Chapter 3
Modern Encryption Techniques

3.1 Inroduction

The objective of this part is to illustrate the principle of modern encryption techniques.
We focus on the most widely used encryption algorithm: the
"Data encryption standard (DES)"

3.1 Simplified Data Encryption Standard (S-DES)

S-DES is an educational rather than a secure encryption algorithm. It has similar structure to Data encryption standard DES but with much smaller parameters. It was developed by professor Edward Schaefer of Santa Clara University.

3.2 S-DES Structure

As shown below the overall structure of DES:
The S-DES **encryption** algorithm takes an
(1) 8 bits block of text (example 10111101),
(2) 10-bit keys as input and
(3) Produces an 8-bit block of cipher.

Also the S-DES decryption algorithm takes an 8-bit block of ciphertext and the same 10-bits key used to produce the original 8-bits plaintext.

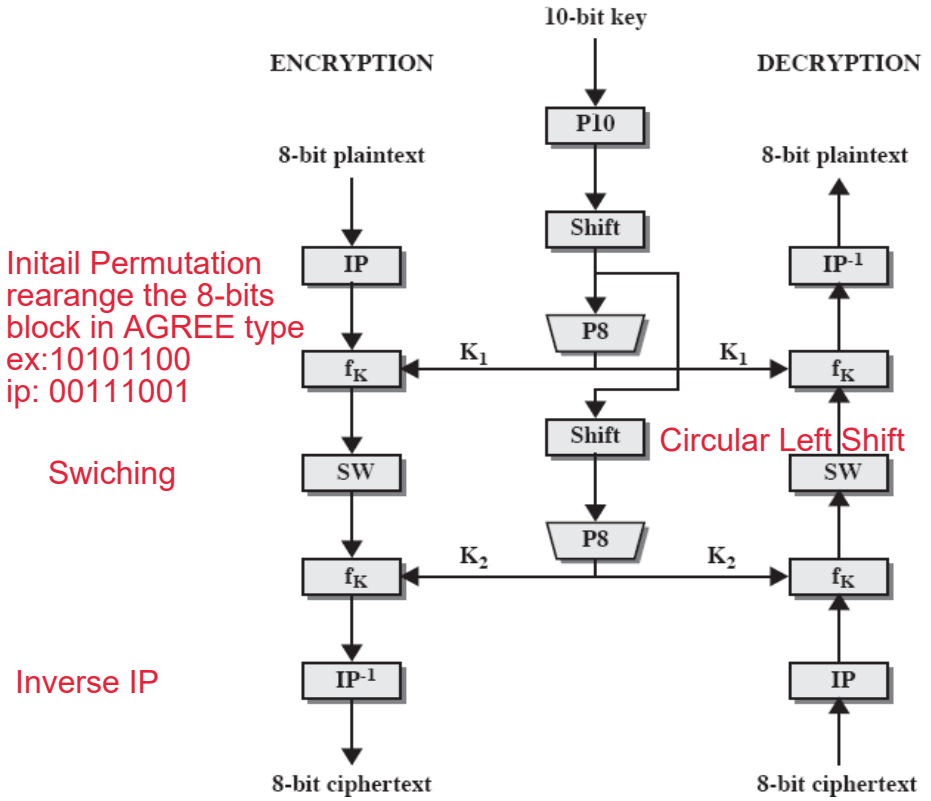


Figure 3-1. S-DES Scheme

1. Encryption:

The encryption algorithm involves five functions:

a. An initial permutation (IP);

b. A complex functions labeled f_k which involves both permutation and substitution operations and depends on a key input a simple permutation function

c. Switches (SW) the two halves of the data;

d. The function f_k again, and

e. Finally a permutation function that is the inverse of initial permutation (IP^{-1}).

We can express the encryption algorithm as a composition of functions:

$$IP^{-1}.F_{K2}.SW.F_{K1}.IP$$

Which can be written as:

$$\text{Ciphertext} = IP^{-1}(F_{K2}(SW(F_{K1}(IP(\text{Plaintext})))))$$

Where: $K_1 = P_8$ (shift(P_{10}(key))) and $K_2 = P_8$ (shift(shift(P_{10}(key)))

2- Decryption: 1010110010 >> 0101100101>> 1011001010
1011>>0111>> 1110>>1101>>

Decryption is also shown in the above figure and essentially it is the reverse of encryption, i.e.

