Cryptography

Professor Victor Buttigieg
Department of Communications and Computer Engineering

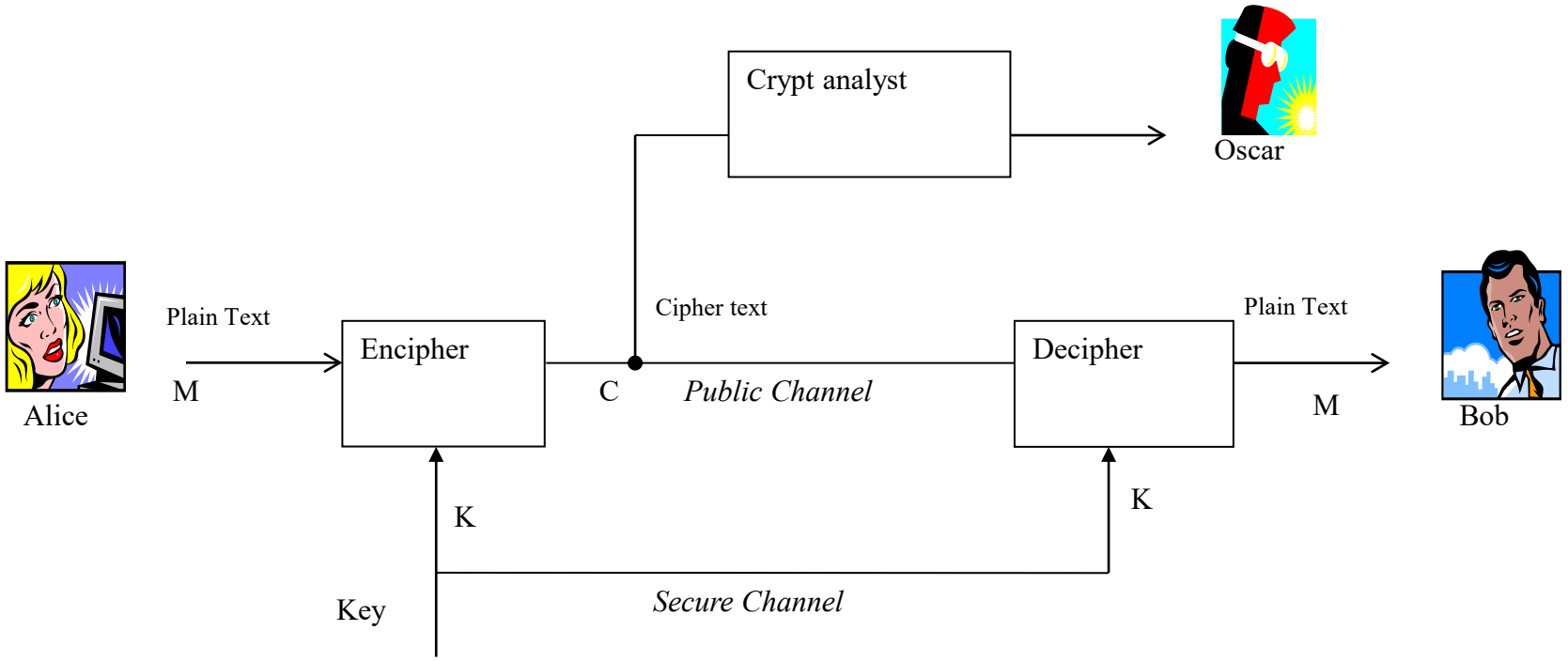
Cryptography

- The word cryptography is derived from the Greek secret writing.
- Some terminology
  - *Cryptography* is the study of ways of secret writing ➔ the development of cryptosystems.
  - *Cryptanalysis* is the study of ways to compromise secret writing ➔ the cracking of cryptosystems.
  - *Cryptology* is the amalgamation of cryptography and cryptanalysis.
  - *Cipher* is the encryption process.
  - *Cipher text* is the encrypted text.
  - *Plain text* is the original message.
- The main objectives of a cryptosystem are the following:
  - Privacy

and/or

  - Authentication (Digital Signatures)

Classical Cryptosystem



Attacks on Cryptosystems

- In modern cryptographic systems, it is assumed that all the details of the working of a cryptosystem is known to the cryptanalyst (Oscar) and that therefore all security resides in the secret key. This is known as Kerckhoff's Principle.

Cipher text only attack:

- Oscar is assumed to have available reasonable amounts of encrypted messages.

Known plain text attack:

- Oscar is assumed to have reasonable amounts of plaintext and corresponding cipher text.

Chosen plain text attack:

- Oscar may choose a reasonable number of plaintext and see their cipher texts

Chosen cipher text:

- Oscar has obtained temporary access to the decryption machinery hence he can choose the cipher text y and construct the corresponding plaintext x.

Types of Cryptosystems

There are two basic types of cryptosystems:

- *Block ciphers*
  - Plain text is segmented into chunks of n symbols and each block is encrypted independently of the previous ones. (Same block will always be encrypted in the same way). Example is DES (**D**ata **E**ncryption **S**tandard).
- *Stream Ciphers*
  - Each plaintext symbol m_i is encrypted with the ith element k_i of a sequence of symbols (key stream) generated with the key. E.g. One Time Pad or its approximates.

