[Part 2]
Asymmetric-Key Encipherment

Chapter 10

Asymmetric-Key Cryptography

Forouzan, B.A. Cryptography and Network Security (International Edition). United States: McGraw Hill, 2008.

Objectives

- To distinguish between two cryptosystems: symmetric-key and asymmetric-key;
- To discuss the RSA cryptosystem;
- To introduce the usage of asymmetric-key cryptosystems;
- To introduce the attacks in asymmetric-key cryptosystems;

Contents

Contents

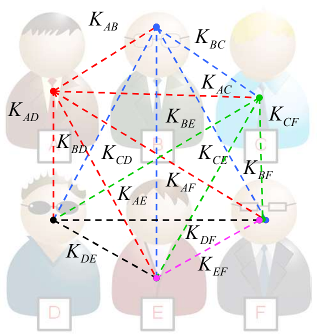
- *Symmetric-* and *asymmetric-key* cryptography will exist in parallel and continue to serve the community.
- Asymmetric-key cryptography also known as *public-key* cryptography.
- Asymmetric-key cryptography are complements to the symmetric-key (secret-key).
- The conceptual differences between them are based on how these systems keep a secret (key).

Symmetric-key cryptography:	Asymmetric-key cryptography:
• Based on **sharing** secrecy. • the secret must be shared between two persons.	• Based on **personal** secrecy. • the secret is personal (unshared). • Each person creates and keeps his/her own secret.
• For n people, we need: $n(n-1)/2$ shared secrets.	• For n people, we need: only n shared secrets.
• Based on substitution and permutation of **symbols** (characters or bits).	• Based on applying mathematical functions to **numbers**.

- **Example**: Symmetric-key cryptography; Each one has a secret key (K) and shared with all.

$n = 6$

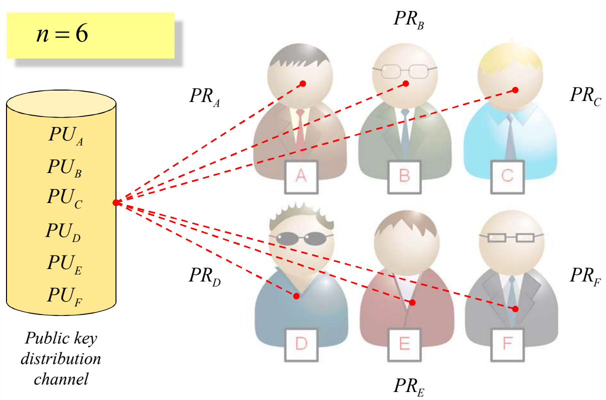
$$= n(n-1)/2$$
$$= 6(6-1)/2$$
$$= 15$$



http://4vector.com/i/free-vector-business-people-icon-02-vector_019614_2.jpg

- **Example**: Asymmetric-key cryptography; Each one has a public key (PU) and a private key (PR); All PUs are shared.

$n = 6$



http://4vector.com/i/free-vector-business-people-icon-02-vector_019614_2.jpg

10.1 Introduction

- Asymmetric-key cryptography can be used for confidentiality (privacy / secrecy), authentication, or both.
- The most widely used asymmetric-key cryptography is RSA.
- The difficulty of attacking RSA is based on the difficulty of finding the prime factors of a composite number.

Stalling, W. Cryptography and Network Security: Principles and Practices (Fourth Edition). United States of America: Pearson, Prentice Hall, 2006. (page 258)

Contents

10.2 Asymmetric-Key Cryptography

Keys

- Asymmetric-key cryptography uses two separate keys: one *private* key and one *public* key.

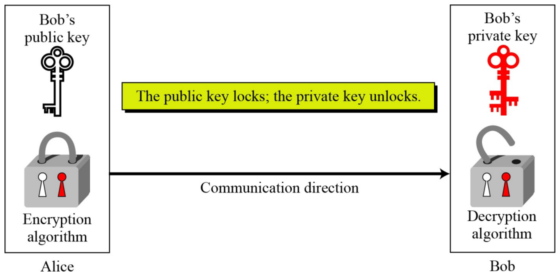


Figure 10.1: Locking and unlocking in asymmetric-key cryptosystem.

10.2 Asymmetric-Key Cryptography

Description:

•Encryption and decryption are thought of as locking and unlocking padlocks with keys.

•The padlock is locked with a public key can be unlock only with the corresponding private key.

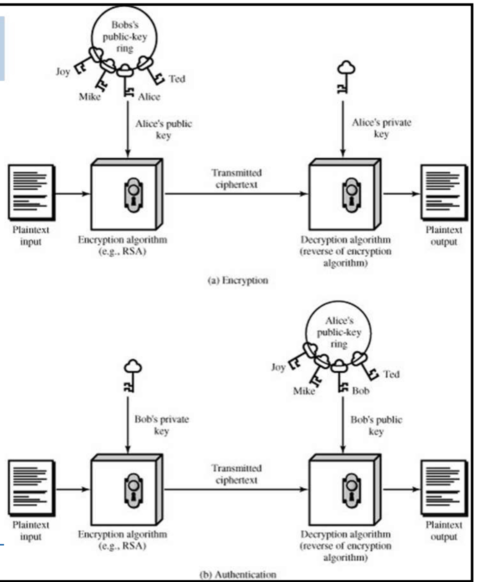
•**Example**:

- Alice locks the padlock with Bob's public key, then only Bob's private key can unlock it.

Chapter 10

The components of asymmetric-key cryptography in general.

