St. Joseph’s College of Engineering & Technology Palai

Department of Computer Science & Engineering

S8 CS

RT801 Security in Computing

Module 3

Security in Computing - Module 3

Syllabus

Cryptography: Basic Encryption & Decryption - Transposition & substitution ciphers - Caesar substitution - Polyalphabetic substitutions

Crypt analysis Symmetric key algorithms - Fiestel Networks - Confusion Diffusion

DES Algorithm Strength of DES

Comparison & important features of modern symmetric key algorithms

Public key cryptosystems The RSA Algorithm - Diffice Hellman key exchange

Comparison of RSA & DES

Message Authentication & Hash functions - Digital signature

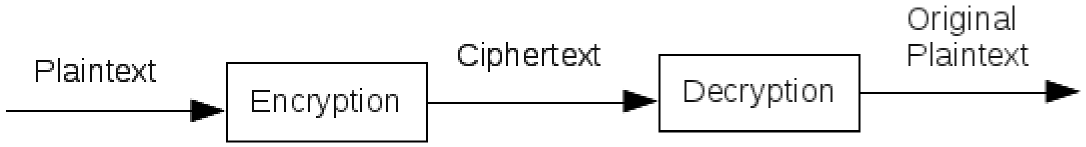
Contents

Cryptography

Cryptography is the strongest tool for controlling against different kinds of security threats. Cryptography is secret writing.

Encryption is the process of encoding a message so that its meaning is not obvious. The many schemes used for encryption constitute the area of study known as cryptography.

Part I. Basic Encryption and Decryption



Encryption (Enciphering) is the process of encoding a message so that its meaning is not obvious. The original message is called plaintext. The encodes message is called ciphertext.

Let the plain text be P.

Let the ciphertext be C.

Then the formula for encryption is, $C = E(P)$

Here E is an encryption algorithm.

Thus an encryption algorithm is applied to plaintext, P and as a result, ciphertext, C is produced.

The reverse process, transforming an encrypted message back to its original form is called decryption (deciphering).

The formal notation for decryption is, $P = D(C)$, or

$$P = D(E(P))$$

where D is the decryption algorithm.

A decryption algorithm is applied to ciphertext and original plaintext is recovered.

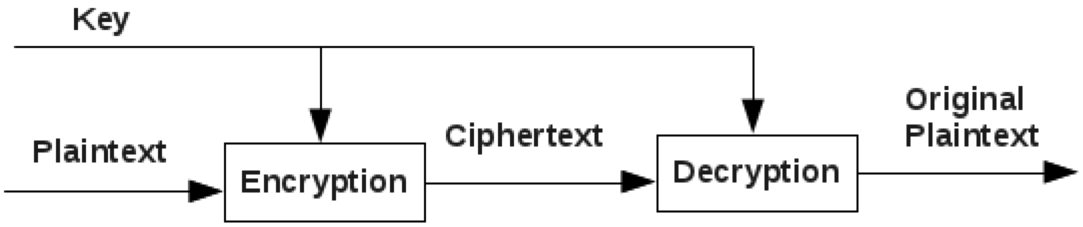
1 Encryption and Decryption Algorithms

A plaintext is transformed to ciphertext using an encryption algorithm. The ciphertext is transformed to plaintext using a decryption algorithm.

These encryption and decryption algorithms use a key. This key is secret.

Symmetric Encryption

If the encryption and decryption algorithms use the same key, the process is called symmetric encryption.



If the key used is K, then encryption is,

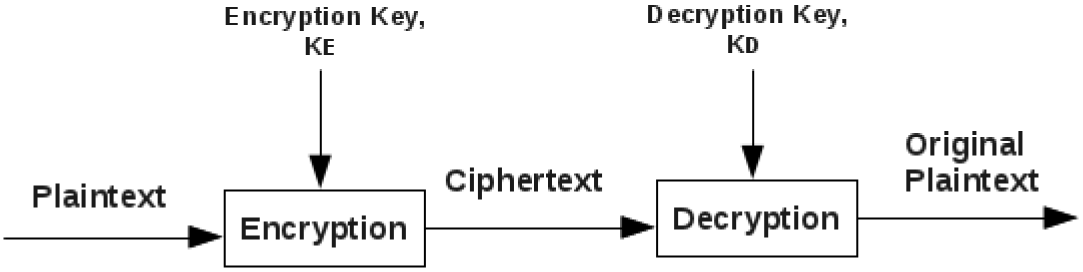
$C = E(K, P)$

Decryption is,

$P = D(K, E(K, P))$

Asymmetric Encryption

If the encryption and decryption algorithms use different keys, then the process is called asymmetric encryption.



If the key used in encryption algorithm is K_E, then encryption is,

$C = E(K_E, P)$

If the key used in decryption algorithm is K_D, then decryption is,

$P = D(K_D, E(K_E, P))$

Of all these objects, only the key is kept secret. Others such as ciphertext, algorithms are normally visible to others.

Cryptanalysis

Cryptanalyst is a person who tries to break an encryption mechanism. Since the cryptanalyst does not know the key used, he tries to break the ciphertext by a number of means.

For example, he may use patterns in the ciphertext, use weaknesses in the encryption algorithm, guessing the key to break ciphertext.

Part II. Encryption Techniques

Two basic building blocks of encryption are,

Transposition, and

Substitution.

A substitution technique is one in which letters of plaintext are replaced by other letters or by numbers or symbols.

A transposition technique is one in which order of letters is rearranged.

2 Substitution Ciphers

In a substitution cipher, one letter is exchanged with another.

Different types of substitution ciphers we study are,

Caeser cipher, and

Polyalphabetic cipher.

2.1 The Caeser Cipher

Julius Caeser is said to have been the first to use this scheme.

In this scheme, each letter is translated to the letter a fixed number of places after it in the alphabet.

Caeser used a shift of 3.

A translation chart of Caeser cipher is shown below:

Plaintext:	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Ciphertext:	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c

For example, a message,

'treaty impossible'

is encoded as

Plaintext:	t	r	e	a	t	y	i	m	p	o	s	s	i	b	l	e
Ciphertext:	w	u	h	d	w	b	l	p	s	r	v	v	l	e	o	h

Thus the formula for encryption is,

$$C = E(3, P)$$

$$= (P + 3) mod 26$$

In general, if we shift by k amounts, the formula is,

$$C = E(k, P)$$

$$(P + k) mod 26$$

The formula for decryption is,

$$P = D(k, C)$$

$$= (C - k) mod 26$$

Drawbacks of Caeser Cipher

It is possible to perform a brute force cryptanalysis and decrypt the message. This is because,

1. The encryption and decryption algorithms are known.
2. There are only 25 keys to try.
3. The language of plain text is easily recognisable.

2.2 Polyalphabetic Cipher

It is a substitution technique. Here, different alphabetic substitutions are used as one proceeds through the plaintext message.

One simplest type of polyalphabetic substitution is Vigenere cipher.

Vigenere Cipher

The following table is used in this scheme.

Plaintext

		a	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**	**m**	**n**	**o**	**p**	**q**	**r**	**s**	**t**
	a	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
	b	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
	c	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
	d	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
	e	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
	f	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
	g	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
	h	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
	i	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
	j	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
	k	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
	l	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
Key	**m**	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
	n	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
	o	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
	p	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
	q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
	r	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
	s	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
	t	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
	u	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
	v	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
	w	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
	x	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
	y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
	z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S

	Plaintext					
Key	**u**	**v**	**w**	**x**	**y**	**z**
a	U	V	W	X	Y	Z
b	V	W	X	Y	Z	A
c	W	X	Y	Z	A	B
d	X	Y	Z	A	B	C
e	Y	Z	A	B	C	D
f	Z	A	B	C	D	E
g	A	B	C	D	E	F
h	B	C	D	E	F	G
i	C	D	E	F	G	H
j	D	E	F	G	H	I
k	E	F	G	H	I	J
l	F	G	H	I	J	K
m	G	H	I	J	K	L
n	H	I	J	K	L	M
o	I	J	K	L	M	N
p	J	K	L	M	N	O
q	K	L	M	N	O	P
r	L	M	N	O	P	Q
s	M	N	O	P	Q	R
t	N	O	P	Q	R	S
u	O	P	Q	R	S	T
v	P	Q	R	S	T	U
w	Q	R	S	T	U	V
x	R	S	T	U	V	W
y	S	T	U	V	W	X
z	T	U	V	W	X	Y

The process of encryption is simple.

Given a key letter, x and a plain text letter, y,

the ciphertext letter is at the intersection of the row labeled x and the column labeled y.

For example,

if the key letter is ’d’ and plaintext letter is ’m’,

the ciphertext letter will be ’P’.

To encrypt a message, a key is needed as long as the message. Usually, the key is a repeating keyword.

For example, if the keyword is ’deceptive’, and the plaintext message is 'we are discovered',

the encryption is done as follows:

Key:	d	e	c	e	p	t	i	v	e	d	e	c	e	p	t
Plaintext:	w	e	a	r	e	d	i	s	c	o	v	e	r	e	d
Ciphertext:	Z	I	C	V	T	W	Q	N	G	R	Z	G	V	T	W

Decryption is also simple.

The key letter identifies the row.

The position of the ciphertext letter in that row identifies the column, and

the plaintext letter is at the top of that column.

Venam Cipher

It is a type of polyalphabetic substitution. Here a key is selected which has the same length of the plaintext, Thus there is no repetition of the key.

This system works on binary data.

The formula for encryption is,

$C_i = P_i \bigoplus K_i$

where

C_i is the i^{th} binary digit of ciphertext.

P_i is the i^{th} binary digit of plaintext.

K_i is the i^{th} binary digit of key.

$\bigoplus$ is the XOR operation.

The formula for decryption is,

$P_i = C_i \bigoplus K_i$

The formulae for encryption and decryption look similar due to the property of XOR operation.

One Time Pad

It is a type of polyalphabetic substitution. It is an improvent over Venam cipher.

Here a random key is usedthat is as long as the message, so that the key is not repeated. Also the key is used to encrypt and decrypt a single message, and then it is discarded.

Thus every new message requires a new key. This scheme is unbreakable.

For example,

Let the plaintext be 'mr mustard with knife'

Let the random key generated for this is 'pxlmvmsydofuyrvzwc'.

Then the encryption is as follows:

Key:	p	x	l	m	v	m	s	y	d	o	f	u	y	r	v	z	w	c
Plaintext:	m	r	m	u	s	t	a	r	d	w	i	t	h	k	n	i	f	e
Ciphertext:	B	O	X	G	N	F	S	P	G	K	N	N	F	B	I	H	B	G

Drawbacks of One time pad scheme

A difficulty with this scheme is the problem of making a large number random keys.

Another problem is key distribution and protection. For every message, a key of equal length is needed by both sender and receiver.

Part III. Cryptanalysis

Cryptanalyst is a person who tries to break encryption.

A cryptanalyst faces four possible situations, depending on what information is avaialbale. They are,

analyst has only ciphertext available,

analyst has full or partial plaintext avaialble,

analyst has ciphertext of any plaintext avaialble,

analyst has algorithm and ciphertext available,

analyst has ciphertext and plaintext available.

Analyst has only ciphertext available (Ciphertext only Attack)

Here decryption is done based on probabilities, distributions and characteristics of the avaialable ciphertext and publicly availability of public knowledge.

Analyst has full or partial plaintext avaialble (Known Plaintext Attack)

Here analyst may use additional information. For example, he may know that message was sent between USA and Russia. From that information, he may guess some words in the message.