Plaintext = IP^{-1} (F_{K1}(SW(F_{K2} (IP(Ciphertext))))

Block Cipher Principle:

- ***Stream Cipher***: it encrypts data as stream of characters or bytes, e.g. Vigenere Cipher.

- ***Block cipher***: a block of data is encrypted together, block size of 64 or 128 bits is used. E.g. S-DES, DES, 3DES, etc.

More International Symmetric Algorithms

This part include the most important symmetric block cipher in current use. The cipher were selected based on a number of criteria:
1- They are popular in internet applications.
2- They illustrate modern symmetric block cipher techniques that have been developed.
3- Considerable cryptographic strength.

These algorithms are:

DES, Blowfish, RC5, RC2 CAST and IDEA.

Chapter 4
Public Key Cryptography

4.1 Introduction

There were two problems associated with symmetric system;

1. Key distribution; This is only achieved by one the following methods

a- Already distributed shared key.

b- Use of key distribution center.

(Both methods are liable to be compromised)

2. Digital signature problem; Electronic documents would need the equivalent of signatures used on paper documents.

Diffie and Hellman have achieved new technique in 1976 that addresses these problems and was completely different from all previous methods of ciphering, it is called public-key cryptosystem.

4.2 Principle of Public key cryptosystems

Public key algorithms rely on two keys, one key for encryption and another key for decryption. These keys are related and the algorithms have the following important characteristics:

1- It is computationally infeasible to determine the decryption key given only knowledge of the algorithm and the encryption key.

2- Either of the two related keys can be used for encryption with the other used for decryption.

Example of public-key system is RSA (Rivest, Shamir and Adleman).

Figure 4-1 illustrates the public key encryption process and figure 4-2 illustrates the public key authentication. It consists of six

1010110010

ingredients; i.e.

1. Plaintext.
2. Encryption algorithm. ~~1010110010 >> 010110010~~
3. Public key.
4. Private key.
5. Ciphertext.
6. Decryption algorithm.

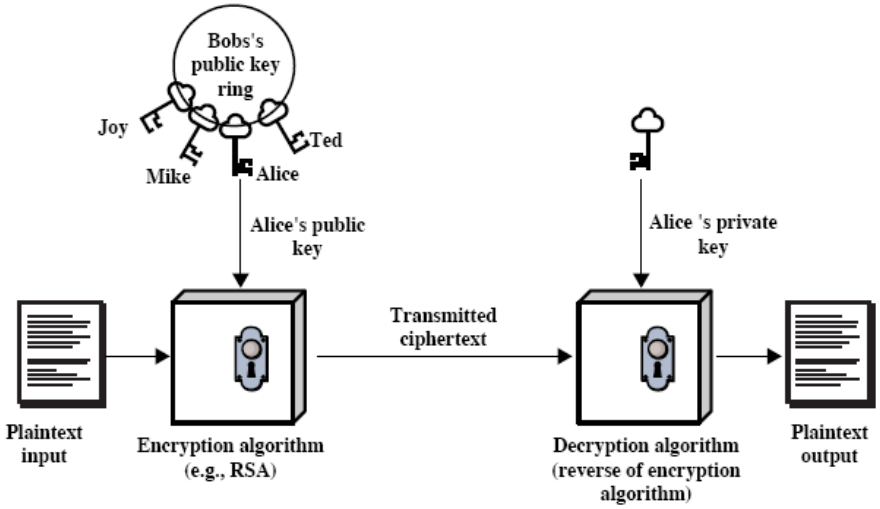


Figure 4-1. Public key encryption

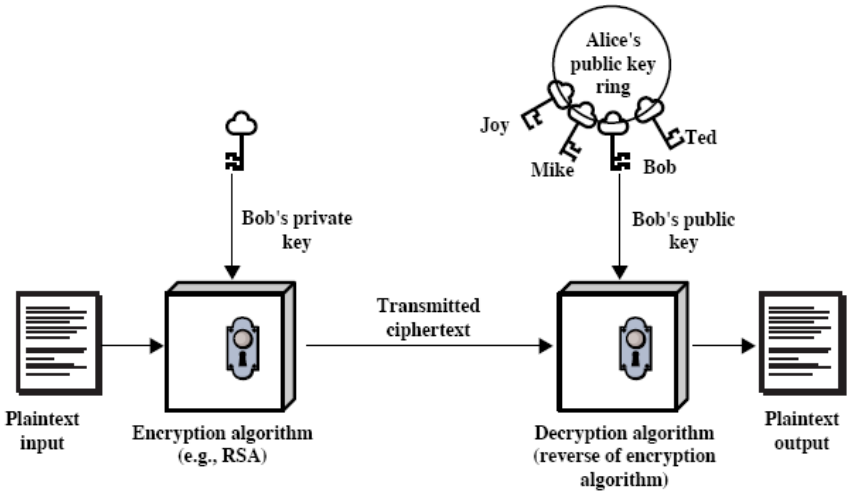


Figure 4-2. Public key Authentication

The essential steps for secrecy shown in figure 4-1 are the following:

1- Each end system (user) in the network generates two keys, one for encryption of message at the sender end and the other for decryption at the receiver.