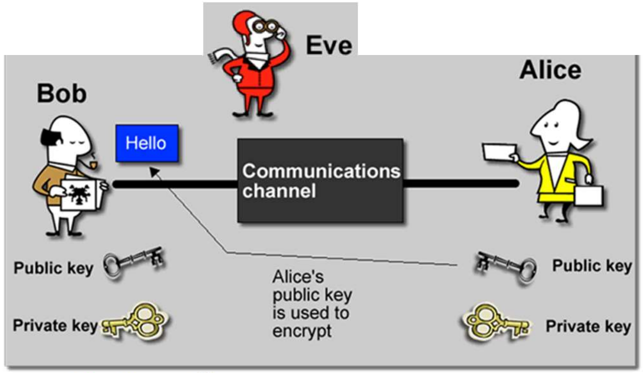
- *Plaintext.*
- *Encryption algorithm.*
- *Public and private keys.*
- *Ciphertext.*
- *Decryption algorithm.*



Stalling, W. Cryptography and Network Security: Principles and Practices (Fourth Edition). United States of America: Pearson, Prentice Hall, 2006. (page 261) http://flylib.com/books/3/190/1/html/2/images/09fig01.jpg

Chapter 10 — 10.2 Asymmetric-Key Cryptography

Example:



$M = D_{K_{private}}(C)$ $C = E_{K_{public}}(M)$

http://cryptographicsoftware.com/wp-content/uploads/2011/08/what-is-Public-Key-Cryptography.gif

1.14

Chapter 10 — 10.2 Asymmetric-Key Cryptography

Example: Alice sent a message M to Bob.

K_{public} $K_{private}$

$C = E_{K_{public}}(M)$ $M = D_{K_{private}}(C)$

1.15

Chapter 9 — 10.2 Asymmetric-Key Cryptography

Exercise 10.1 Bob is sending a plaintext *M* to Alice using an asymmetric-key cryptosystem. Assume that Alice's public key and private key are K_{Alice1} and K_{Alice2} respectively.

a)What is the function of the ciphertext C generated by Bob's encryption E?
b)Show how Alice get the original M from Bob during the decryption D.

Solution 10.1:
a) $C = E_{K_{Alice1}}(M)$
b) $M = D_{K_{Alice2}}(C)$

1.16

General Idea

- Unlike symmetric-key cryptography, there are distinctive keys in asymmetric-key cryptography: a *private key* and a *public key*.

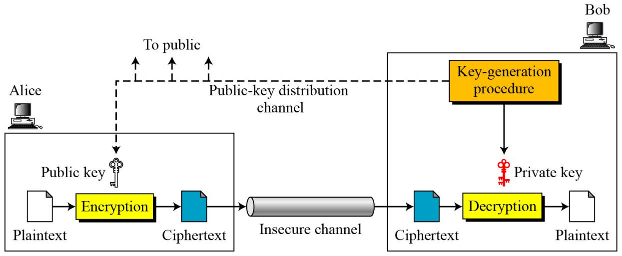


Figure10.2: General idea of asymmetric-key cryptosystem.

Recall

The **secret key** : a string of symbols

The **secret key** used in symmetric-key cryptography is different from the nature of the **private key** used in asymmetric-key cryptography.

The **private key** : a number or a set of numbers

A secret key is not exchange-able with a private key because they are different types of secret !

(Figure 10.2 shows several important facts)

Important **fact (1)**:

- The burden of providing security is mostly on the receiver (Bob). Bob needs to:
  - create two keys: one private and one public.
  - distribute the public key to the community through a public-key distribution channel.
- Although the channel does not required to provide secrecy, it must provide *authentication* and *integrity*.

- Attacker should not be able to advertise his/her public key to the community pretending that it is Bob's public key.

Important **fact (2)**:

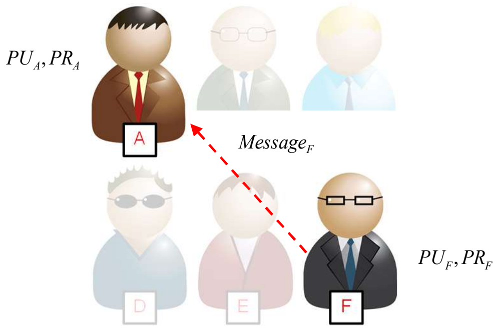
•Bob and Alice cannot use the same set of keys for two-way communication.

•Each entity in the community should create its own private and public keys.

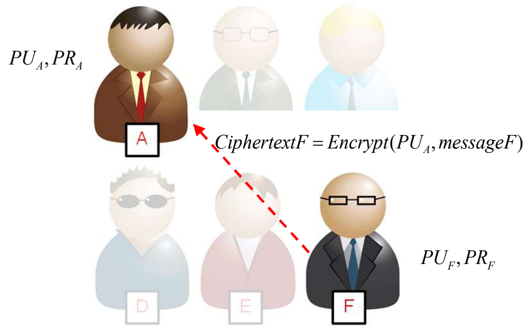
•Figure 10.2 show how Alice can use Bob's public key to send encrypted message to Bob.

•If Bob wants to reply, Alice needs to establish her own private and public keys.

- **Example**: *F* wants to send a message to *A*; *F* will use *A*'s public key.



- **Example**: *F* wants to send a message to *A*; *F* will use *A*'s public key.

