, 1)	(5, 12)
(7, 0)	(7, 0)
(8, 1)	(8, 12)
(10, 6)	(10, 7)
(11, 2)	(11, 11)
(12, 5)	(12, 8)

Points



Graph

- Some values of y^2 do not have a square root in modulo 13 arithmetic.
- These are not points on this elliptic curve.
- Example, the points with x = 2, x = 3, x = 6, and x = 9 are not on the curve.
- Each point defined for the curve has an inverse. The inverses are listed as pairs. (7, 0) is the inverse of itself.
- Note that for a pair of inverse points, the y values are additive inverses of each other in Zp. For example, 4 and 9 are additive inverses in Z13.
- So if 4 is y, then 9 is −y.
- The inverses are on the same vertical lines.

Adding Two Points

- elliptic curve group defined earlier, but calculations are done in GF
- Let us add two points R = P + Q, where P = (4, 2) and Q = (10, 6).

a. $\lambda = (6 - 2) \times (10 - 4)^{-1} \bmod 13 = 4 \times 6^{-1} \bmod 13 = 5 \bmod 13$.

b. $x = (5^2 - 4 - 10) \bmod 13 = 11 \bmod 13$.

c. $y = [5\ (4 - 11) - 2] \bmod 13 = 2 \bmod 13$.

d. R = (11, 2), which is a point on the curve

Example

1. In the elliptic curve E(1, 2) over the GF(11) field:

a. Find the equation of the curve.

b. Find all points on the curve

2. In the elliptic curve E(1, 6) over the GF(7) field:

a. Find the equation of the curve.

b. Find all points on the curve

Multiplying a Point by a Constant

- In arithmetic, multiplying a number by a constant k means adding the number to itself k times.
- Multiplying a point P on an elliptic curve by a constant k means adding the point P to itself k times.
- For example, in E13 (1, 1), if the point (1, 4) is multiplied by 4, the result is the point (5, 1).
- If the point (8, 1) is multiplied by 3, the result is the point (10, 7).

Elliptic Curves over GF(2^n)

- elliptic curve group can be defined over the GF(2^n) field
- elements of the set in this field are n-bit words that can be interpreted as polynomials with coefficient in GF(2).
- Addition and multiplication on the elements are the same as addition and multiplication on polynomials
- elliptic curve over GF(2^n), is defined by

$$y^2 + xy = x^3 + ax^2 + b$$

where b ≠ 0. Note that the value of x, y, a, and b are polynomials representing n-bit words.

- ***Finding Inverses***

  If P = (x, y), then −P = (x, x + y)

- find the points on the curve using generators for polynomial
- GF(2^3) with elements {0, 1, g, g^2, g^3, g^4, g^5, g^6} using the irreducible polynomial of f(x) = x3 + x + 1, which means that $g^3 + g + 1 = 0$ or $g^3 = g + 1$. Other powers of g can be calculated accordingly. The following shows the values of the g's.

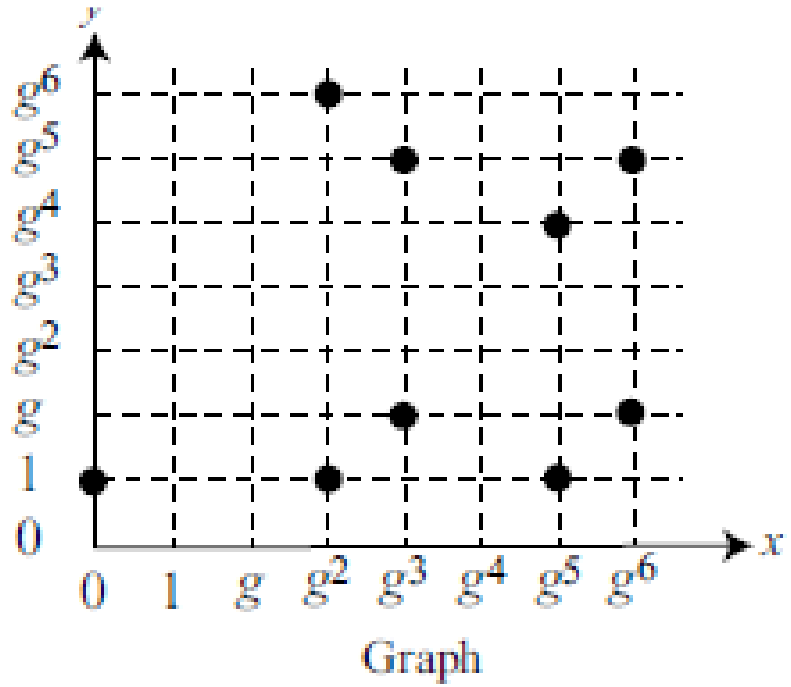
0	000	$g^3 = g + 1$	011
1	001	$g^4 = g^2 + g$	110
g	010	$g^5 = g^2 + g + 1$	111
g^2	100	$g^6 = g^2 + 1$	101