Also the message may be a letter from a corporate president to the Sales force. The nthe letter may have a particular form, (TO: Sakles force, From: President, Subject: weekly sales update, Date:dd/mm/yy).

In the baove cases, analyst can use probable plaintext analysis. Analyst may find places where the known message fits with the deciphered parts, thereby giving more clues about the total message.

Analyst has ciphertext of any plaintext avaialble (Chosen Plaintext Attack)

Here the analyst may intrude into the sender's network and send his own messages. When the receiver gets these messages, he decrypt them and analyst can observe the statistics.

Analyst has algorithm and ciphertext available (Chosen Ciphertext Attack)

In this analyst can run the encryption algorithm on massive amounts of plaintext to find one plaintext message that encrypts as the ciphertext. This may help him to deduce the key.

Analyst has ciphertext and plaintext available

This may help cryptanalyst to deduce the key.

3 Confusion and Diffusion

An encryption algorithm should transform the plaintext to ciphertext such that a cryptanalyst cannot readily recognise the message.

Confusion

The analyst should not be able to predict what will happen to the ciphertext by changing one character in the plaintext. Thsi characteristic is called confusion.

It will take an analyst a long time to determine the relationship between plaintext, key and ciphertext, if the algorithm provides good confusion. It will take an analyst long time to break the code.

Consider the Caeser cipher.

The algorithm does not provide good confusion because an analyst who is able to deduce the transformation of few letters can also predict the transformation of remaining letters.

Consider One time pad scheme.

It provides good confusion because one plaintext letter can be transformed to any letter at different places.

Diffusion

The encryption algorithm should spread the information from plaintext over the entire ciphertext so that changes in the plaintext affect many parts of the ciphertext. This principle is called diffusion.

Diffusion is thus the characteristic of distributing the information from single plaintext letters over the entire input.

Good diffusion means the analyst needs to acces too much of the ciphertext to be able to infer the algorithm.

Part IV. Data Encryption Standard (DES)

It is a symmetric encryption algorithm.

It is a system developed for the US government. It is the current cryptographic standard specified by ISO. Many hardware and software systems are based on this.

DES encryption algorithm is a combination of two operations,

substitution, and

transposition.

There are 16 cycles in the algorithm. The key is 64 bits long.

The entire data is divided into blocks of 64 bits each.

The DES algorithm uses two techniques to hide information. They are,

confusion, and

diffusion.

Substitution provides confusion, and

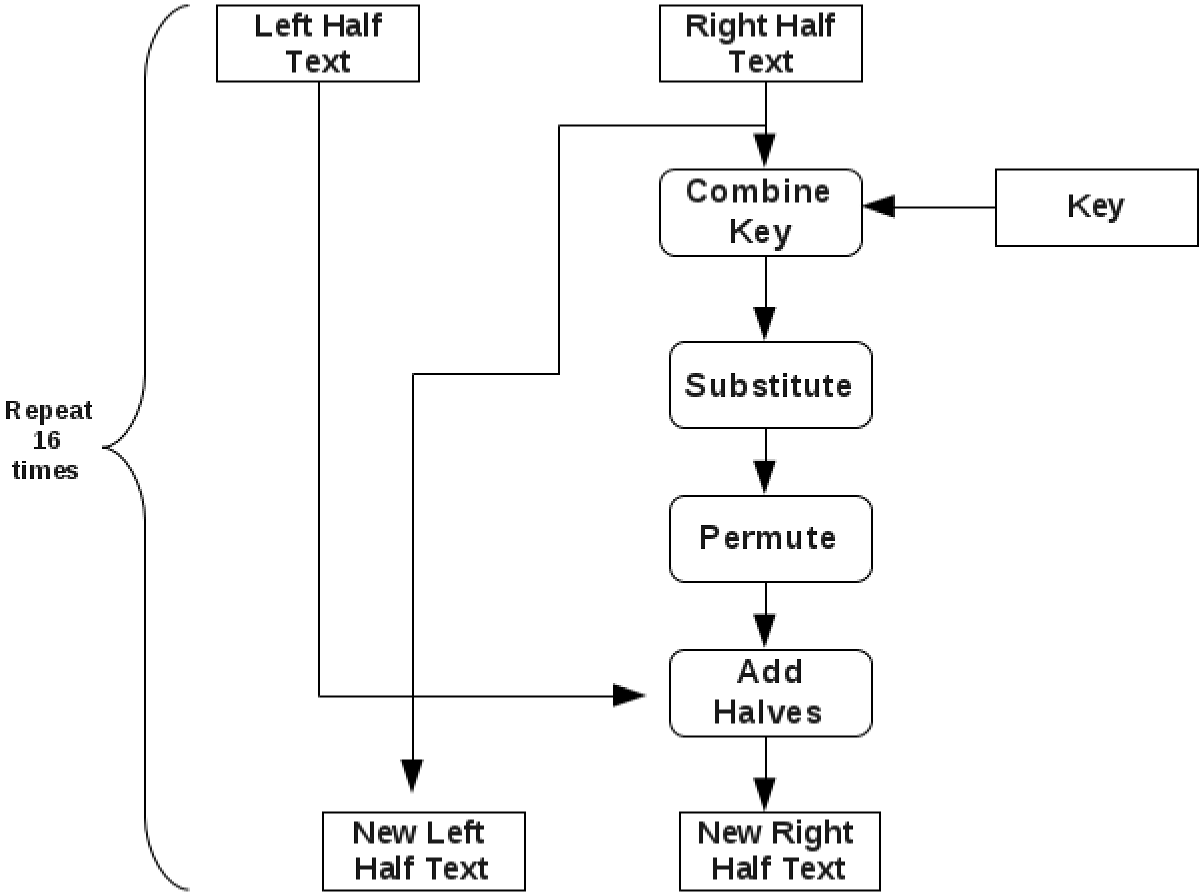
Transposition provides diffusion.

Thus the algorithm ensures that output bits have no relation to input bits.

DES uses only standard arithmetic and logic operations so that it can be easily implemented in hardware and software.

Outline of DES

An out line of the operations in the algorithm is shown below:

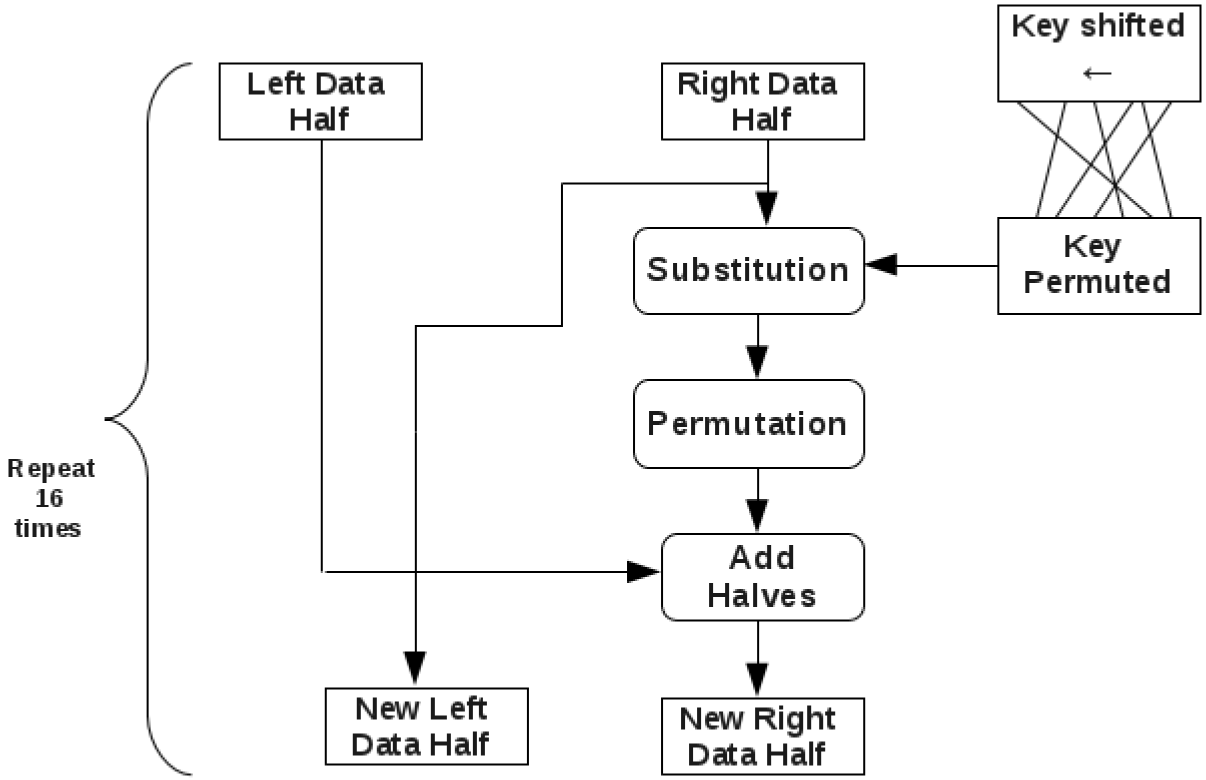


4 Operation of DES Algorithm

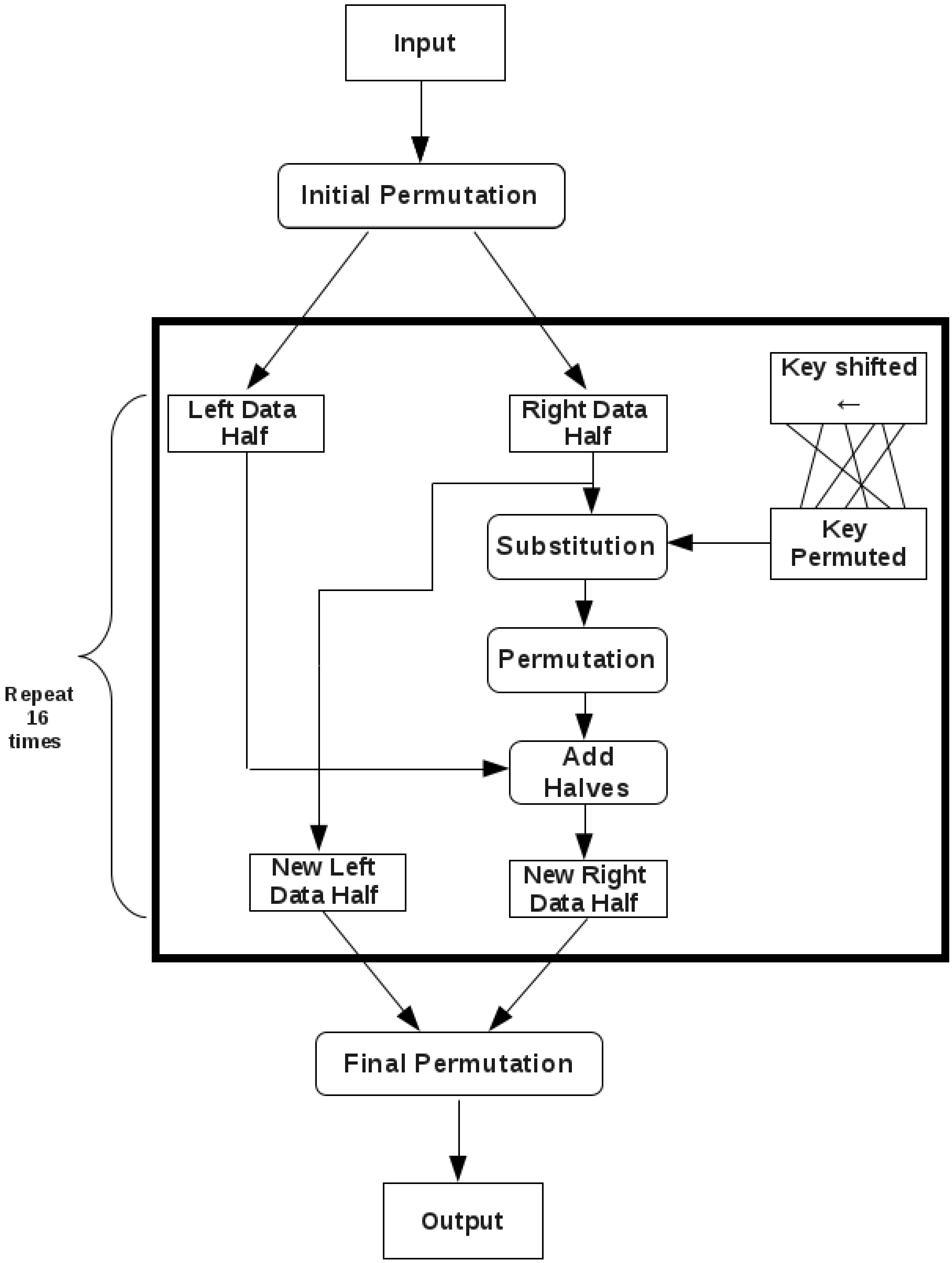
DES splits data blocks (64 bits) in half (32 bits), works with each half independently. It combines key with one half and swaps the two halves. Thsi process is repeated 16 times.