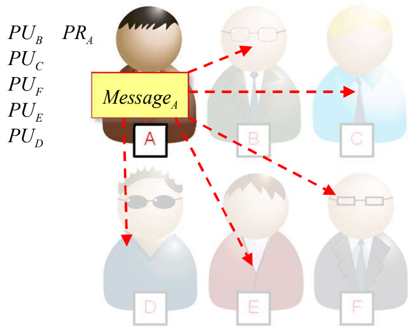


Important **fact (3)**:

- Bob needs only one private key to receive all correspondence from anyone in the community.
- However, Alice needs n public keys to communicate with n people in the community, one public key for each person.

- Alice needs a ring of public keys.

- **Example**: *A* only needs his private key to read any message from *B, C, D, E* and *F*. But *A* needs all public key of them to send message.



Plaintext / Ciphertext

- In asymmetric-key cryptography, the plaintext and ciphertext are treated as integers.

- The message must be *encoded* as an integer before **encryption**.

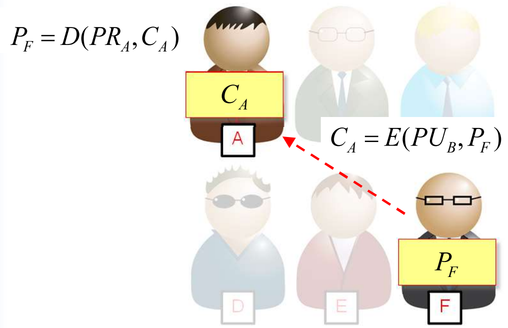
$$C = f(K_{public}, P)$$

- The integer must be *decoded* into the message after **decryption**.

$$P = g(K_{private}, C)$$

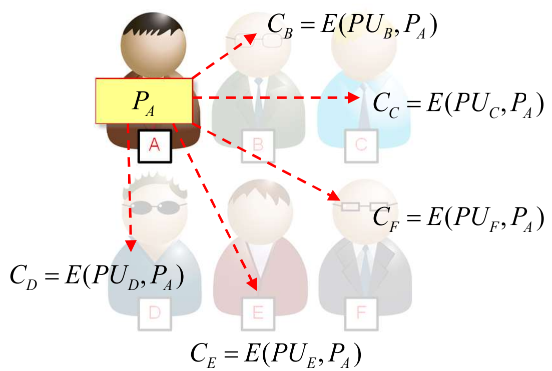
- The encryption function f is used only for encryption;
- The decryption function g is used only for decryption.

- **Example**: A only needs his private key to decrypt message received from F.



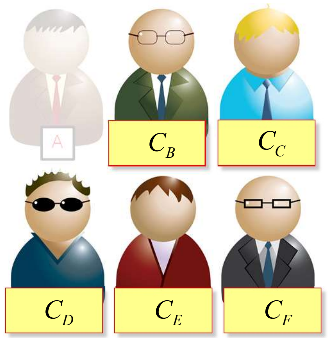
http://4vector.com/i/free-vector-business-people-icon-02-vector_019614_2.jpg

- **Example**: A sends message to all by encrypting it using corresponding public key of them.



http://4vector.com/i/free-vector-business-people-icon-02-vector_019614_2.jpg

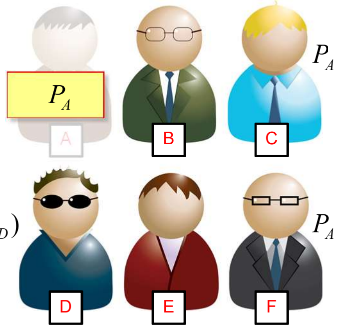
Exercise 10.(Write the decryption function to decrypt the message C from A for each user B, C, D, E and F.



http://4vector.com/i/free-vector-business-people-icon-02-vector_019614_2.jpg

Solution 10.0:

$P_A = D(PR_B, C_B)$



$P_A = D(PR_C, C_C)$

$P_A = D(PR_D, C_D)$

$P_A = D(PR_F, C_F)$

$P_A = D(PR_E, C_E)$

http://4vector.com/i/free-vector-business-people-icon-02-vector_019614_2.jpg

- Asymmetric-key cryptography is normally used to encrypt or decrypt small pieces of information, such as the cipher key for a symmetric-key cryptography.
- In other words, asymmetric-key cryptography normally is used for **ancillary goals** instead of message encipherment that play a very important role in cryptography today.

Need for Both

There is a very important fact that is sometimes misunderstood:
The advent of asymmetric-key cryptography DOES NOT ELIMINATE the need for symmetric-key cryptography.

Reasons:

1) Asymmetric-key cryptography is much slower than symmetric-key cryptography because it uses mathematical functions for encipherment.
2) Asymmetric-key cryptography is still needed for *authentication*, *digital signatures*, and *secret-key exchanges*.

Trapdoor One-Way Function

- The main idea behind asymmetric-key cryptography is the concept of the *trapdoor one-way function* $f : x \to y$, *that is*
  - f is one-to-one.
  - f if a public.
  - One-Way Function (OWF)

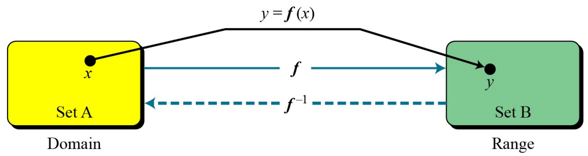


Figure10.3: A function as rule mapping a domain to a range.

- **Function :**

$$y = f(x)$$

A rule that associates (maps) one element in set A, called the *domain*, to one element in set B, called the *range*.

- **Invertible Function :**

$$x = f^{-1}(y)$$

A function that associates each element in the *range* with exactly one element in the *domain*.

One-Way Function (OWF)

A function that satisfies the following two properties :

1) f is easy to compute.

   Given x, $y = f(x)$ can be easily computed.