Points on an elliptic curve over GF(2^n)

- Using the elliptic curve $y^2 + xy = x^3 + g^3x^2 + 1$, with $a = g^3$ and $b = 1$, we can find the points on this curve,

$(0, 1)$	$(0, 1)$
$(g^2, 1)$	(g^2, g^6)
(g^3, g^2)	(g^3, g^5)
$(g^5, 1)$	(g^5, g^4)
(g^6, g)	(g^6, g^5)

Points



Graph

- finite field with the irreducible polynomial f(x)= x^4 +x+1
- . This yields a generator that satisfies with f(x)=0 with a value of

$g4 = g + 1$, or in binary, $g = 0010$.We can develop the powers of g as follows

$g^0 = 0001$	$g^4 = 0011$	$g^8 = 0101$	$g^{12} = 1111$
$g^1 = 0010$	$g^5 = 0110$	$g^9 = 1010$	$g^{13} = 1101$
$g^2 = 0100$	$g^6 = 1100$	$g^{10} = 0111$	$g^{14} = 1001$
$g^3 = 1000$	$g^7 = 1011$	$g^{11} = 1110$	$g^{15} = 0001$

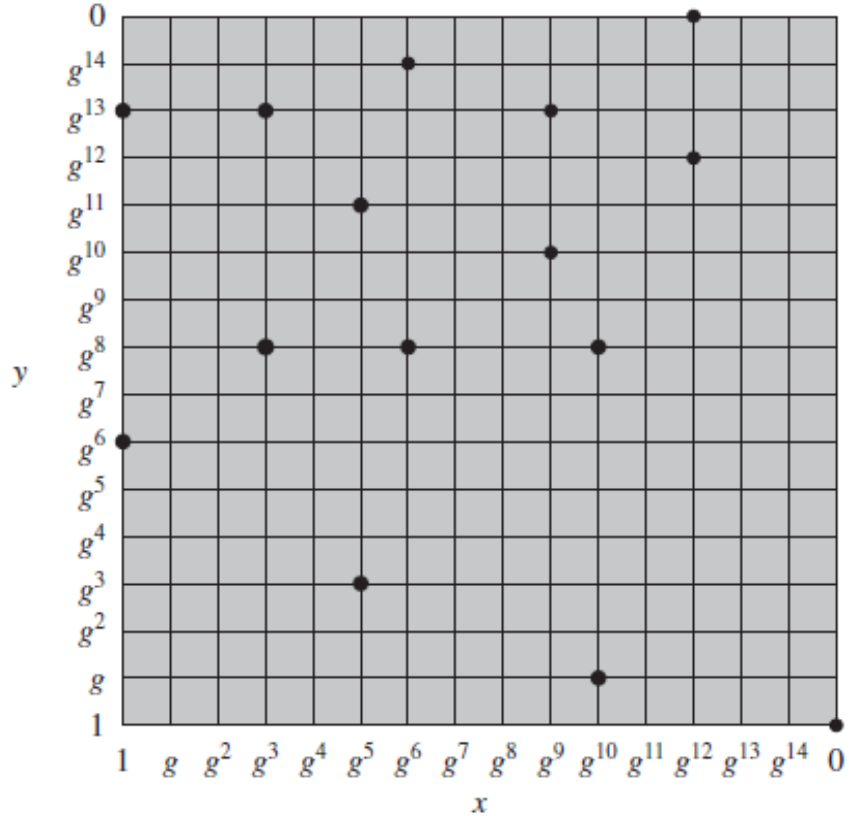


Figure 10.6 The Elliptic Curve $E_{2^4}(g^4, 1)$

The rules for adding points in GF(2^n) is slightly different from the rules for GF(p).