It is an interactive algorithm using just table lookups and simple bit operations.

One cycle of DES is shown below:



The overall operation of DES is shown below:



Initial Permutation

Input data is divided into blocks of 64 bits each. The 64 bit blocks are permuted using initial permutation.

Following table is used for this.

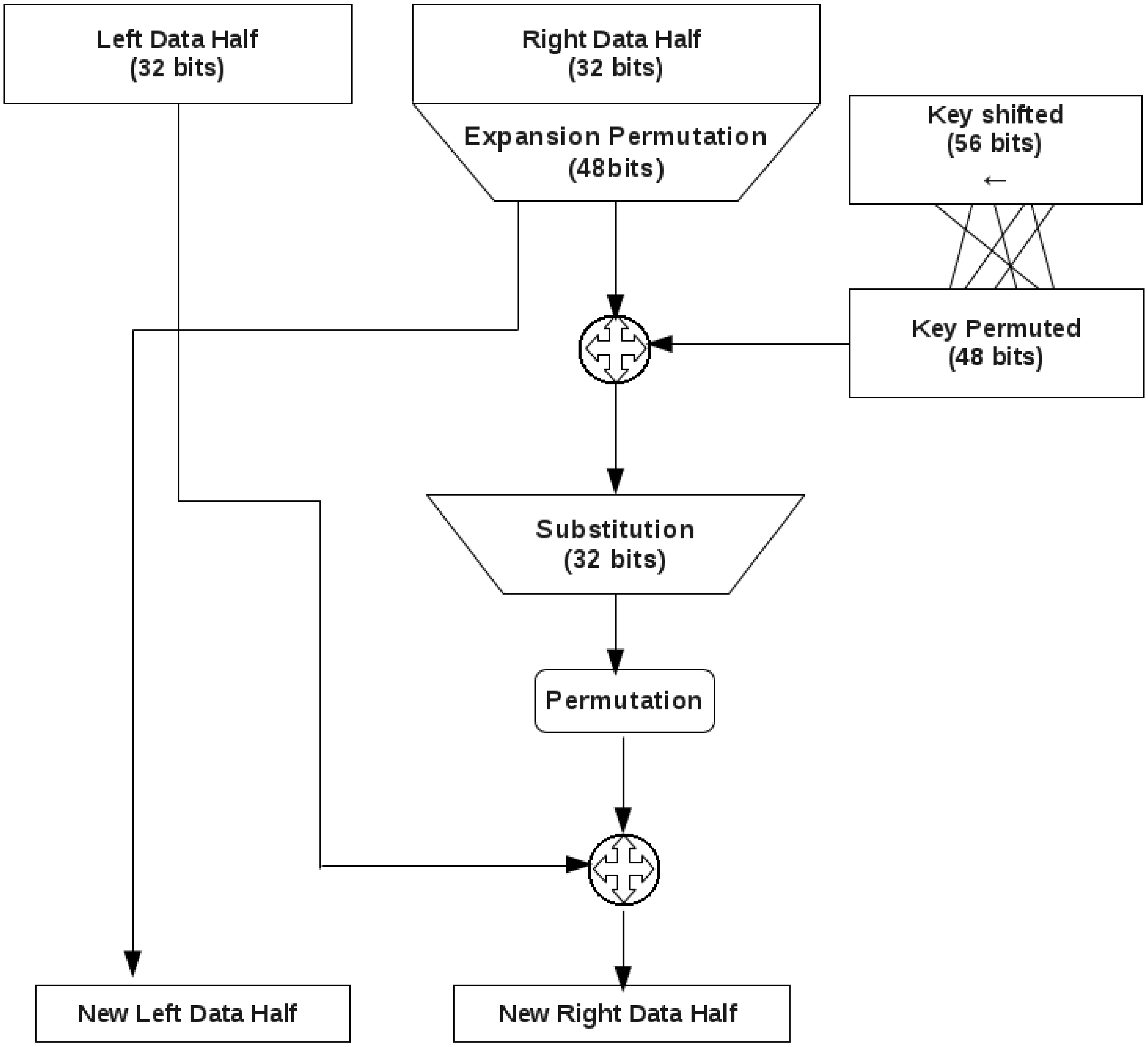
Goes to Position

Bit								
1-8	40	8	48	16	56	24	64	32
9-16	39	7	47	15	55	23	63	31
17-24	38	6	46	14	54	22	62	30
25-32	37	5	45	13	53	21	61	29
33-40	36	4	44	12	52	20	60	28
41-48	35	3	43	11	51	19	59	27
49-56	34	2	42	10	50	18	58	26
57-64	33	1	41	9	49	17	57	25

One Cycle of DES

The 64 permuted bits are divided into left half and right half of 32 bits each.

The following shows the operation in one cycle in more detail.



Key shifting and Permuting

The key is shifted left by a number of bits and permuted.

The 64 bit key is reduced to 56 bit key by removing every 8th bit.

This 56 bit key is divided into two halves (28 bits each).

These halves are shifted left by a number of digits. How much the halves are shifted in every cycle is shown below:

Cycle Number	Bits Shifted
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1

These halves are then pasted together.

Next, 48 of these 56 bits is permuted to use as the key. Which among these 56 bits are selected are shown below:

Key Bit	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Selected for Position	5	24	7	16	6	10	20	18	_	12	3	15	23	1

Key Bit	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Selected for Position	9	19	2	_	14	22	11	_	13	4	_	17	21	8

Key Bit	29	30	31	32	33	34	35	36	37	38	39	40	41	42
Selected for Position	47	31	27	48	35	41	_	46	28	_	39	32	25	44

Key Bit	43	44	45	46	47	48	49	50	51	52	53	54	55	56
Selected for Position	_	37	34	43	29	36	38	45	33	26	42	_	30	40

Now the key contains 48 bits.

Right Half (Expansion Permutation)

The algorithm expands the 32 bit right half 48 bits by repeating certain bits. How to expand this is shown below:

Bit	1	2	3	4	5	6	7	8
Moves to Position	2, 48	3	4	5, 7	6, 8	9	10	11, 13

Bit	9	10	11	12	13	14	15	16
Moves to Position	12, 14	15	16	17, 19	18, 20	21	22	23, 25

Bit	17	18	19	20	21	22	23	24
Moves to Position	24, 26	27	28	29, 31	30, 32	33	34	35, 37

Bit	25	26	27	28	29	30	31	32
Moves to Position	36, 38	39	40	41, 43	42, 44	45	46	47, 1

Now the right half contains 48 bits.

Combining right Half and Key

Now we have a 48 bit right half and a 48 bit key.

These 48 bit right half and key are combined using an XOR operation.

The results are moved into S boxes.

Substitution

This is done using 8 S boxes.

From the above, result contains 48 bits. These 48 bits are divided into 8 blocks each containing 6 bits.

In every box, 6 bits are replaced with 4 bits using the following table.

Box	**Row**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
S1																	
	0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
	3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Box	**Row**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
S2																	
	0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
	1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
	2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
	3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

Box	**Row**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
S3																	
	0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
	1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
	2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
	3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

Box	Row	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S4																	
	0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
	1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
	2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
	3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

Box	Row	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S5																	
	0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
	1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
	2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
	3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

Box	Row	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S6																	
	0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
	1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
	2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
	3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

Box	Row	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S7																	
	0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
	1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
	2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
	3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

Box	Row	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S8																	
	0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
	1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
	2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
	3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

This is done as follows:

Let the 6 bit data in block S1 be 101011.

b_1	b_2	b_3	b_4	b_5	b_6
1	0	1	0	1	1

Here b_1 and b_6 are taken together (11- decimal value is 3). Call this r.

Then b_2, b_3, b_4, b_5 are taken together (0101 - decimal value is 5). Call this c.

From the above table, value corresponding to row, r and column, c is selected. [Here the value corresponding to row 3 and column 5 in block S1 is 9. The binary value is 1001].

The decimal value from the table is converted to 4 bit binary value.

In this way, 6 bit blocks are transformed to 4 bit blocks.

In this way, substitution is done for every block and result will have 32 bits (8 blocks, each produces 4 bits).

Permutation

From the above step, we get 32 bits. These 32 bits are permuted using the table given below.

Goes to Position

Bit								
1-8	9	17	23	31	13	28	2	18
9-16	24	16	30	6	26	20	10	1
17-24	8	14	25	3	4	29	11	19
25-32	32	12	22	7	5	27	15	21

For instance bit 5 of the result moves to bit 13.

Combining Left Half and above Result

The above 32 bit value is XORed with the 32 bit left half data. The result is the new 32 bit right data half.

Final Permutation

After 16 cycles, there is a final permutation. This final permutation is the inverse of initial permutation.

Goes to Position

Bit								
1-8	58	50	42	34	26	18	10	2
9-16	60	52	44	36	28	20	12	4
17-24	62	54	46	38	30	22	14	6
25-32	64	56	48	40	32	24	16	8
33-40	57	49	41	33	25	17	9	1
41-48	59	51	43	35	27	19	11	3
49-56	61	53	45	37	29	21	13	5
57-64	63	55	47	39	31	23	15	7

4.1 Summary of DES

The 64 bit key is reduced to 56 bits.

A block of 64 data bits is permuted using initial permutation.

16 cycles of the following operation is done:

The key is shifted and permuted.

Half of the data block is transformed with substitution and permutation functions, and the results are combined with the remaining half of the data block.

After 16 cycles, the data block is permuted with the final permutation.

5 Decryption of DES

The above algorithm itself is used for decryption. thus DES is a reversible procedure.

6 Strength of DES

The level of security provided by DES falls in 2 areas:

key size, and

nature of algorithm.

Key Size

The key length is 56 bits. There are 2^{56} keys. A computer may take around 1000 years to break the cipher.

But it is estimated that a parallel computer with 1 million encryption devices would take only 10 hours to break the cipher.

In July 1998, Electronic Frontier Corporation announced that it had broken a DES encryption using a special DES cracker machine. Thus it is proved that DES is insecure.