2- *Each system (user) publishes its encryption key by placing it in a public register or file. This is the public key and the companion key is kept private. The user also keeps the private keys of all other users.*

3- *If A (Bob) wishes to send a message to B (Alice), he encrypts the message using B's public key.*

4- *When B (Alice) receives the message, B decrypts it using her owns private key. No other recipient can decrypt the message because only B knows B's private key*

(Note: No private key distribution, but only public key).

4.3 Symmetric versus public –key Encryption

The following table summarizes some important aspects of Conventional (symmetric) and Public Key (asymmetric) encryption systems.

Conventional Encryption	Public- key Encryption
Needed work: • The same algorithm with the same key is use for encryption and decryption. • The sender and receiver must share the algorithm and key.	Needed work: • 1. One algorithm is used for encryption and decryption with a pair of keys, one for encryption and one for decryption. • 2- The sender and receiver must each have one of the matched pair of keys (not the same one).
Need for Security: ○The key must be kept secret. ○It must be impossible or at least impractical to decipher a message if no other information is available.	Need for Security: • One of the two keys must be kept secret. • 2-It must be impossible or at least impractical to decipher a message if no other information is available. • 3-Knowledge of the algorithm

○Knowledge f the algorithm plus samples of ciphertext must be insufficient to determine the key.	plus one of the keys plus samples of ciphertext must be insufficient to determine the other key.

To discriminate between the two cryptosystems, we will generally refer to the key used in symmetric encryption as a secret key. The two keys used for asymmetric key encryption referred to as public key and private key.

4-4. Essential Elements of public-key encryption

To understand how the system works, consider figure 4-3 below, which is suitable for confidentiality or secrecy.

Let the plaintext message which consists of letters in some finite alphabet at the source A be $X = \{X_1, X_2, \ldots, X_m\}$
Where m is the number of elements of X, (e.g. English language alphabet).

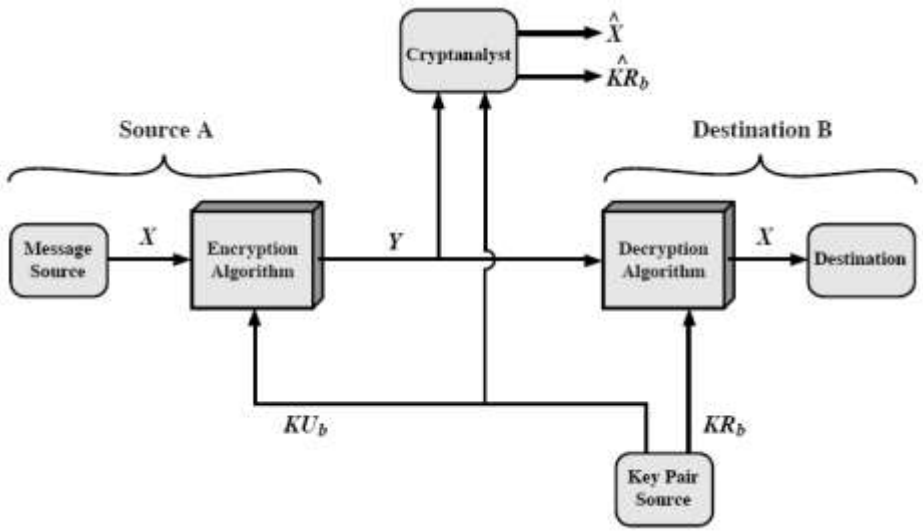


Figure 4-3. Public-key cryptosystem: secrecy

The message is intended to be received by the destination B. The intended receiver B generates related pair of keys; a public key, KU_b, and private key KR_b. Therefore, key KR_b is known only to the receiver B, whereas key KU_b is publicly available and therefore

accessible by the source A.