2) f^{-1} is difficult to compute.

   Given y, it is computationally infeasible to calculate $x = f^{-1}(y)$

Trapdoor One-Way Function (TOWF)

A one-way function (OWF) with a third properties :

3) Given y, and a trapdoor (secret), x can be easily computed.

Example 10.

When n is large, $n = p \times q$ is a one-way function (OWF).

•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.

Example 10.

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

- Given x, k, and n, it is easy to calculate y;
- Given y, k, and n, it is very difficult to calculate x;
- This is the discrete logarithm problem;
- However, if we know the trapdoor, k' such that $k \times k' = 1 \bmod \phi(n)$, we can use $x = y^{k'} \bmod n$ to find x.

Example 10. For $x = 6$, $a = 9$, and $p = 11$, we compute

$$y \equiv x^a \equiv x((x^2)^2)^2 \bmod p$$

with 4 multiplications:

$$y = 6((6^2)^2)^2 \bmod 11 = 6((36)^2)^2 \bmod 11$$
$$= 6((3)^2)^2 \bmod 11 = 6(9)^2 \bmod 11$$
$$= 6(81) \bmod 11 = 6(4) \bmod 11$$
$$= 24 \bmod 11 = 2$$

However, finding an a such that $6^a \equiv 2 \bmod 11$ is hard.

We need to try all possibilities (from 1 to $p-1$) to obtain such a.

Security of Asymmetric-key Cryptography

- Similar to symmetric-key cryptography schemes, the brute force exhaustive search attack is always theoretically possible but keys used are too large (> 512 bits).
- Keys used must be large enough to make brute force attack impractical, but small enough for practical encipherment that requires the use of very large numbers.
- However, the enciperment process is slow compared to symmetric-key cryprography schemes.

Requirements of Asymmetric-key Cryptography

- Computationally easy :
  - to generate the key pairs;
  - for the sender to encrypt;
  - for the receiver to decrypt;
- Computationally infeasible for an opponent,
  - knowing the public key to determine the private key;
  - knowing the public key and a ciphertext, to recover the original message.

Services Provided

Privacy / Confidentiality :

•sender encrypts message with receover's public key;

Authentication (Digital Signature) :

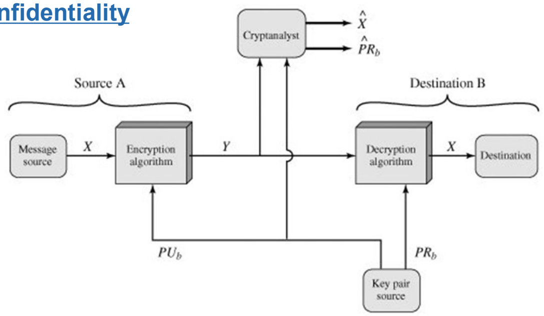
- sender creates signature by encrypting the message with his/her private key;

Integrity

Key Exchange :

To exchange a session key between two entities.

Confidentiality

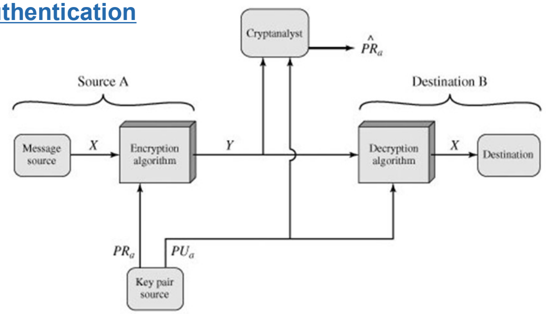


$$Y = E(PU_b, X)$$

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

Figure10.4: Asymmetric-key cryptosystem: secrecy / privacy / confidentiality.

Authentication

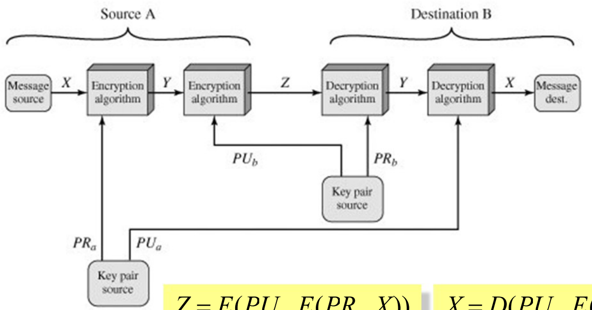


$$Y = E(PR_a, X)$$

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

Figure10.5: Asymmetric-key cryptosystem: authentication.

Authentication and Confidentiality



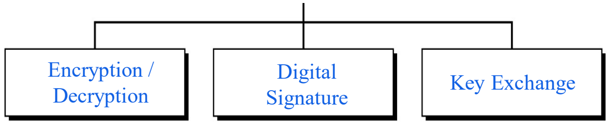
$$Z = E(PU_b, E(PR_a, X))$$

$$X = D(PU_a, E(PR_b, Z))$$

Figure10.6: Asymmetric-key cryptosystem: authentication and confidentiality.