Nature of DES Algorithm

Cryptanalysis is possible by exploiting the features of the algorithm. It is concerned with eight S boxes. Over the years, a number of regularities and unexpected behaviours of S boxes have been discovered.

Number of Iterations

Many wondered whether 16 iterations are sufficient. But experiments suggest that 8 iterations are sufficient to eliminate any observable dependence.

Complements

If P is a plaintext and K is a key, then

$$C = DES(P, K)$$

Then,

$$\neg C = DES(\neg P, \neg K)$$

This is a weakness.

Weak Keys

Initial key is split into two halves and these halves are independently shifted. If the values shifted are all 0s or 1s, then the key used in each cycle is the same as for all other cycles. For keys having 0s and 1s only, encryption is same as decryption.

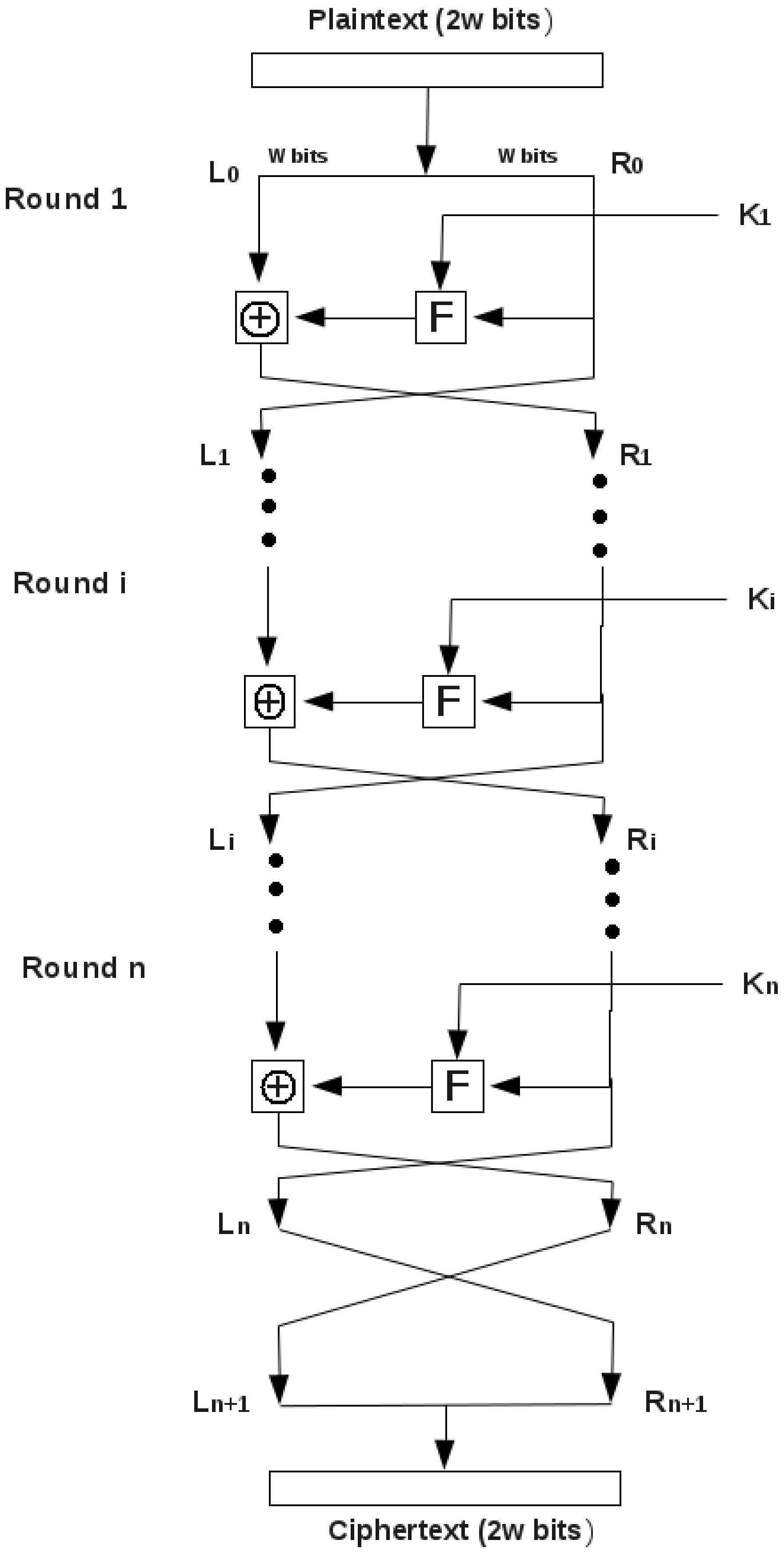
Some pairs of keys have identical decryption.

Part V. Feistel Networks

Feistel proposed a model for encryption in 1973. His model of encryption uses alternate substitutions and permutations.

The DES algorithm we learned in the last section is based on this model.

Following figure shows the structure proposed by Feistel called as Feistel Network.



From the diagram, input is plaintext of size 2w bits and a key K.

The plaintext is divided into two halves, L_0 and R_0.

The two halves pass through n cycles of processing and then combine to produce ciphertext.

Each cycle i has inputs L_{i-1}, R_{i-1}, derived from the previous cycle.

Also subkey K_i derived from overall key K.

All cycles have the same structure.

Consider a cycle.

A substitution is performed on the left half of the data. this is done as follows:

A function, F is applied to the right half of data.

Then this result is XORed with left half of the data.

Then a permutation is performed that consists of interchanging the two halves.

Realization of Freistel Network

To implement an encryption mechanism using such a network, the folowing are to be considered.

Block size

Larger blocks provide more security. Normally, data is divided into blocks of 64 bits.

Key Size

Larger key means greater security. Key sizes of 64 bits were used (DES). But now 128 bit keys are used.

Number of cycles

Normally 16 cycles are used.

Other considerations are choice of function, F and subkey generation algorithm.

Feistel Decryption Algorithm

The process of decryption is same as the encryption algorithm.

Ciphertext is given as input. the subkeys are used in reverse order. Kn is used in the first cycle, K_{n-1} in the second cycle and so on.

Part VI. Public Key Cryptosystems (Asymmteric Encryption)

In our previous encryption mechanisms, keys were kept secret. Also a single secret key was used for encryption as well as decryption.

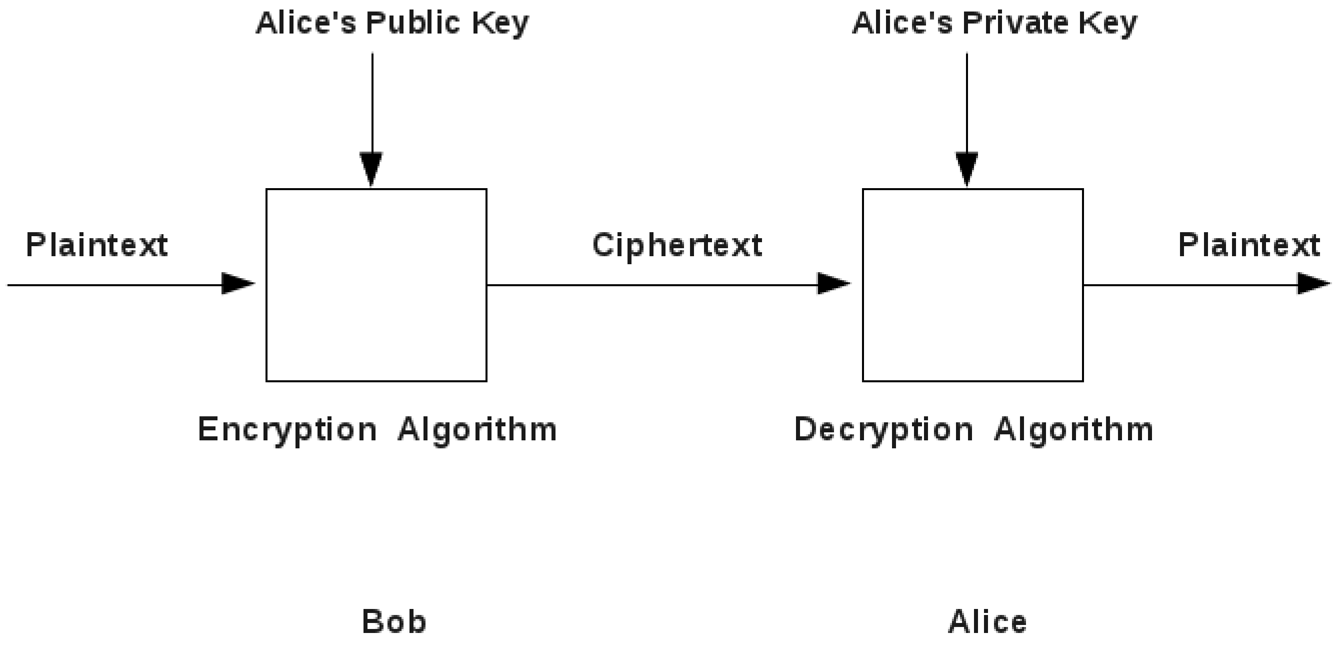
In public key cryptosystems, keys can be made public.

Public key cryptosystems are based on mathematical models rather than substitution and permutation. In public key cryptography, two keys are used, one for encryption and other for decryption.

Eg. for public key cryptosystem is RSA algorithm.

A problem with symmetric encryption is key distribution. That is, same key is used for for encryption and decryption. So both sender and receiver should use the same key.

7 Components of a Public key Cryptosystem



A public key encryption scheme has the following components:

Plaintext

Encryption algorithm

Public and Private keys

A pair of keys are selected. One is called public key, and the other is called private key.

Public key can be made public. Private key is kept secret.

One of the keys is used for encryption, and the other is used for decryption.

For example, if public key is used for encryption, then private key is used for decryption.

Also, if private key is used for encryption, then public key is used for decryption.

Ciphertext

Decryption Algorithm

8 Steps in Public key Encryption and Decryption

Encryption using Public Key and Decryption using Private Key

The steps are as follows:

1. A user generates a pair of keys (public key and private key) to be used for encryption and decryption.

2. The user places one key in a public register. This is the public key.

   The other key is kept secret. This is the private key.

   A user maintains a file that contains the public keys of different users.

   For example, in the above diagram, Bob knows the public key of Alice.

3. If Bob wishes to send Alice a message, Bob encrypts the message using Alice's public key. Ciphertext is produced and sent to Alice.

4. When Alice decrypts the ciphertext, she does it using her private key.

Thus in this mechanism, all users can access public keys. Private keys are kept secret by each participant.

At any time, a person can change the keys by publishing the new public key and keeping the corresponding private key secret.

The notation for encryption is,

$C = E(K_{pub}, P)$

where K_{pub} is the public key of the receiver.

P is the plaintext,

E is the encryption algorithm,

C is the ciphertext.

The notation for decryption is,

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

where $K_{private}$ is the private key of the receiver.