1- With the message X and the encryption key KU_b as input, the ender A forms the ciphertext: $Y = \{Y_1, Y_2, \ldots, Y_m\}$ by $Y = E_{KUb}(X)$
2- The receiver B, in possession of the matching private key KR_b, is able to invert the transformation: $X = D_{KRb}(Y)$

We mentioned earlier that either of the two related key can be used for encryption, with the other being used for decryption.

> NOTES: This model for secrecy may be attacked by opponent who either;
> 1- Interested in the current message and tries to recover a plaintext X^, or
> 2- Interested in future messages, and tries to recover KR_b by generating an estimate KR_b^.

Figure 4-3, which resembles the action of figure 4-1, has shown the use of public-key cryptosystem for secrecy of message. However, it does not provide for authentication. For authentication purpose, illustrated in figure 4-2, the arrangement of figure 4-4 can be used.

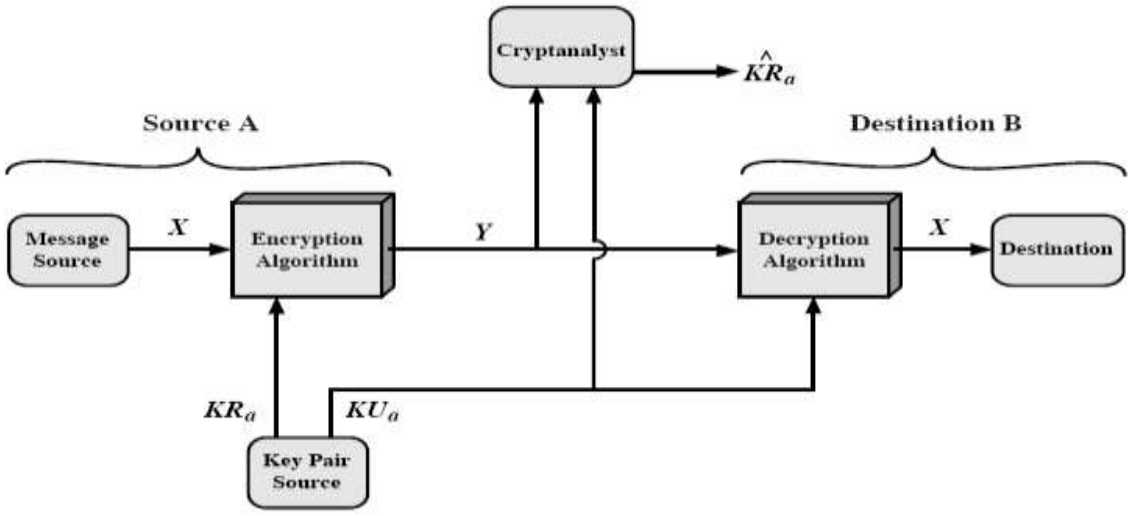


Figure 4-4. Public-key cryptosystem: authentication

In this case, A prepares a message to B, but he/she encrypts it using his own private-key before transmitting it to B, i.e.

$$Y = E_{KRa}(X)$$

Then B can decrypt it using A's public key.

$$X = D_{KUa}(Y)$$

Because the message was decrypted with A's public key, therefore, only A could have encrypted it. This means the entire encrypted message serves as a "**digital signature**".

It is obvious that the encryption process used when **digital signature** is implemented means that there will be no confidentiality because the public key of the sender is available and can be used by anybody to encrypt the message.

However, if **authenticity of sender** and **secrecy (confidentiality) of a message** are required, this can be achieved by double use of public-key scheme, see figure 4-5.
Here, the message X is first encrypted using the sender's private key (signing), and then the resulted cryptogram Y is encrypted using the intended recipient public key. The signed and encrypted message Z is then sent over the unsecure channel to the intended destination.