Applications

- Asymmetric-key cryptography are characterized by the use of a cryptographic algorithm with the two keys.
- 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, asymmetric-key cryptography can be classified the use into **three categories**:



10.2 Asymmetric-Key Cryptography

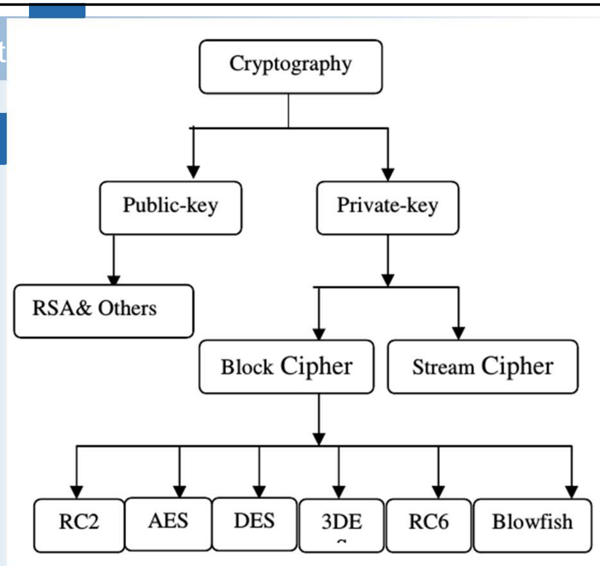
- Some algorithms are suitable for all three applications, whereas others can be used only for one or two of these applications.

Table: Applications for asymmetric-key cryptography.

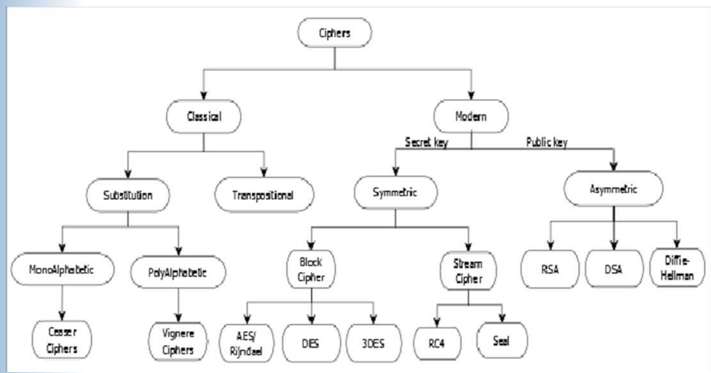
Algorithm	Encryption/De-cryption	Digital Signature	Key Exchange
RSA	Yes	Yes	Yes
Elliptic Curve	Yes	Yes	Yes
Diffie-Hellman	No	No	Yes
DSS (Digital Signature Standard)	No	Yes	No

Stalling, W. Cryptography and Network Security: Principles and Practices (Fourth Edition). United States of America: Pearson, Prentice Hall, 2006. (page 266) http://flylib.com/books/en/3.190.1.83/1/

Contents



Contents



Contents

Introduction

- The most common public-key algorithm is the RSA cryptosystem, named for its inventors (Rivest, Shamir, and Adleman).
- RSA uses two exponents: e (Public) and d (Private)
- Suppose P is the plaintext and C is the ciphertext;
- Alice uses $C = P^e \bmod n$ to create ciphertext from plaintext;
- 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 (will discuss later).

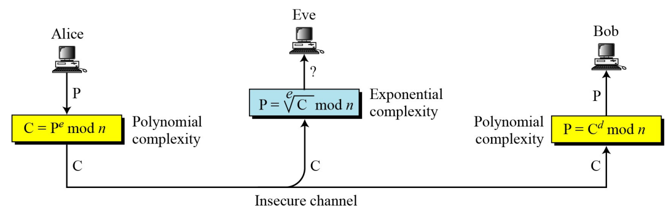


Figure10.4: Complexity of operations in RSA.

- Based on number theory operations and the difficulty to find prime factors for a large number, $n = pq$, where p and q are primes.

[Back]

- Encryption and decryption use modular exponentiation.
- Modular logarithm is as hard as factoring the modulus, for which there is no polynomial algorithm yet.
- Figure 10.4 show the idea:
  - 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.

Summary of RSA idea (Figure 10.4):

- 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 anymore.

Procedure

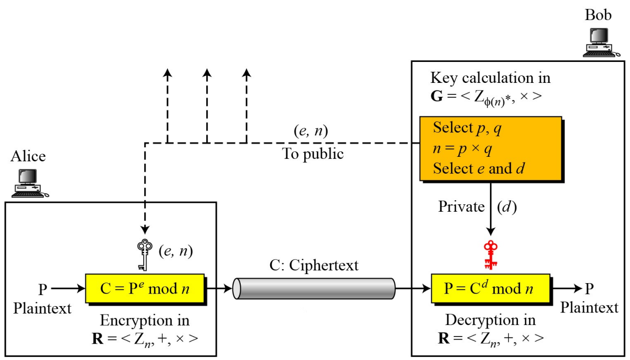


Figure10.5: Encryption, decryption, and key generation in RSA.

RSA Structure

- RSA uses two algebraic structures:

Encryption/Decryption Ring :

$$R = < Z_n, +, \times >$$

- Encrypt/decrypt using the commutative ring **R** with two arithmetic operations (+, x).
- This ring is public since modulus n is public.

Key-Generation Group :

$$G = < Z_{\phi(n)}*, \times >$$

- RSA uses multiplicative group G for key generation.
- Hidden from public because modulus $\phi(n)$ is hidden from public.

Key Generation

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

n must be at least 1024 bits;

Algorithm : *RSA key generation.*

```
RSA_Key_Generation
{
    Select two large primes p and q such that p ≠ q.
    n ← p × q
    ϕ(n) ← (p − 1) × (q − 1)
    Select e such that 1 < e < ϕ(n) and e is coprime to ϕ(n)
    d ← e^−1 mod ϕ(n)                  // d is inverse of e modulo ϕ(n)
    Public_key ← (e, n)                // To be announced publicly
    Private_key ← d                    // To be kept secret
    return Public_key and Private_key
}
```


Encryption and Decryption

Algorithm : *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
}
```

Algorithm : *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
}
```

Example 10.4 Given $p = 5$ and $q = 3$.