P is the plaintext,

D is the decryption algorithm,

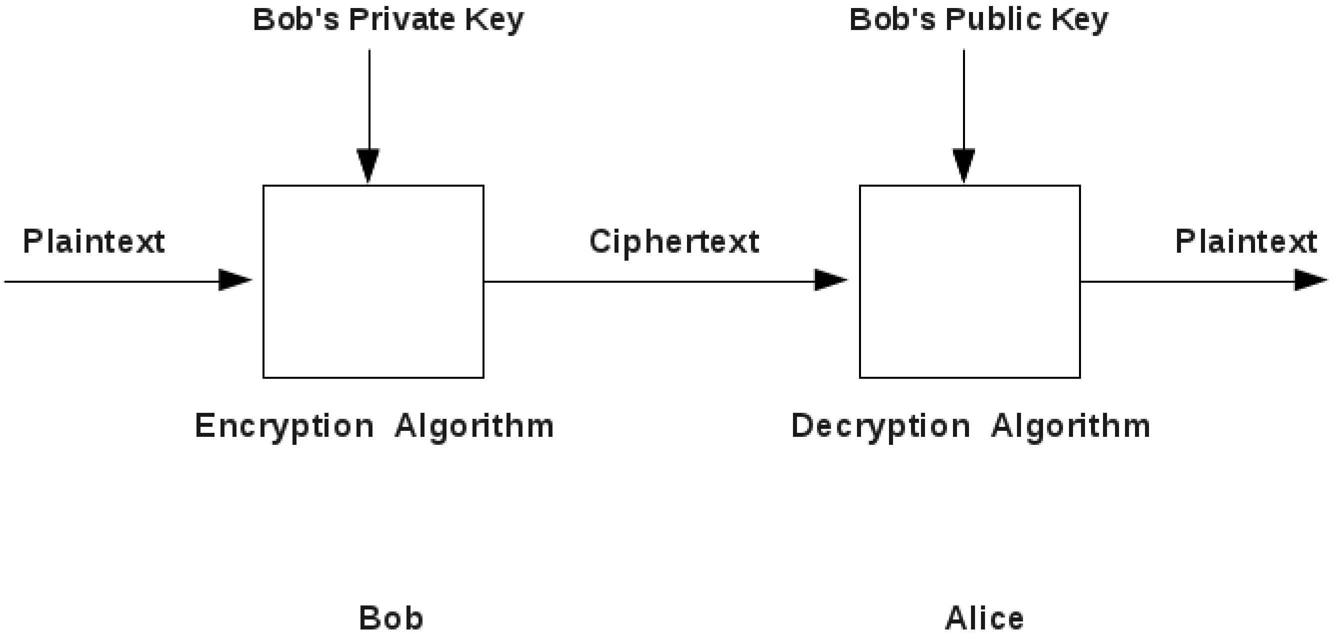
C is the ciphertext.

Encryption using Private Key and Decryption using Public Key

Encryption can be done using the private key. The resulting ciphertext is sent. The receiver can decrypt the ciphertext using public key.

Here private and public keys correspond to the sender.

This is shown below:



Here, the notation for encryption is,

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

where $K_{private}$ is the private key of the sender.

P is the plaintext,

E is the encryption algorithm,

C is the ciphertext.

The notation for decryption is,

$P = D(K_{pub}, C)$

where K_{public} is the public key of the sender.

P is the plaintext,

D is the decryption algorithm,

C is the ciphertext.

Thus anyone of the keys can be used for encryption. Then the other key is used for decryption.

9 RSA Algorithm (Rivest-Shamir-Adleman)

It is the most widely accepted public key encryption algorithm.

Following shows the RSA algorithm:

Part I: Key Generation

The public and private keys are generated as follows:

1. Select p, q. [p and q both prime, $p \neq q$]
2. Calculate $n = pxq$.
3. Calculate $\phi(n) = (p-1)(q-1)$.
4. Select integer e. [$gcd(\phi(n), e) = 1; 1 < e < \phi(n)$]

5. Calculate d. [$d = e^{-1}(mod(\phi(n))$]
6. Public key. [public key = {e, n}]
7. Private key. [private key = {d, n}]

Example:

Consider the following example for generating keys:

1. Select two prime numbers, p=17, q= 11.
2. Calculate $n = pxq = 17x11 = 187$
3. Calculate $\phi(n) = (p-1)(q-1) = 16x10 = 160$
4. Select integer e such that e is relatively prime to $\phi(n) = 160$ and less than $\phi(n)$;

   we choose $e = 7$.
5. Determine d such that $de = 1(mod\, 160)$ and $d < 160$.

   The value of d is 23, since $23x7 = 161 = 10x160 + 1$.

Now we got e=7, d=23.

Thus the keys are,

Public key = {e, n} = {7, 187]

Private key = {d, n] = {23, 187}

Part II: Encryption

Encryption using public key is done as follows:

1. Plaintext, M $M < n$
2. Ciphertext, C $C = M^e mod\, n$

Example:

Let the plaintext, M=88.

We have the public key = {7, 187}

Then,

$C = M^e mod\, n = 88^7 mod\, 187$

$= 11$

Thus the ciphertext is 11.

Part III: Decryption

Decryption using private key is done as follows:

1. Ciphertext, C
2. Plaintext, M $M = C^d mod\, n$

Example:

We have the ciphertext, C = 11.

We have the private key = {23, 187}

Plaintext, $M = 11^{23} mod\, 187$

= 88

Security of RSA

In four possible ways, RSA algorithm can be attacked.

Brute force

This involves trying all possible private keys.

Mathematical attacks

It is done by factoring the product of two primes.

Timing attacks

This depends on the running time of decryption algorithm.

Chosen ciphertext attacks

This uses the properties of RSA algorithm.

Part VII. Diffie - Hellman Key Exchange

The purpose of Diffie - Hellman key exchange algorithm is to enable two users to exchange a key securely.

The algorithm is as follows:

Part I: Global Public Elements

There are two publicly known numbers:

q,	a prime number
$\alpha,$	a primitive root of q.

Example:

Let the prime number, q = 353.

Select α,

Let us select,$\alpha = 3$, which is a primitive root of 353.

[Primitive root of a prime number, p is one whose powers modulo p generate all integers from 1 to p-1. That is, if a is a primitive root of prime number, p, then numbers

a mod p, a^2mod p, a^3mod p, ... , a^{p-1} mod p. they are distinct and consists of integers from 1 through p-1.

Here, for $\alpha = 3$, 3 mod 353, 3^2mod 353, 3^3mod 353,.......]

Part II: User A Key Generation

Select private , X_A, $X_A < q$

Calculate public , Y_A, $Y_A = \alpha^{X_A} mod\, q$

Example:

We have, q = 353, $\alpha = 3$.

Let user A selects, $X_A = 97$.

Then user A computes its public, $Y_A = \alpha^{X_A} mod\, q = 3^{97} mod\, 353 = 40$

Part III: User B Key Generation

Select private , X_B, $X_B < q$

Calculate public , Y_B, $Y_B = \alpha^{X_B} mod\, q$

Example:

We have, q = 353, $\alpha = 3$.

Let user B selects, $X_B = 233$.

Then user B computes its public, $Y_B = \alpha^{X_B} mod\, q = 3^{233} mod\, 353 = 248$

Part IV:

Users A and B keep their private keys (X_A, X_B).

They exchange or make public their public keys (Y_A, Y_B).

Part V: Calculation of Secret Key by User A

$K = (Y_B)^{X_A} mod$ q

User A computes the key using the above formula.

Example:

User A computes,

$K = (Y_B)^{X_A} mod$ q $= 248^{97} mod$ 353 = 160

Part VI: Calculation of Secret Key by User B

$K = (Y_A)^{X_B} mod$ q

User B computes the key using the above formula.

Example:

User B computes,

$K = (Y_A)^{X_B} mod$ q $= 40^{233} mod$ 353 = 160

Thus, User A and User B have got the secret key, 160.

Part VIII. Digital Signature

In some situations, there is no complete trust between sender and receiver. A solution to this problem is digital signature.

Consider an example communication between John and Mary.

Let John sends a message to Mary. Following problems can occur.

1. Mary may forge a different message and claim that it came from John.

2. John can deny sending a message to Mary.

Two approaches for implementing digtial signature are,

Direct digital signature, and

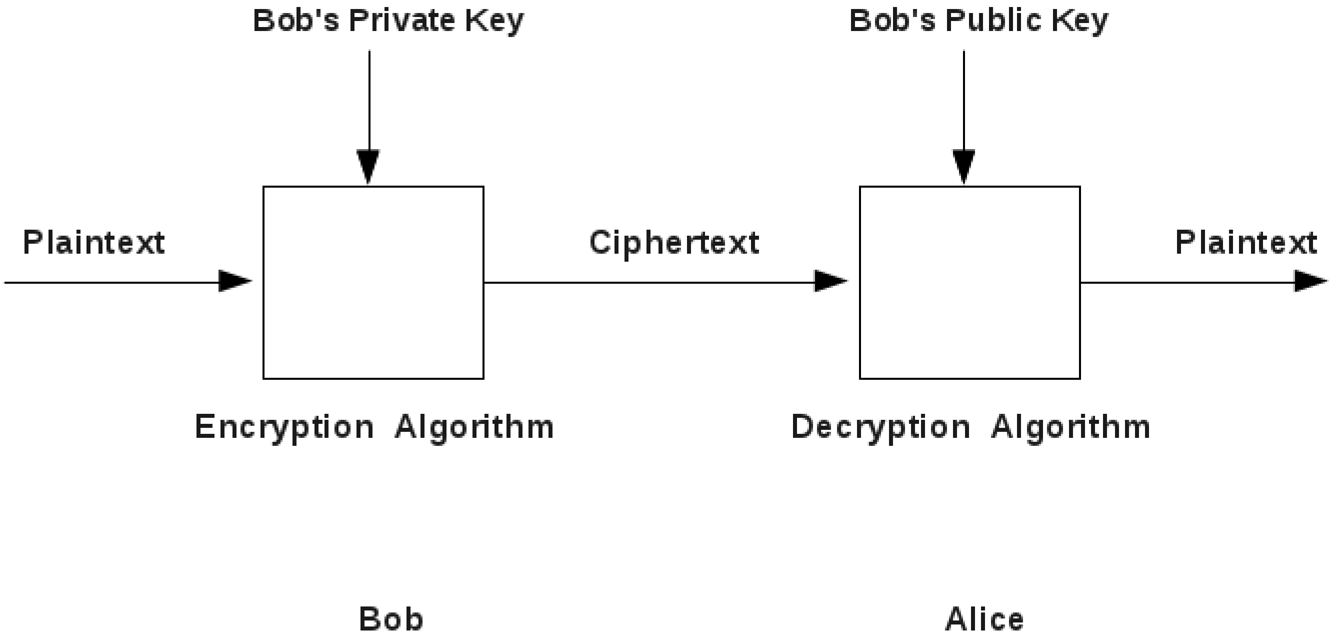
Arbitrated digital signature.

10 Direct Digital Signature

This mechanism uses the asymmetric encryption scheme we learned in the earlier section.

Here there is a sender and receiver. Receiver knows the public key of the sender.