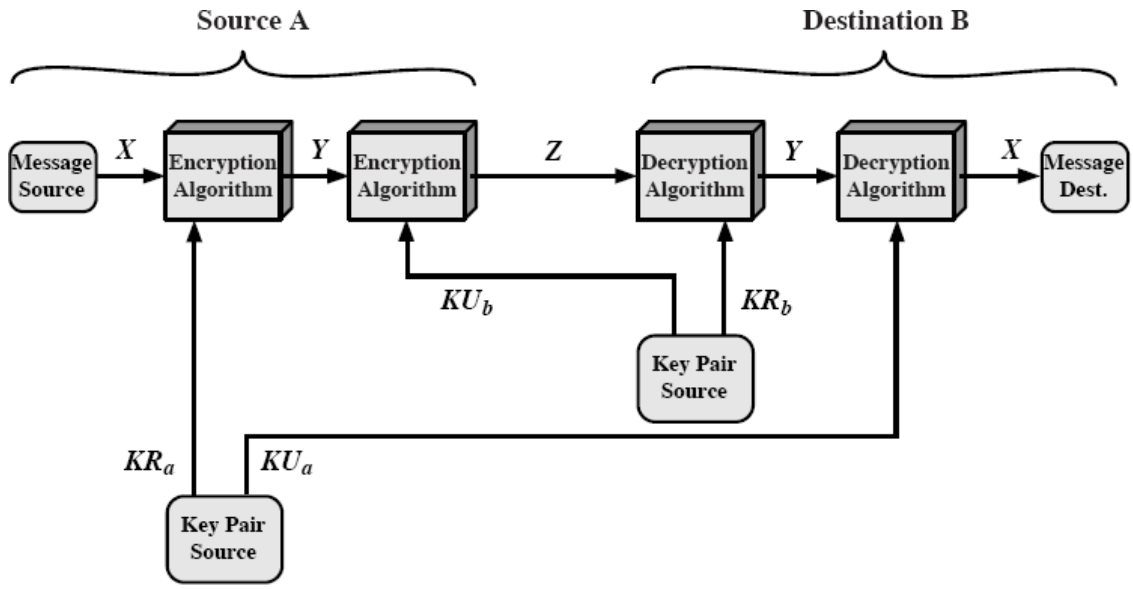


Figure 4-5 Public-key cryptosystem: secrecy and authentication

$$Z = E_{KUb}[E_{KRa}(X)]$$

Upon reception at the destination, the recipient uses his private key to decrypt the received message Z, and then use the sender's public

key to check the authenticity of the message, i.e.

$$X = D_{KUa} [D_{KRb} (Z)]$$

This process provides digital signature of A and encryption with B's key for secrecy at the sender A side and then decryption at the B's side first then checking the senders signature.

4.5 Applications for public- key cryptosystems

In broad terms, we can classify the use of public-key cryptosystems into three categories:

- **Encryption/decryption:** The sender encrypts a message with the recipient's public key.
- **Digital signature:** the sender "signs" a message with its private key. Signing is achieved by a cryptographic algorithm applied to the message or to a small block of data that is a function of the message.
- **Key exchange:** two sides cooperate to exchange a session key.

Some algorithms are suitable for all three applications, where others can be used only for one or two of these applications, as show in the following table.

Algorithm	Encryption/ Decryption	Digital signature	Key exchange
RSA	Yes	Yes	Yes
Elliptic Curve	Yes	Yes	Yes
Diffie-Hellman	No	No	Yes
DSS	No	Yes	No

4.6 The RSA Algorithm

The most widely used public-key cryptosystem scheme since year 1978. It is named after three scientists; Ron Rivest, Adi Shamir and Len Adleman (RSA).

- The RSA system is a block cipher in which integer representation for plantext and ciphertext are between **0** and **n-1** for some **n.**

- This system rest upon the computational difficulty involved in factoring very large prim composite integer **n.** [large mean between 100 and 200 bits at the beginning. However, now **n = 1024** bits (or 309 decimal digits) may considered large but more for military applications].

- RSA has proved popular in email communication.

Description of RSA algorithm:

- It uses exponentiation.

- Plaintext encrypted in blocks with binary value **< n.**
  In practice, block size is 2^k, where $2^k < n <= 2^{k+1}$

- If M is the message then ciphertext C is calculated by:
  $C = M^e \bmod n$
  And M can be recovered by:
  $M = C^d \bmod n$
  $= (M^e \bmod n)^d \bmod n = (M^e)^d \bmod n$
  $= M^{ed} \bmod n$

  Where **n** is known to both sender and receiver,
  e known to sender and **d** is known to receiver only.

Thus, this is a public-key encryption/decryption algorithm with public key **KU = {e, n}** and private key **KR = {d, n}.**

Now, we need to find a relationship of the form $M^{ed} = M \bmod n$.