- Calculates $n = p \times q = 5 \times 3 = 15$.
- The value of $\phi(n) = (p - 1)(q - 1) = (5 - 1)(3 - 1) = 8$.
- Choose integer e, → $\gcd(\phi(n), e) = 1$ and $1 < e < \phi(n)$.

  Say $e = 5$
- Calculates $d = e^{-1} \bmod \phi(n) = 5^{-1} \bmod 8$.

  Use Euler's theorem to find the inverse:

  $$a^{-1} = a^{\phi(n)-1} \bmod n$$

  $$d = 5^{\phi(8)-1} \bmod 8$$

$\phi(p^e) = p^e - p^{e-1}$

$\phi(8) \rightarrow \phi(2^3)$

$= 2^3 - 2^{3-1}$

$= 8 - 4 = 4$

$$d = 5^{\phi(8)-1} \bmod 8$$

$$d = 5^{4-1} \bmod 8$$

$$= 5^3 \bmod 8$$

$$d = 5$$

- Public Key, K_{pu} = { e, n } = { 5, 15 }
- Public Key, K_{pr} = { d, n } = { 5, 15 }

Example 10.4

- Public Key, K_{pu} = { e, n } = { 5, 15 }
- Public Key, K_{pr} = { d, n } = { 5, 15 }

Given a message $M = 4$.

Message encryption :

$$C = M^e \bmod n = 4^5 \bmod 15 = 4$$

Message decryption :

$$M = C^d \bmod n = 4^5 \bmod 15 = 4$$

Exercise 10. a) Find the value of $\phi(15)$.

b) Using the Euler's theorem, proof that $4^5 \bmod 15 = 4$

Solution 10.1 (Second version of Euler's theorem)

Let $k = 1$;

Exercise 10. Given $p = 11$ and $q = 13$. Assume that $e = 11$ is used to encrypt a message $M = 7$,

a) calculate the value of d, and the ciphertext C.

b) Show the decryption process to get the original message.

Solution 10.1:

Example 10.5 Bob chooses 7 and 11 as p and q.

- Calculates $n = p \times q = 7 \times 11 = 77$.
- The value of $\phi(n) = (7 - 1)(11 - 1) = 60$.
- Now Bob chooses two exponents, e and d, from $Z_{60}*$.
- If Bob chooses $e = 13$, then $d = 37$.

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

Example 10.5 Now imagine that Alice wants to send the plaintext 5 to Bob.

$e = 13$

$d = 37$

$n = 77$

$P = 5$

$C = 26$

- Alice uses the public exponent 13 to encrypt 5:

$$C = P^e \bmod n = 5^{13} \bmod 77 = 26$$

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

$$P = C^d \bmod n = 26^{37} \bmod 77 = 5$$

Exercise 10. From Example 10.5a, proof that $d = 37$.

(Multiplicative inverses from Euler's theorem)

$$a^{-1} \bmod n = a^{\phi(n)-1} \bmod n$$

Exercise 10.2 From Example 10.5a, proof that $d = 37$.

(Multiplicative inverses from Euler's theorem)

$$a^{-1} \bmod n = a^{\phi(n)-1} \bmod n$$

Solution 10.2:

$$d = e^{-1} \bmod \phi(n)$$
$$= 13^{-1} \bmod \phi(77)$$
$$= 13^{\phi(60)-1} \bmod 60$$
$$= 13$$

Exercise 10.3 From Example 10.5b, proof that :

a) $5^{13} \bmod 77 = 26$

b) $26^{37} \bmod 77 = 5$

Solution 10.3:

Example 10.6 Azizah creates a pair of keys for herself. She chooses 397 and 401 as p and q.

- Calculates $n = p \times q = 397 \times 401 = 159197$.
- The value of $\phi(n) = (397 - 1)(401 - 1) = 158400$.
- Now Azizah chooses two exponents, e and d, from $Z_{\phi(159197)}*$.
- If Azizah chooses $e = 343$, then $d = 12007$.

Example 10.6 Suppose Mubassyir wants to send a message "NO" to Azizah.

Code	Character
00	A
01	B
02	C
03	D
04	E
05	F
06	G
07	H
08	I
09	J
10	K
11	L
12	M
13	N
14	O
15	P
16	Q
17	R
18	S
19	T
20	U
21	V
22	W
23	X
24	Y
25	Z

- 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.

- Figure 10.7 shows the process.

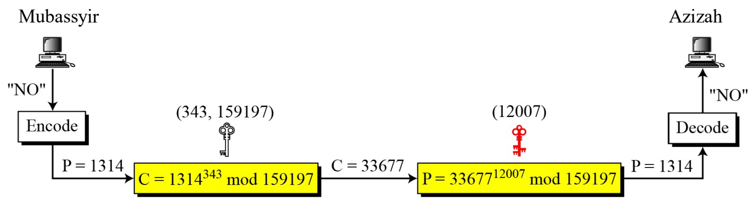


Figure10.7: Encryption and decryption in Example 10.5b.

Exercise 10.3 From Example 10.5a, proof that $d = 12007$.