Suppose Bob wants to send a text to Alice with this facility. Then the communication is as follows:



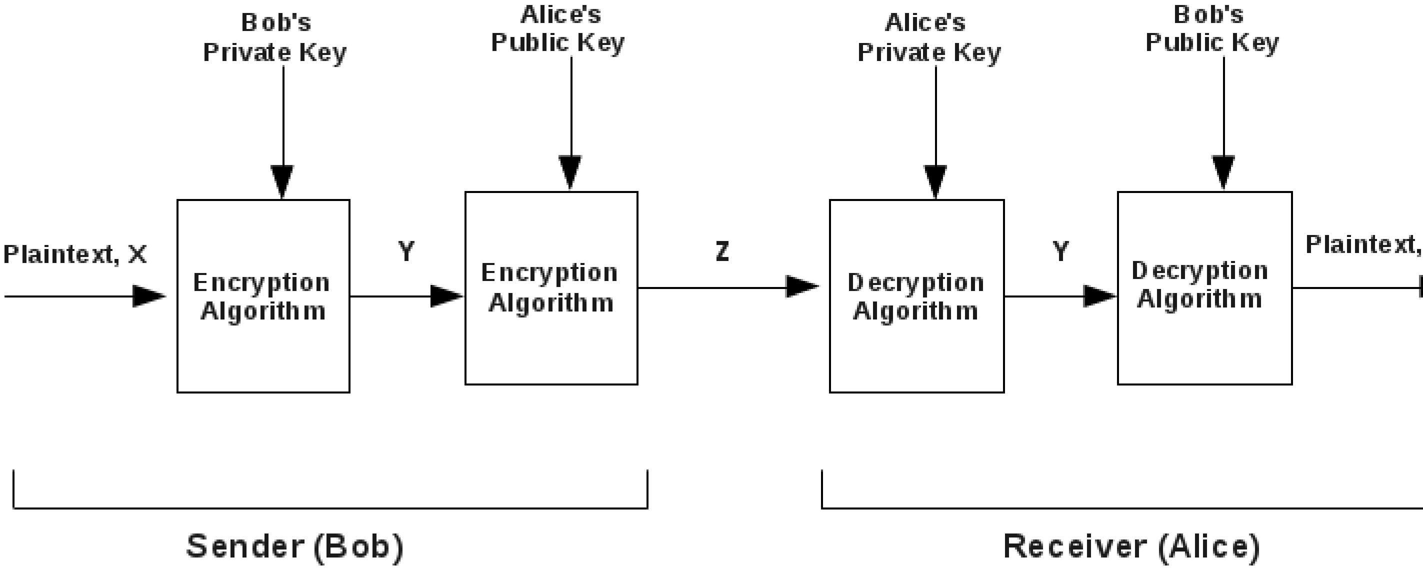
Then Bob encrypts the text with his private key. The encrypted text or ciphertext is the digital signature.

On getting the ciphertext, Alice decrypts it using Bob's public key. She will get the original plaintext.

When a dispute occurs, a third party can decrypt the ciphertext (digital signature) using Bob's public key.

In the above, the ciphertext can be decrypted by anyone who knows Bob's public key. It is to be made secure.

This is done as follows:



Here Bob encrypts the message (X) using his private key. The result (Y) is again encrypted using Alice's public key. The resulting ciphertext is Z.

The ciphertext, Z is sent to Alice.

Alice decrypts the ciphertext, Z using her private key. The resulting ciphertext (Y) is again decrypted using Bob's public key. The result is the original plain text, X.

This mechanism provides both security as well as digital signature.

11 Arbitrated Digital Signature

Here a signed message from sender X to receiver Ygoes through an arbiter A.

The arbiter, a exposes the signature to a number of tests to check its origin and content. The message is then sent to Y with an indication that it has been verified.

Different ways this can be done. One way is explained below:

Here, X sends a message to Y.

First, X sends the message to arbiter, A as follows:

$$X \longrightarrow A : \; M \,||\, E\,(\,K_{xa}, [ID_x||H(M)]\,)$$

K_{xa} is the secret key shared by X and A.

X constructs the message, M and computes its hash value, H(M). X sends the message plus signature to A.

Signature consists of an identifier ID_X plus hash value, H(M).

A on getting the message, transmits the message to Y as follows:

$A \longrightarrow Y : \; E\,(\,K_{ay}, [\,ID_x||M||\; E\,(K_{xa}, [ID_x||H(M)\,]\;||T])$

K_{ay} is the secret key shared by A and Y.

A encrypts the message with K_{ay}.

The message includes ID_x, original message from X, signature and timestamp, T.

Y can decrypt this to recover the message and signature.

When a dispute comes, Y sends the following message ot A.

$E\,(\,K_{ay}, [\,ID_x||M||\; E\,(K_{xa}, [ID_x||H(M)\,])]\,)$

The arbiter, A uses K_{ay} to recover ID_x , M and signature. A uses K_{xa} to decrypt the signature and to verify the hash code.

Part IX. Message Authentication and Hash Functions

Message authentication is a procedure to verify that received messages come from the actual source and have not been changed.

Different techniques used for message authentication are as follows:

Message Encryption,

Message authentication code,

Hash Function.

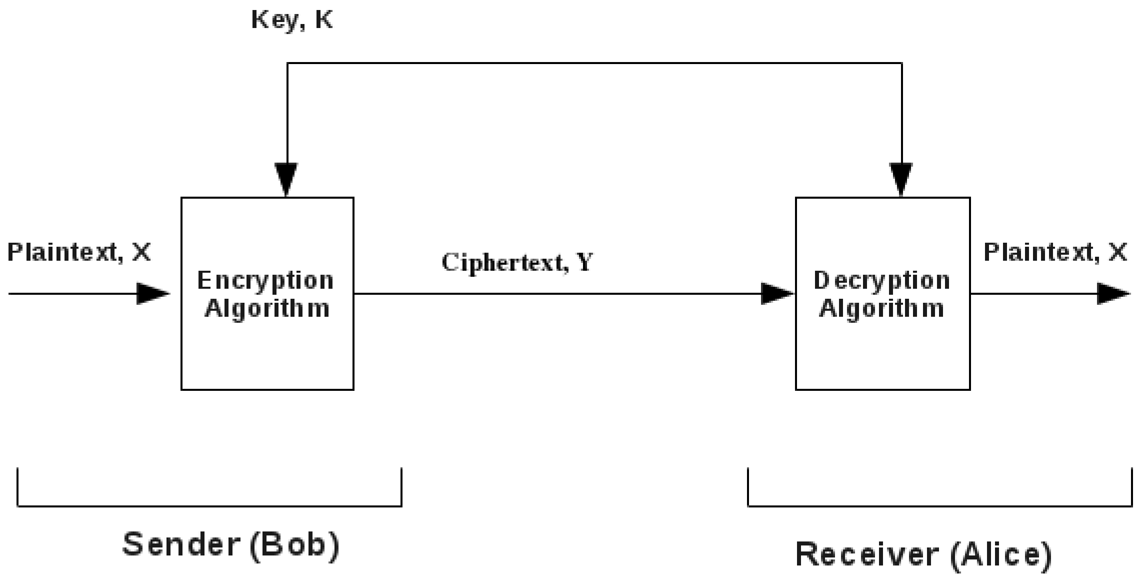
12 Message Encryption

Here the ciphertext itself is used for authentication.

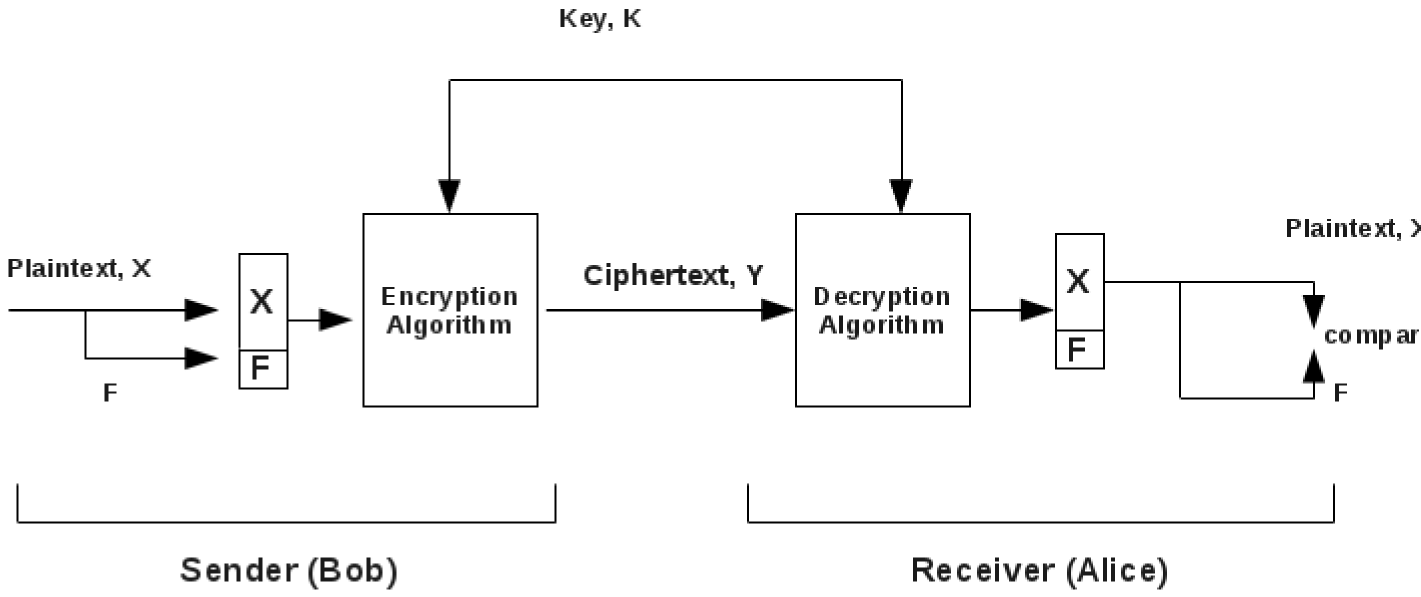
We will consider how this is done for symmetric encryption and public key encryption.

Authentication in Symmetric Encryption

We learned symmetric encryption technique as in the following figure.



This mechanism can be modified to include authentication as given below:



Here Bob prepares the plaintext, X. He calculates the checksum or some error correcting code, F from X. The code, F is appended to the end of X.

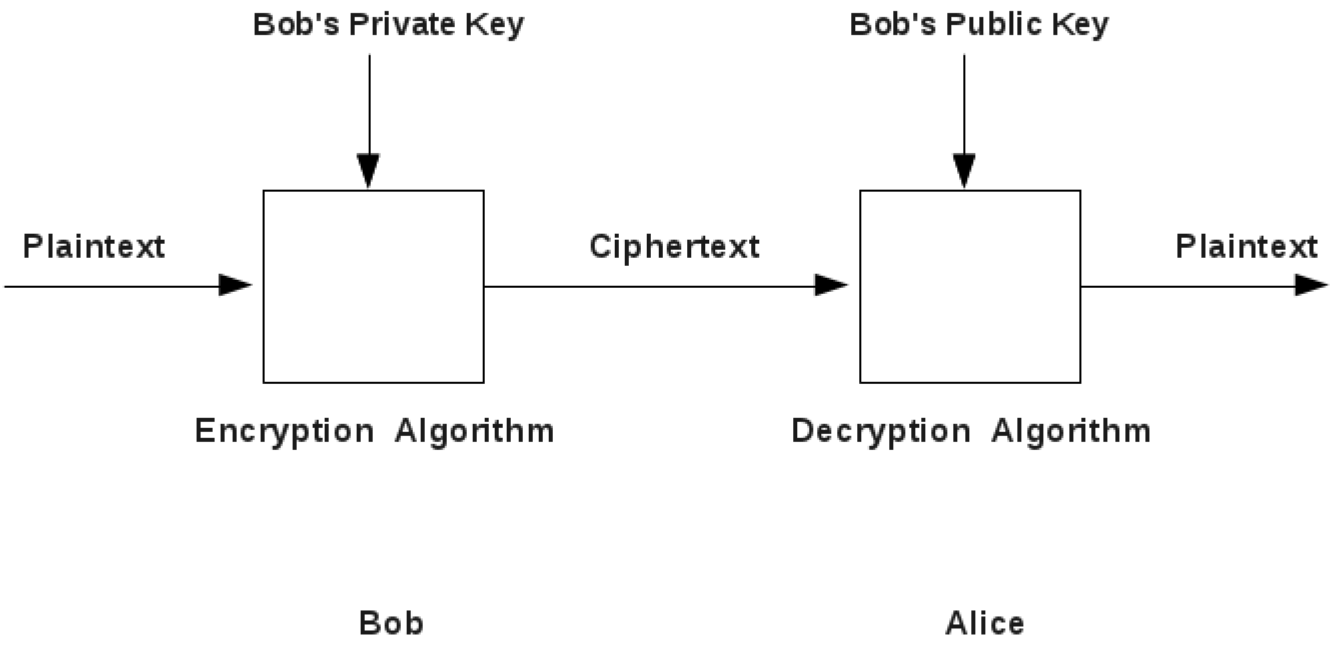
The combination X plus F is encrypted by Bob using the secret key, K. The resulting ciphertext is sent to Alice.