From Euler's theorem,

Given two prime numbers, p & q, and two integers n & m, such that n = pq and 0 < m < n, and an arbitrary integer k, the following relationship holds:

$$m^{k\phi(n)+1} = m^{k(p-1)(q-1)+1} \equiv m \bmod n$$

Where $\phi(n)$ is the Euler tutent function, which is the number of positive integers less than n and relatively prime to n.
It is shown that $\phi(pq) = (p-1)(q-1)$. Thus we can achieve the desired relationship if

$$e\,d = k\,\phi(n) + 1$$

This is equivalent to saying

$$e\,d \equiv 1 \bmod \phi(n)$$
$$d \equiv e^{-1} \bmod \phi(n)$$

that is e and d are multiplicative inverses mod $\phi(n)$. It must be noted that this is only true if and only if e and d are relatively prime to $\phi(n)$. Equivalently $\gcd(\phi(n), d) = 1$.

RSA Design

RSA scheme ingredients are:

- Two large positive prime integers;
  p and q are chosen (private, chosen)
- Their product N=pq is calculated (public, calculated)
- Now a positive integer e is chosen which is prime to (p-1)(q-1) or $\phi(n)$, the Euler's tutent function. (public, chosen)
- Fermat's theorem is applied to calculate another integer d, i.e. $d \equiv e^{-1} \bmod \phi(n)$ (private, calculated)

[Euclid stated :
"If e and $\phi(n)$ satisfy $1 < e < \phi(n)$ and $\gcd(e, \phi(n)) = 1$, then there is a unique integer d, where $1 < d < \phi(n)$ such that $e\,d \equiv 1 \bmod \phi(n)$"].

Hence,

The **public key** consists of {e, n} and the **private key** consists of {d, n}.

In conclusion, RSA technique involves three main operations; Key generation, Encryption and Decryption as summarized in figure 4-6.

Key Generation

Select p, q	p and q both prime, $p \quad q$
Calculate $n = p \times q$	
Calculate $\phi(n) = (p-1)(q-1)$	
Select integer e	$\gcd(\phi(n), e) = 1; \quad 1 < e < \phi(n)$
Calculate d	$d \equiv e^{-1} \bmod \phi(n)$
Public key	$KU = \{e, n\}$
Private key	$KR = \{d, n\}$

Encryption

Plaintext:	$M < n$
Ciphertext:	$C = M^e \pmod{n}$

Decryption

Ciphertext:	C
Plaintext:	$M = C^d \pmod{n}$

Therefore, the message **M** can be encrypted by the sender using the equation $C = M^e \bmod n$, and decrypted by the receiver using the relation $M = C^d \bmod n$.

To prove this; we have $e\,d \equiv 1 \bmod \phi(n)$

Since $C \equiv M^e \bmod n$

But $M \equiv C^d \bmod n$

$\equiv (M^e \bmod n)^d \bmod n \equiv (M^{ed}) \bmod n$

Generally, to encipher a message X, it is first divided into blocks X_1, X_2, X_3, . . . , X_m at the sender end, and then each block X_i is encrypted by:

$C_i = X_i^e \bmod n$, C_i is ciphertext block.

At the receiver end, it is deciphered by:

$X_i = C_i^e \bmod n$

4.7 Simple RSA Implementation examples:

Example 1: Select two prime integers; $p = 5$ *and* $q = 7$

Then calculate $n = p\,q = 5 \times 7 = 35$

Now $\phi(35) = (5-1)(7-1) = 4 \times 6 = 24$

Then chose $d = 11$ *, which is relative prime to 24 and < 35,*

Calculate e using $e\,d \bmod 24 = 1$, *i.e.*

$e \times 11 \bmod 24 = 1$ ➔ *therefore* $e = 11$

Let e & n be public key, or [11, 35], and
d & n be the private key [11, 35]

Now take any message (number) M (such that 0 <= M <=24), for example M = 3, calculate the ciphertext C;

$C = M^e \bmod n = 3^{11} \bmod 35 = 12$

C is sent to the receiver on insecure channel, who will convert it back

by using the private key [11, 35] in the equation;

$C^d \bmod n = 3^{11} \bmod 35$ ➔ *the result is =* ***3***,

which is the original message.

Example 2:

1- *Select two primes;* ***p = 7*** *and* ***q = 17.***
2- *Calculate* ***n = pq*** *= 7 X 17 =* ***119.***
3- *Calculate* $\phi(n)$;

$\phi(n) = (p-1)(q-1) = 6 \text{ X } 16 = 96.$

4- *Select* ***e****, relatively prime to* $\phi(n)$ *and <* $\phi(n)$; *Chose* ***e = 5****.*
5- *Calculate* ***d****;*