(Multiplicative inverses from Euler's theorem)

$$a^{-1} \bmod n = a^{\phi(n)-1} \bmod n$$

Solution 10.3: $d = e^{-1} \bmod \phi(n)$

$$= 13^{-1} \bmod \phi(77)$$
$$= 13^{\phi(60)-1} \bmod 60$$
$$= 13$$

Exercise 10.4 From Example 10.5b, proof that :

a) $1314^{343} \bmod 159197 = 33677$

b) $33677^{12007} \bmod 159197 = 1314$

Solution 10.4:

Exercise 10.5 From Example 10.5b, proof that :

a) $1314^{343} \bmod 159197 = 33677$

b) $33677^{12007} \bmod 159197 = 1314$

Solution 10.5:

Example 10.7

Encrypt ‘RENAISSANCE’ using $p = 53$ and $q = 61$.

$n = p * q = 3233$

Say $e = 71$, then $d = 791$

(check the validity of e and d)

Break the message into blocks of 4 digits where A = 00, B = 01, …, Z = 25 (in practice, characters would be represented by their 8 bit ASCII codes)

Thus RE NA IS SA NC E = 1704 1300 0818 1800 1302 0426

The 1st block is encrypted as 1704^{71} mod 3233 = 3106

c = 3106 0100 0931 2691 1984 2927

Example 10.8

$p = 61$, $q = 53$, $pq = 3233$,

$e = 17$ (public exponent), $d = 2753$ (private exponent)

Public key is (pq, e).

Private key is d.

C = encrypt (T) = (T^{17}) mod 3233

T = decrypt(C) = (C^{2753}) mod 3233

Encrypt (123) = (123^{17}) mod 3233 = 337587917446653715596592958817679803 mod 3233 = 855

Decrypt (855)
= 855^{2753} mod 3233

[illegible]
84375 mod 3233 = 123

Real use of RSA

- In general, RSA is not used to encrypt long messages.
- Instead it is used for:
  - Transmitting short secret key / value such as credit card, key for use in symmetric encrypt/decrypt system.
  - Digital signature
  - Authentication such as identifying an entity.
  - Certificate.

Introduction

- No devastating attacks on RSA have been yet discovered.
- Several attacks have been predicted based on the :
  - weak plaintext,
  - weak parameter selections, or
  - inappropriate implementation.
- Figure 10.7 shows the categories of potential attacks.

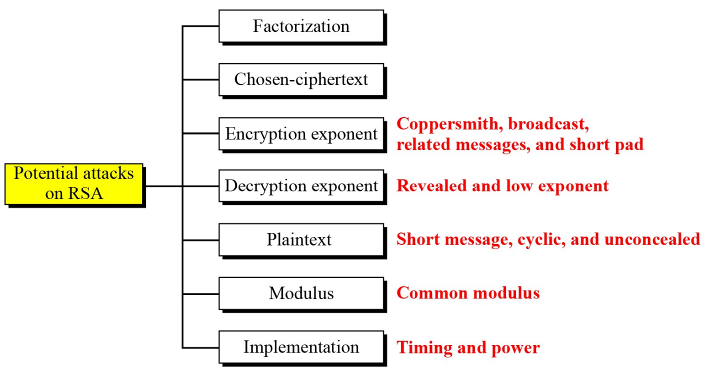


Figure10.4: Taxonomy of potential attacks on RSA.

- The obvious way to do this attacks is to factor the public modulus, n, into its two prime factors, p and q.
  - From p, q and e, the attacker can easily get d.
- The hard part is factoring n:
  - Security on RSA depends on factoring being difficult.
  - In fact, the task of recovering the private key is equivalent to the task of factoring the modulus.
  - It should be noted that the hardware improvements alone will not weaken the RSA, as long as appropriate key length are used.

- Another way to break the RSA is to find a technique to compute eth roots mod n.
  - Since $C = M^e \bmod n$, the eth root of C mod n is the message m.
  - This would allow someone to recover encrypted messages and forge signatures even without knowing the private key.
  - No general methods are currently known that attempt to break RSA in this way.
  - However, in special cases where multiple related messages are encrypted with the same small exponent, it may be possible to recover the messages.

- There are no attack against the algorithm but instead the protocol.
  - Attacker sees a ciphertext and guesses that the message might be.

Contents

- There are another asymmetric-key or public-key cryptosystems:
  - Rabin cryptosystem.
  - ElGamal cryptosystem.
  - Elliptic Curve cryptosystem (ECC).

Rabin Cryptosystem

- The Rabin cryptosystem can be thought of as an RSA cryptosystem in which the value of e and d are fixed.
- Based on quadratic congruence.
- The encryption is $C \equiv P^2 \pmod{n}$ and the decryption is $P \equiv C^{1/2} \pmod{n}$.
- The Rabin cryptosystem is not deterministic: Decryption creates four plaintexts.

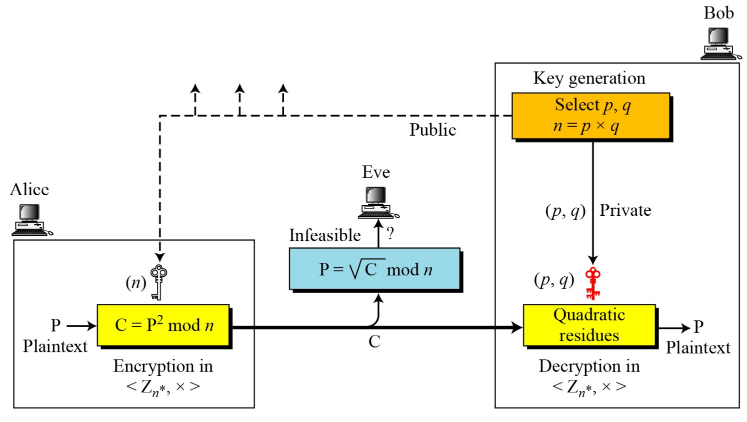


Figure10.8: Encryption, decryption, and key generation in the Rabin cryptosystem.