Alice uses the secret key, K to decrypt the message. She calculates the code from X. If the calculates code is same as F the message, X is considered authentic.

Authentication in Public Key Encryption

This we learned in the section on digital signatures.

Suppose Bob wants to send a text to Alice.. Then the communication is as follows:

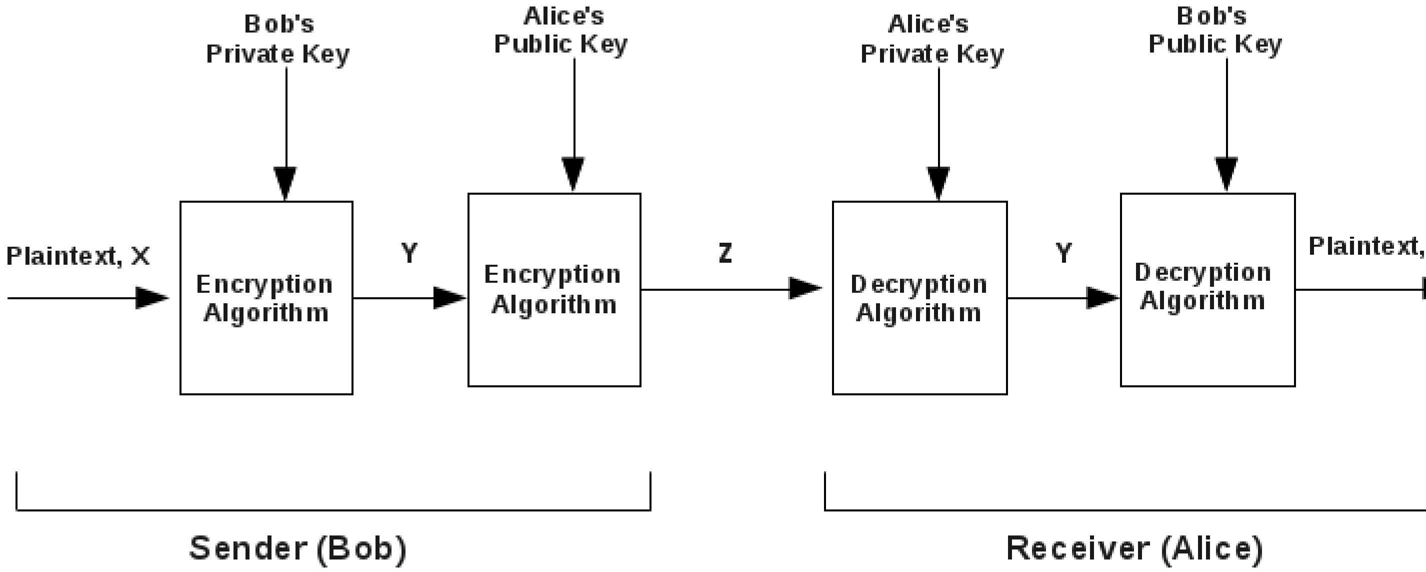


Then Bob encrypts the text with his private key. The encrypted text or ciphertext is the digital signature.

On getting the ciphertext, Alice decrypts it using Bob's public key. She will get the original plaintext.

In the above, the ciphertext can be decrypted by anyone who knows Bob's public key. It is to be made secure.

This is done as follows:



Here Bob encrypts the message (X) using his private key. The result (Y) is again encrypted using Alice's public key. The resulting ciphertext is Z.

The ciphertext, Z is sent to Alice.

Alice decrypts the ciphertext, Z using her private key. The resulting ciphertext (Y) is again decrypted using Bob's public key. The result is the original plain text, X.

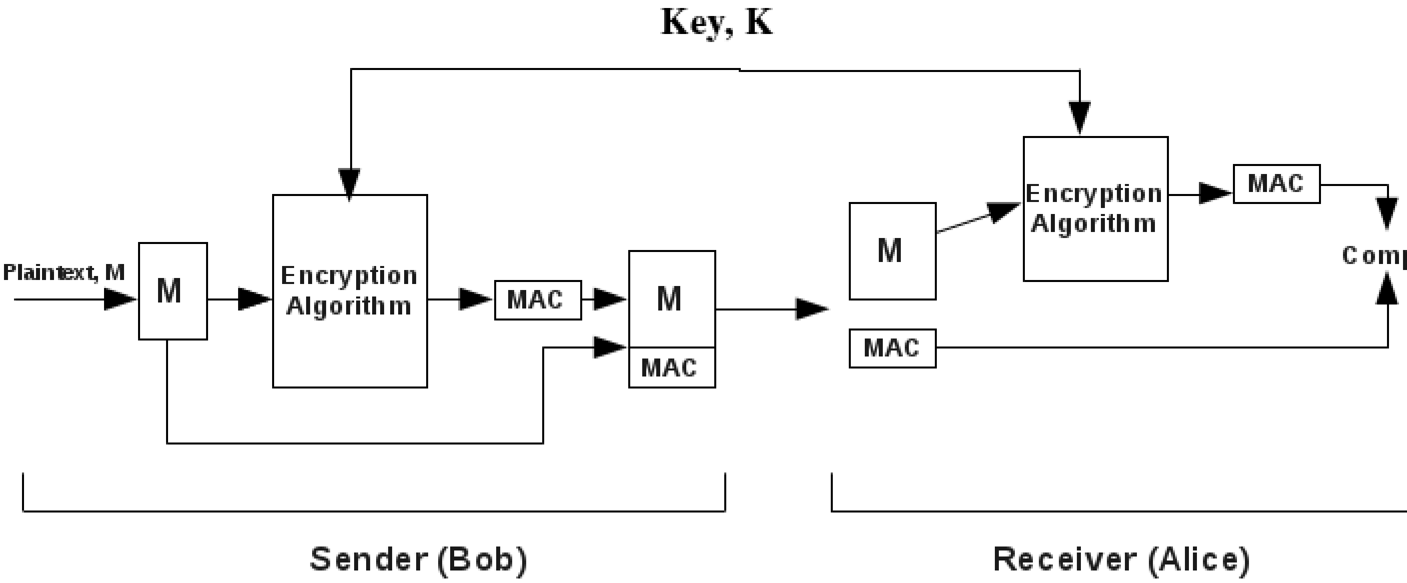
This scheme provides authentication and security.

13 Message Authentication Code (MAC)

This scheme uses a secret key to produce a small block of data. This block of data is called Message Authentication Code (MAC). This MAC is appended to the message and sent.

Let Bob wants to send a message to Alice.

Bob and Alice share a common secret key, K. Then this scheme works as shown below:



Bob calculates MAC by,

$MAC = C(K, M)$

where

M is the input message,

K is the shared secret key,

C is the MAC function which is an encryption function,

MAC is message authentication code.

The message, M plus MAC is transmitted to Alice.

When Alice gets this, she performs same calculations on the recieved message, using the secret key, K to generate a new MAC.

The received MAC is compared with the new MAC. If they are same, then it means the message is authentic.

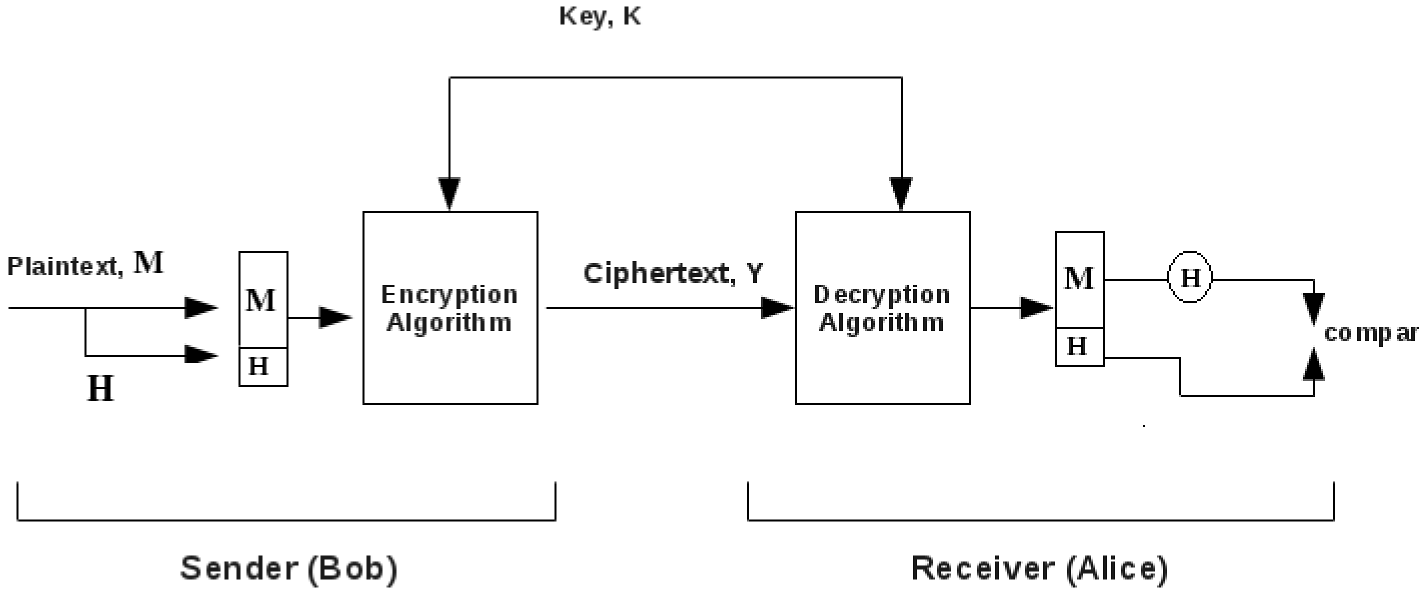
14 Hash Function

A hash function accepts a message, M as input and produces an output called as hash code, H(M).

Hash code does not use a key. Hash code is just a function of the input message. It provides error detection capability.

A commonly used algorithm for producing hash code is SHA (Secure Hash Algorithm).

One scheme for message authentication using hash code is given below:



A hash value, h generated from a hash function, H is of the form

$$h = H(M)$$

where M is a message.

This hash value, h is appended to the end of the message at the source.

The receiver authenticates the message by recomputing the hash value, h.

Requirements of a Hash Function

1. Hcan be applied to a block of data of any size.
2. H produces a fixed length output.
3. H(x) is relatively easy to compute for a given x.
4. For a given hash code, h, it is relatively easy to find x such that H(x) = h.
5. For a data block, x, it is computationally hard to find $y \neq x$ such that H(y) = H(x).
6. It is computationally hard to find a pair (x,y) such that H(x)=H(y).