$d \equiv e^{-1} \bmod 96 = 5^{-1} \bmod 96$

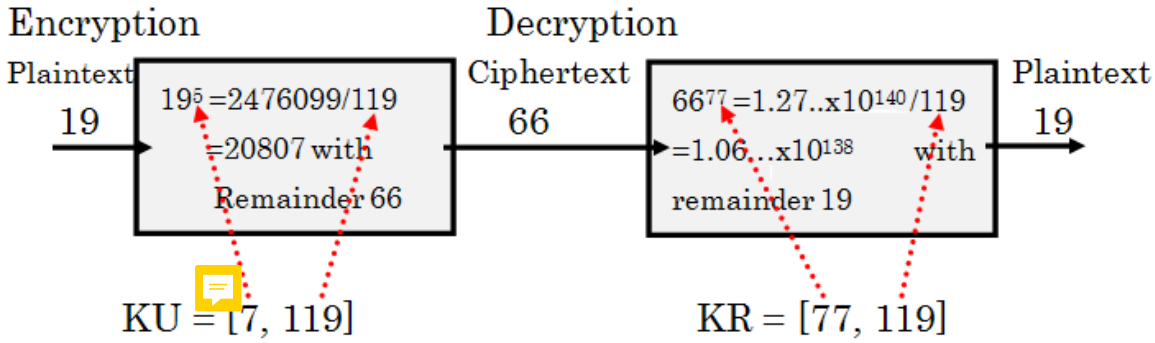
or $d \text{ X } 5 \equiv 1 \bmod 96$ ➔ $d = 77$,

(because $5 \text{ X } d = k \text{ X } 96 + 1$ *or*

$5 \text{ X } 77 = k \text{ X } 96 + 1$, *i.e.*

$385 = 4 \text{ X } 96 + 1$ *or*

$385 = 384 + 1$*).*



Now *using* $KU = [5, 119]$ *as public key and* $KR = [77, 119]$ *as private key*

For a message $M = 19$ *for example, one does the following:*

6- *For encryption at the sender;*

$C = 19^5 \bmod 119 = 66 \bmod 119 = 66$

7- *For decryption at the receiver;*

$M = C^{77} \bmod 119 = 66^{77} \bmod 119 = 19$

Example 3:

1- *Select two primes;* ***p = 17*** *and* ***q = 11.***

2- Calculate $n = pq = 17 \times 11 = 187$.

3- Calculate $\phi(n)$;

$$\phi(n) = (p-1)(q-1) = 16 \times 10 = 160.$$

4- Select **e**, relatively prime to $\phi(n)$ and $< \phi(n)$; Chose **e = 7**.

5- Calculate **d**;

$$d \equiv e^{-1} \bmod \phi(n) \quad = 7^{-1} \bmod 160$$

or $d \times 7 \equiv \bmod 160 \rightarrow d = 23$

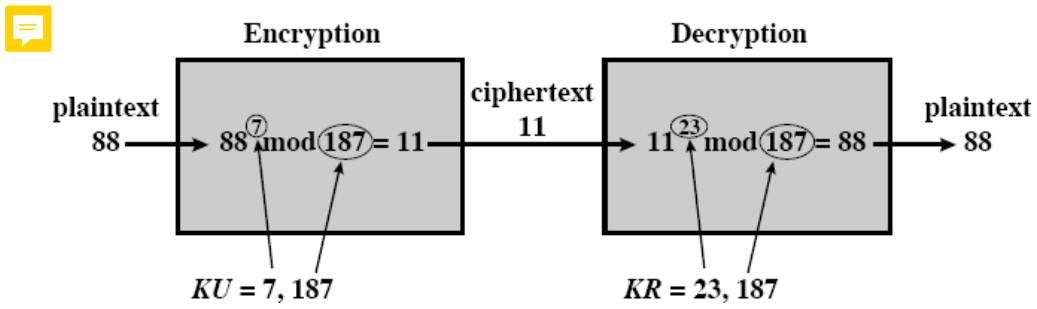
Now *using {7, 187} as public key and {23, 187} as private key for a message M = 88 for example, one does the following:*

6- For encryption at the sender;

$$C = 88^7 \bmod 187 \quad = 11$$

7- For decryption at the receiver

$$M = C^{23} \bmod 187$$
$$= 11^{23} \bmod 187$$
$$= [11^1 \bmod 187][11^2 \bmod 187][11^4 \bmod 187][11^8 \bmod 187]$$
$$= [(11)(121)(55)(33)] \bmod 187$$
$$= [79 \times 720 \times 245] \bmod 187$$
$$= 88$$



4.8 Mini RSA:

To demonstrate RSA algorithm on a mini – case, take the following example.