1. If P = (x1, y1), Q = (x2, y2), Q ≠ −P, and Q ≠ P, then R = (x3, y3) = P + Q can be found as

$$\lambda = (y_2 + y_1) / (x_2 + x_1)$$

$$x_3 = \lambda^2 + \lambda + x_1 + x_2 + a \qquad y_3 = \lambda (x_1 + x_3) + x_3 + y_1$$

If Q = P, then R = P + P (or R = 2P) can be found as

$$\lambda = x_1 + y_1 / x_1$$

$$x_3 = \lambda^2 + \lambda + a \qquad y_3 = x_1^2 + (\lambda + 1) x_3$$

R = P + Q, where P = (0, 1) and Q = (g^2, 1). We have λ = 0 and R = (g^5, g^4)

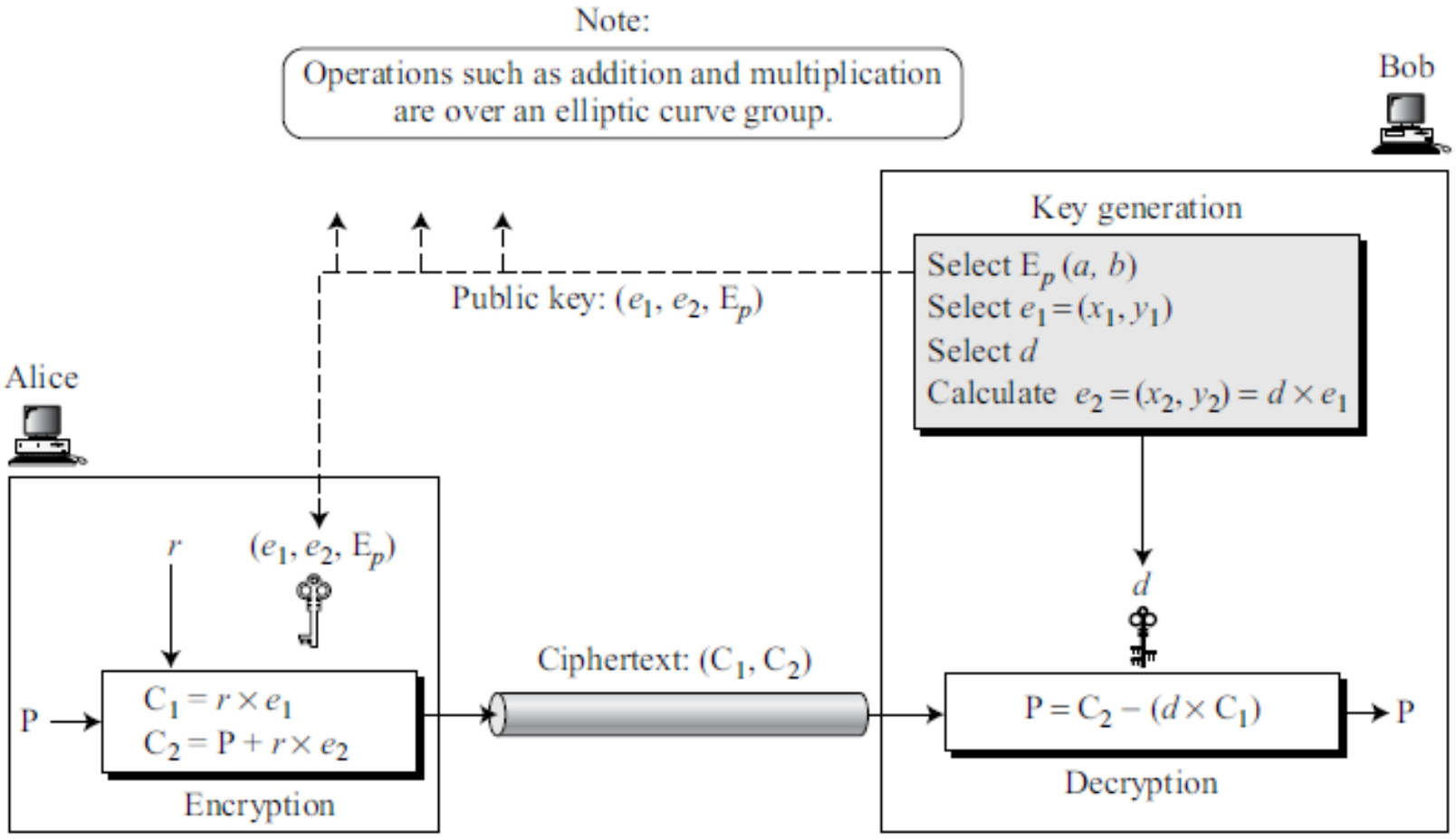
find R = 2P, where P = (g2, 1). We have λ = g2 + 1/g2 = g2 + g5 = g + 1 and R = (g6, g5)