ElGamal Cryptosystem

- Besides RSA and Rabin, another public-key cryptosystem is ElGamal.
- ElGamal is based on the discrete logarithm problem.
- For the ElGamal cryptosystem, p must be at least 300 digits and r must be new for each encipherment.
- The bit-operation complexity of encryption or decryption in ElGamal cryptosystem is polynomial.

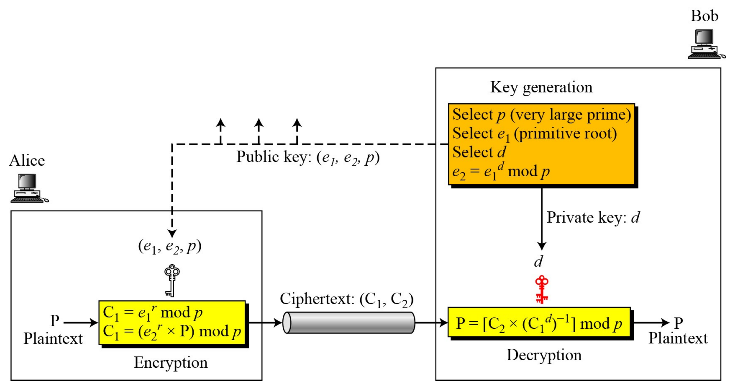


Figure10.9: Encryption, decryption, and key generation in the ElGamal cryptosystem.

Elliptic Curve Cryptosystem

- Although RSA and ElGamal are secure asymmetric-key cryptosystems, their security comes with a price, their large keys.
- Researchers have looked for alternatives that give the same level of security with smaller key sizes.
- One of these promising alternatives is the elliptic curve cryptosystem (ECC).

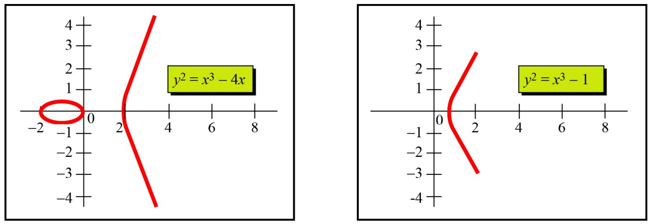
- Based on theory of elliptic curves.
- The general equation for an elliptic curve is

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

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

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

- The security of ECC depends on the difficulty of solving the elliptic curve logarithm problem



a. Three real roots

b. One real and two imaginary roots

Figure10.10: Two examples elliptic curves over a real field.

Contents

- There are two ways to achieve secrecy: symmetric- and asymmetric-key cryptography that complement each other.
- The conceptual differences between the two systems are basically based on how they keep a secret.

	Symmetric-Key	Asymmetric-Key
Keys	Single key: secret-key	Two keys: public-key, private-key.
Secret	Shared between two entities.	Unshared.
Implementation	Based on substitution and permutation of symbols.	Based on applying mathematical functions to numbers.

- In asymmetric-key cryptography:
  - ❑ Encryption and decryption can be thought of as locking and unlocking padlocks with keys.
    - Locked with a public key;
    - Unlock only with the corresponding private key.
  - ❑ The burden of providing security is mostly in the receiver, who needs to:
    - create two keys (public and private key).
    - Distribute the public key to the community via a public-key distribution channel.

- The main idea behind symmetric-key cryptography is the concept of the trapdoor one-way function (TOWF), which is a function such f is easy to compute, but f^{-1} is computationally infeasible unless a trapdoor is used.
- The most common public-key algorithm is the RSA cryptosystems.
- No devastating attacks have yet been discovered on RSA.
- Another asymmetric-key cryptography algorithms are Rabin cryptosystem, ElGamal cryptosystem, and Elliptic Curve cryptosystem (ECC).

Exercise 10.1 In RSA:

a) Given $n = 221$ and $e = 5$, find d.

b) Given $n = 3937$ and $e = 17$, find d.

c) Given $p = 19$, $q = 23$ and $e = 3$, find n, $\phi(n)$ and d.

Forouzan,B.A. Cryptography and Network Security (International Edition). Singapore: McGraw-Hill, 2008. (page 334)

Chapter 10 Exercises

Exercise 10. Perform encryption and decryption using the RSA algorithm for the following:

a) $p = 3, q = 11, e = 7$, and $M = 5$.

b) $p = 5, q = 11, e = 3$, and $M = 9$.

c) $p = 7, q = 11, e = 17$, and $M = 8$.

d) $p = 11, q = 13, e = 11$, and $M = 7$.

e) $p = 17, q = 31, e = 7$, and $M = 2$.

Stalling, W. Cryptography and Network Security: Principles and Practices (Fourth Edition). United States of America: Pearson, Prentice Hall, 2006. (page 282)

Chapter 10 Exercises

Exercise 10.4 To understand the security of the RSA algorithm, find d if you know that $e = 17$ and $n = 187$.

Forouzan, B.A. Cryptography and Network Security (International Edition). Singapore: McGraw-Hill, 2008. (page 334)

Chapter 10 Exercises

Exercise 10. In a public-key system using RSA algorithm, you intercept the ciphertext $C = 10$ sent to a user whose public key is $(n, e) = (35, 5)$. What is the plaintext M?

Stalling, W. Cryptography and Network Security: Principles and Practices (Fourth Edition). United States of America: Pearson, Prentice Hall, 2006. (page 282)