Let p = 59 and q = 89 [both integers]

And let d = 3403 [also integer]

Now

Calculate n = pq = 59 *89 = 5251

Calculate $\phi(n)$

= (p-1)(q-1) = 58 * 88 = 5104

Select d = 3403, calculate e by: e d = 1 mod $\phi(n)$ or

E * 3403 mod 5104 = 1

The result is e = 3.

Now if the following character set is used:

00	A	10	K	20	U	30	4	40	>
01	B	11	L	21	V	31	5	41	$
02	C	12	M	22	W	32	6	42	+
03	D	13	N	23	X	33	7	43	/
04	E	14	O	24	Y	34	8	44	%
05	F	15	P	25	Z	35	9	45	(
06	G	16	Q	26	0	36	?	47	)
07	H	17	R	27	1	37	.		
08	I	18	S	28	2	38	SPACE		
09	J	19	T	29	3	39	<		

Every two characters are mixed per block [which is simplistic].

Then for the message :

"**Quoth the raven never more**" without spaces, then the message is segmented into 11 blocks, each of two characters. And encryption is achieved using the equation:

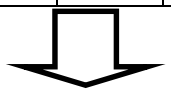
$C_i = M_i^3 \bmod 5251$

(11 block $M_1, M_2, \ldots, M_{11}$) as shown below.

And decryption using the equation:

$M_i = C_i^{3403} \bmod 5251,$

Plaintext	Q	U	O	T	H	T	H	E	R	A	V	E	N	N	E	V	E	R	M	O	R	E
	16	20	14	19	07	19	07	04	17	00	21	04	13	13	04	21	04	17	12	14	17	04
block	1620		1419		0719		0704		1700		2104		1313		0421		0417		1214		1704	
ciphertext	3340		0676		2924		0467		1619		1853		1723		1751		0654		4612		1663	



decryption	1620		1419		0719		0704		1700		2104		1313		0421		0417		1214		1704	
Plaintext	16	20	14	19	07	19	07	04	17	00	21	04	13	13	04	21	04	17	12	14	17	04
	Q	U	O	T	H	T	H	E	R	A	V	E	N	N	E	V	E	R	M	O	R	E

4. 9 Computational Aspects

For computer application of RSA, digital representation of messages and keys is implemented. In this context, there are two important issues; i.e.

- Encryption and Decryption.
- Key generation.

Encryption/Decryption

The results of calculation are reduced to practical values because of **modulo n**. Therefore, for both of the above aspects, exponentiation is used. (Square and Multiply).

Generally, suppose we wish to find value of a^m, with **a** and **m** are *positive integers*. If **m** is binary number $b_k, b_{k-1}, b_{k-2}, \ldots, b_0$, then

$$m = \sum 2^i \qquad , b_0 \neq 0$$

$$a^m = a^{(\sum 2\text{^}i)} = \prod a^{(2\text{^}i)} \quad , b_0 \neq 0$$

Therefore, $a^m \bmod n = (\prod a^{(2\text{^}i)}) \bmod n, \qquad b_0 \neq 0$

$$= [\prod a^{(2\text{^}i)} \bmod n] \bmod n$$

The algorithm is shown below for $a^b \bmod n$.

```
c ← 0; d ← 1
for i ← k downto 0
    do  c ← 2 × c
        d ← (d × d) mod n
        if  bi = 1
            then  c ← c +
```

Example: Employ the above algorithm for a = 7, b = 560 and n = 561
Solution:

We can represent b in binary as b = 560 = 1000110000, therefore k = 9.
The values of **i, b_i, c** and **d** in the algorithm would be as shown in the table below.

I	9	8	7	6	5	4	3	2	1	0
Bi	1	0	0	0	1	1	0	0	0	0
C	1	2	4	8	17	35	70	140	280	560
D	7	49	157	526	160	241	298	166	67	1

The results of the fast modular exponentiation algorithm for
$a^m \bmod n$, i.e. for

Therefore $d = 7^{560} \bmod 561 = = > 1$
(The variable c is included for explanatory purposes).

Key generation
Each user must generate a pair of keys. It involves
Select 2 prime numbers **p** and **q** = = > **n = pq.**
Selecting either **e** or **d** and then calculate the other.

The problem is mainly finding a large prime integer. It must be noted that there is no short cut, but there are few available algorithm, such as *Miller – Robin algorithm*. It is a probabilistic method that is characterized by selecting an odd number and testing its' primarily.