Multiplying a Point by a Constant

To multiply a point by a constant, the points must be added continuously with attention to the rule for R = 2P.

Elliptic Curve Cryptography Simulating ElGamal

simulate the ElGamal cryptosystem using an elliptic curve over GF(p) or GF(2^n)



Generating Public and Private Keys

Generating Public and Private Keys

1. Bob chooses E(a, b) with an elliptic curve over GF(p) or GF(2n).

2. Bob chooses a point on the curve, e1(x1, y1).

3. Bob chooses an integer d.

4. Bob calculates e2(x2, y2) = d × e1(x1, y1). Note that multiplication here means multiple addition of points as defined before.

5. Bob announces E(a, b), e1(x1, y1), and e2(x2, y2) as his public key; he keeps d as his private key.

Encryption and Decryption

Encryption

Alice selects P, a point on the curve, as her plaintext, P. She then calculates a pair of points on the text as ciphertexts

$$C_1 = r \times e_1 \qquad C_2 = P + r \times e_2$$

Decryption

Bob, after receiving C1 and C2, calculates P, the plaintext using the following formula

$$P = C_2 - (d \times C_1)$$ The minus sign here means adding with the inverse.

P calculated by Bob is the same as that intended by Alice

$$P + r \times e_2 - (d \times r \times e_1) = P + (r \times d \times e_1) - (r \times d \times e_1) = P + O = P$$

Example

1. Bob selects $E_{67}(2, 3)$ as the elliptic curve over GF(p).
2. Bob selects e1 = (2, 22) and d = 4.
3. Bob calculates e2 = (13, 45), where e2 = d × e1.
4. Bob publicly announces the tuple (E, e1, e2).
5. Alice wants to send the plaintext P = (24, 26) to Bob. She selects r = 2.
6. Alice finds the point C1 = (35, 1), where C1 = r × e1.
7. Alice finds the point C2 = (21, 44), where C2 = P + r × e2.
8. Bob receives C1 and C2. He uses 2 × C1 (35, 1) to get (23, 25).
9. Bob inverts the point (23, 25) to get the point (23, 42).
10. Bob adds (23, 42) with C2 = (21, 44) to get the original plaintext P = (24, 26)

Diffie-Hellman key exchange

- enables two users to securely exchange a key that can then be used for subsequent encryption of messages
- depends for its effectiveness on the difficulty of computing discrete logarithms
- For any integer ***a*** and a primitive root ***b*** of prime number , we can find a unique exponent such that

$$b \equiv a^i \pmod{p} \qquad \text{where } 0 \leq i \leq (p-1)$$

- The exponent ***i*** is referred to as the **discrete logarithm** of for the base ***b*** , mod ***p*** expressed as

$$\text{dlog}_{a,p}(b)$$

- there are two publicly known numbers: a prime number q and an integer α
- Suppose the users A and B wish to exchange a key.
- User A selects a random integer $X_A < q$ and computes $Y_A = \alpha^{XA} \bmod q$ that is a primitive root of q
- B independently selects a random integer $X_B < q$ and computes $Y_B = \alpha^{XB} \bmod q$ that is a primitive root of q and computes
- Each side keeps the value private X and makes the value Y available publicly to the other side
- User A computes the key as $K = (Y_B)^{XA} \bmod q$ and user B computes the key as $K = (Y_A)^{XB} \bmod q$.
- These two calculations produce identical results
- The result is that the two sides have exchanged a secret value

$$
\begin{aligned}
K &= (Y_B)^{X_A} \bmod q \\
&= (\alpha^{X_B} \bmod q)^{X_A} \bmod q \\
&= (\alpha^{X_B})^{X_A} \bmod q \\
&= \alpha^{X_B X_A} \bmod q \\
&= (\alpha^{X_A})^{X_B} \bmod q \\
&= (\alpha^{X_A} \bmod q)^{X_B} \bmod q \\
&= (Y_A)^{X_B} \bmod q
\end{aligned}
$$