Simple Hash Functions

A simple hash function is the bit by bit XOR operation.

This is expressed as follows:

$$C_i = b_{i1} \oplus b_{i2} \oplus b_{i3} \oplus .... \oplus b_{im}$$

where

C_i is i^{th} bit of hash code,

m is the number of blocks,

b_{ij} is i^{th} bit in j^{th} block,

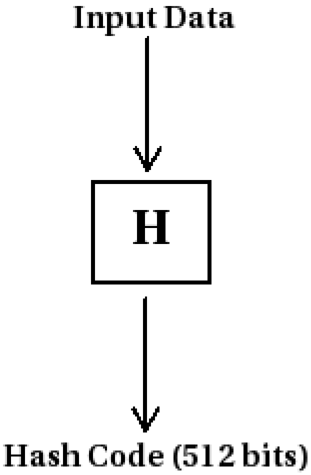
$\oplus$ is XOR operation.

14.1 SHA (Secure Hash Algorithm)

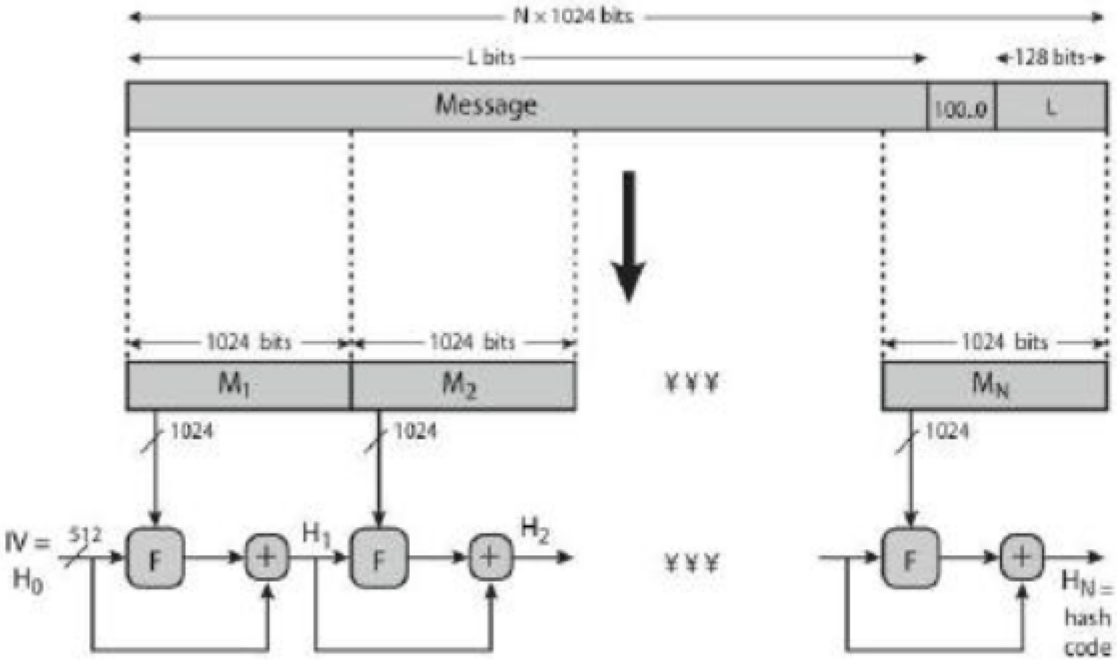
SHA is a popular algorithm for producing hash code. Different hash functions used in this algorithm are SHA-224, SHA-256, SHA-384, SHA-512.

Here we will learn SHA-512 hash function.

The algorithm takes as input a message and produces a 512 bit hash code.



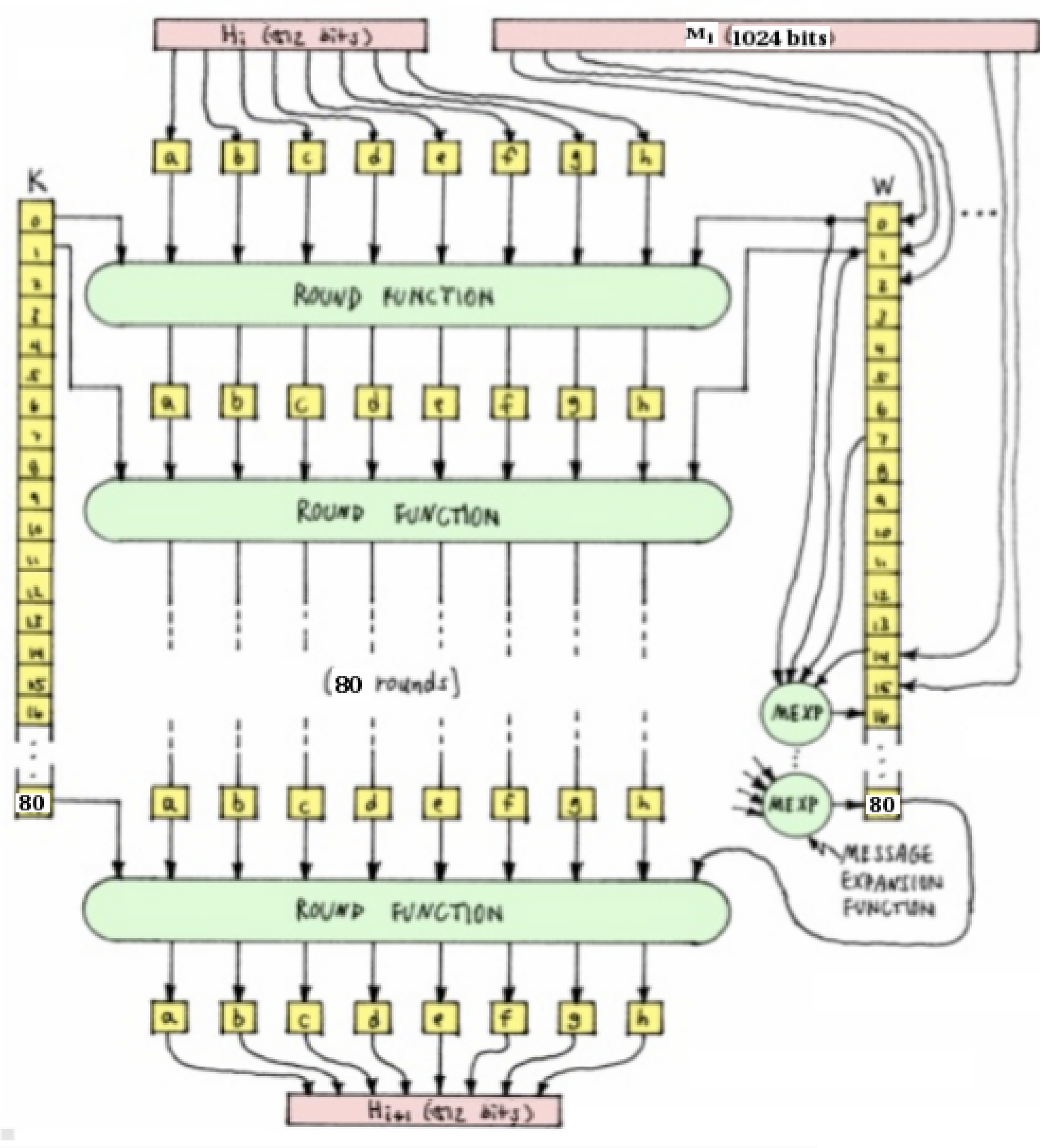
The process of hashing is shown below:



In the above, actual input data consists of L bits. It is padded with a number of bits; Then the length of the input data is added at the end as 128 bits. Now the resulting data should be a multiple of 1024 bits.

This data is divided into blocks of 1024 bits each. They are $M_1, M_2, M_3, ...M_N$.

The module shown as F performs the following as shown below:



In the above a,b,c,d,e,f,g,h are 8 registers. Each register is of size 64 bits. These are initialised with the following values:

a=6A09E667F3BCC908 e=510E527FADE682D1

b=BB67AE8584CAA73B f=9B05688C2B3E6C1F

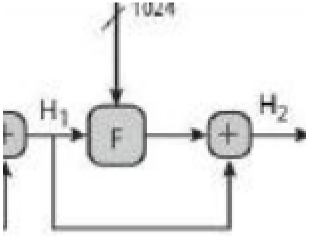
c=3C6EF372FE94F82B g=1F83D9ABFB41BD6B

d=A54FF53A5F1D36F1 h=5BE0CD19137E2179

The module shown as F is processed 80 times.

Input to the first round is the previous hash value H_{i-1}. Each round uses a 64 bit value W_t derived from the 1024 bit block M_i. Also a constant K_t is used. K_t is the first 64 bits of the fractional parts of the cube roots of first 80 prime numbers.

The output of the 80th round is added to the input of the first round H_{i-1} to produce $H_{i.}$



After all this, output is the 512 bit hash code.

Questions

MGU/May2012

1. Explain digital signature (4marks).

2. Write the need for encryption and decryption (4marks).

3a. Explain DES algorithm. Discuss the strength of DES algorithm.

OR

b. i)Explain RSA algorithm. Compare RSA algorithm with DES algorithm.

ii)Explain the importance of hash functions (12marks).

MGU/May2010

1. Explain about cryptography (4marks).

2. Mention the strength of DES (4marks).

3a. Discuss in detail about the following: i. DES algorithm ii. RSA algorithm.

OR

b. Write a technical note on digital signature (12marks).

MGU/Nov2009

1. What are the pitfalls in using RSA (4marks)?

2. What is polyalphabetic substitution? Explain (4marks).

3a. Describe Diffie- Hellman key exchange algorithm.

OR

b. Explain transposition and substitution ciphers in detail (12marks).

MGU/June 2009

1. What is cipher? Explain (4marks).

2. What is Caesar substitution (4marks)?

3a. Explain the requirements and different approaches for the digital signature function.

OR

b. Explain the important features of modern symmetric key algorithms (12marks).

MGU/May2008

1. Define encryption (4marks).

2. Define hash function (4marks).

3a. Explain DES algorithm in detail. Discuss the strength of DES.

OR

b. Explain RSA algorithm. Discuss the three possible approaches to attack RSA algorithm (12marks).

MGU/Nov2008

1. Define cryptography (4marks).

2. What is digital signature (4marks)?

3a. Explain the Diffie-Hellman key exchange method.

OR

b. Discuss the strength and weakness of DES algorithm (12marks).

MGU/July2007

1. What are ciphers (4marks)?

2. Define hash function (4marks).

3a. What is meant by symmetric key algorithm? Explain any one of them.

OR

b. Explain the strengths and disadvantages of DES algorithm (12marks).

MGU/Jan2007

1. What are ciphers (4marks)?

2. Explain use of random numbers in cryptography (4marks).

3a. What is the strength of DES algorithm over others?

OR

b. Explain cryptanalysis and substitution and Caeser cipher with suitable examples (12marks).

MGU/July2006

1. What are the criteria for S-boxes in the design of DES (4marks)?

2. Define public key and symmetrical keys (4marks).

3a. Explain the RSA algorithm.

OR

b. Explain message authentication and hash functions (12marks).

References

.

Stallings, W (2006). Cryptography and Network Security. Pearson Education.

Pfleeger, C, P; Pfleeger, S, L (2007). Security in Computing. Pearson Education.

website: http://sites.google.com/site/sjcetcssz