Global Public Elements

q	prime number
α	$\alpha < q$ and α a primitive root of q

User A Key Generation

Select private X_A	$X_A < q$
Calculate public Y_A	$Y_A = \alpha^{XA} \bmod q$

User B Key Generation

Select private X_B	$X_B < q$
Calculate public Y_B	$Y_B = \alpha^{XB} \bmod q$

Calculation of Secret Key by User A

$$K = (Y_B)^{XA} \bmod q$$

Calculation of Secret Key by User B

$$K = (Y_A)^{XB} \bmod q$$

Figure 10.1 The Diffie-Hellman Key Exchange Algorithm

Example

- $q = 353$, $\alpha = 3$, $X_A = 97$ and $X_B = 233$

$$\text{A computes } Y_A = 3^{97} \bmod 353 = 40.$$
$$\text{B computes } Y_B = 3^{233} \bmod 353 = 248.$$

After they exchange public keys, each can compute the common secret key:

$$\text{A computes } K = (Y_B)^{X_A} \bmod 353 = 248^{97} \bmod 353 = 160.$$
$$\text{B computes } K = (Y_A)^{X_B} \bmod 353 = 40^{233} \bmod 353 = 160.$$

Analog of Diffie-Hellman Key Exchange

- pick a large integer ***q*** , which is either a prime number ***p*** or an integer of the form 2^n , and elliptic curve parameters ***a*** and ***b which*** defines the elliptic group of points $E_q(a,b)$
- pick a *base point* $G = (x1, y1)$ in $Ep(a, b)$ whose order is a very large value n .
- The **order n** of a point on an elliptic curve is the smallest positive integer such that $nG = 0$ and G are parameters of the cryptosystem known to all participants.

A key exchange between users A and B can be accomplished as follows

1. A selects an integer n_A less than n. This is A's private key. A then generates a public key $P_A = n_A \times G$; the public key is a point in $E_q(a, b)$.
2. B similarly selects a private key n_B and computes a public key P_B.
3. A generates the secret key $k = n_A \times P_B$. B generates the secret key $k = n_B \times P_A$.

The two calculations in step 3 produce the same result because

$$n_A \times P_B = n_A \times (n_B \times G) = n_B \times (n_A \times G) = n_B \times P_A$$

Global Public Elements

$E_q(a, b)$	elliptic curve with parameters a, b, and q, where q is a prime or an integer of the form 2^m
G	point on elliptic curve whose order is large value n

User A Key Generation

Select private n_A	$n_A < n$
Calculate public P_A	$P_A = n_A \times G$

User B Key Generation

Select private n_B	$n_B < n$
Calculate public P_B	$P_B = n_B \times G$

Calculation of Secret Key by User A

$K = n_A \times P_B$

Calculation of Secret Key by User B

$K = n_B \times P_A$

Figure 10.7 ECC Diffie-Hellman Key Exchange

Exercise

Users A and B use the Diffie-Hellman key exchange technique with a common prime $q = 71$ and a primitive root $\alpha = 7$.

a. If user A has private key $X_A = 5$, what is A's public key Y_A?
b. If user B has private key $X_B = 12$, what is B's public key Y_B?
c. What is the shared secret key?

Consider a Diffie-Hellman scheme with a common prime $q = 11$ and a primitive root $\alpha = 2$.

a. Show that 2 is a primitive root of 11.
b. If user A has public key $Y_A = 9$, what is A's private key X_A?
c. If user B has public key $Y_B = 3$, what is the secret key K shared with A?

Consider an ElGamal scheme with a common prime $q = 71$ and a primitive root $\alpha = 7$.

a. If B has public key $Y_B = 3$ and A chose the random integer $k = 2$, what is the ciphertext of $M = 30$?

b. If A now chooses a different value of k so that the encoding of $M = 30$ is $C = (59, C_2)$, what is the integer C_2?

References

- Behrouz A. Forouzan and Debdeep Mukhopadhyay – “Cryptography and Network Security”, McGraw Hill, 2nd Edition, 2008.
- William Stallings,“Cryptography And Network Security Principles And Practice”, Fifth Edition, Pearson Education, 2013