Average Time Required for Exhaustive Key Search

Key Size (bits)	*Number of Alternative Keys*	*Time required at 1 decryption/μs*	*Time required at 10^6 decryptions/μs*
32	$2^{32} = 4.3 \times 10^9$	2^{31} μs = 35.8 minutes	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	2^{55} μs = 1142 years	10.01 hours
128	$2^{128} = 3.4 \times 10^{38}$	2^{127} μs = 5.4×10^{24} years	5.4×10^{18} years

- Beware!
  - The encryption method may exhibit a weakness that can be exploited by the cryptanalysist, which can greatly reduce the time required to break the system.

The Caesar Cipher

- Each letter of the alphabet is replaced by the letter *n* places to the left, with the alphabet wrapped round at the end. The actual value used by Caesar was *n*=3.

ABCDEFGHIJKLMNOPQRSTUVWXYZ

DEFGHIJKLMNOPQRSTUVWXYZABC

- Thus HAIL CAESAR becomes KDLO FDHVDU.
- In this case there are only 26 different values for the key, so this system is very vulnerable to a brute force attack, where all possible keys are tested until a meaningful message is decrypted (this is known as exhaustive key search).
- If we consider the letters to be numbered, i.e. A=0, B=1, … Z=25, then the Caesar cipher may be defined by:

$$C = \mathrm{E}(M) = [M+3] \bmod 26$$

or in general

$$C = \mathrm{E}_K(M) = [M+k] \bmod 26$$

Simple (Monoalphabetic) Substitution

- This is similar to the Caesar cipher in that a letter is substituted by any other (not in sequence).
- It is easy to see that the number of different keys is 26! (= $4x10^{26}$) which is much larger than that of the Caesar cipher.
- In this case an exhaustive search is not plausible, however this system may be easily broken by letter frequency analysis since there is redundancy in the source.

ABCDEFGHIJKLMNOPQRSTUVWXYZ

NDHEIJBLFMORWGSATKQVXUYPZC

THIS IS A TEST

becomes

VLFQ FQ N VIQV

Sample Cryptogram (from Beker and Piper, Exercise 1.10)

XNKWBMOW KWH JKXKRJKRZJ RA KWRJ
ZWXCKHI XIH IHNRXYNH EBI THZRCWHIRAO
DHJJXOHJ JHAK RA HAONRJW KWH IHXTHI
JWBMNT ABK EBIOHK KWXK KWH JKXKRJKRZJ
EBI XABKWHI NXAOMXOH XIH GMRKH NRLHNU
KB YH TREEHIHAK WBQHPHI HGMRPXNHAK
JKXKRJKRZJ XIH XPXRNXYNH EBI BKWHI
NXAOMXOHJ RE KWH ZIUCKXAXNUJK TBHJ
ABK LABQ KWH NXAOMXOH RA QWRZW KWH
DHJJXOH QXJ QIRKKHA KWHA BAH BE WRJ
ERIJK CIBYNHDJ RJ KB KIU KB THKHIDRAH
RK KWRJ RJ X TREERZMNK CIBYNHD.

Cryptanalysis of Simple Substitution Cipher

E	120	D	41	G	14
T	90	L	36	B	13
O	82	C	29	V	10
A	78	U	29	K	6
N	73	F	28	X	3
I	68	M	26	J	3
R	66	P	21	Q	2
S	65	Y	18	Z	2
H	59	W	18		

Letter frequencies per 1000 in English (adapted from Davies and Price)

Groupings
Group 1: E
Group 2: T,O,A,N,I,R,S,H
Group 3: D,L
Group 4: C,U,F,M,P,Y,W,G,B
Group 5: V,K,X,J,Q.Z

Letter Frequency Example

H	47	Z	8
K	43	T	7
R	31	C	5
J	28	D	5
X	28	Q	5
I	23	Y	5
W	22	U	4
A	20	P	3
B	20	G	2
N	17	L	2
O	12	F	0
E	11	S	0
M	8	V	0

Vigenère Cipher

- A number of shifted alphabets (similar to the Caesar cipher) are used defined by a key word of *d* letters. The shifted alphabets are used alternately e.g. using the keyword ***DSL***

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
EFGHIJKLMNOPQRSTUVWXYZABCD
TUVWXYZABCDEFGHIJKLMNOPQRS
MNOPQRSTUVWXYZABCDEFGHIJKL
```

- Thus the phrase ABSOLUTE SECURITY is encrypted as EUESEGXX DIVGGVBFC.
- The number of keys in this case is 26^d.
- Ciphers like this which use a number of different alphabets are called polyalphabetic.
- Variations of the Vigenère cryptosystems include ones where the shift of the next alphabet depends on the plaintext or cipher text itself.
- By today's standards this method is not very secure.

Simple Transposition Cipher

- In the simple transposition cipher of order *n*, the plain text is divided into groups of successive *n* letters and each group is re-ordered in a manner defined by the key.
- For example a transposition of order 5 defined by the key 41523 where each number represents the original position of a letter, may be used to encrypt the message

```
12345123 45123451    41523415 2341523…
ABSOLUTE SECURITY as OALBSSUE TEICTUR…
```

- The number of keys in this case is *n*!

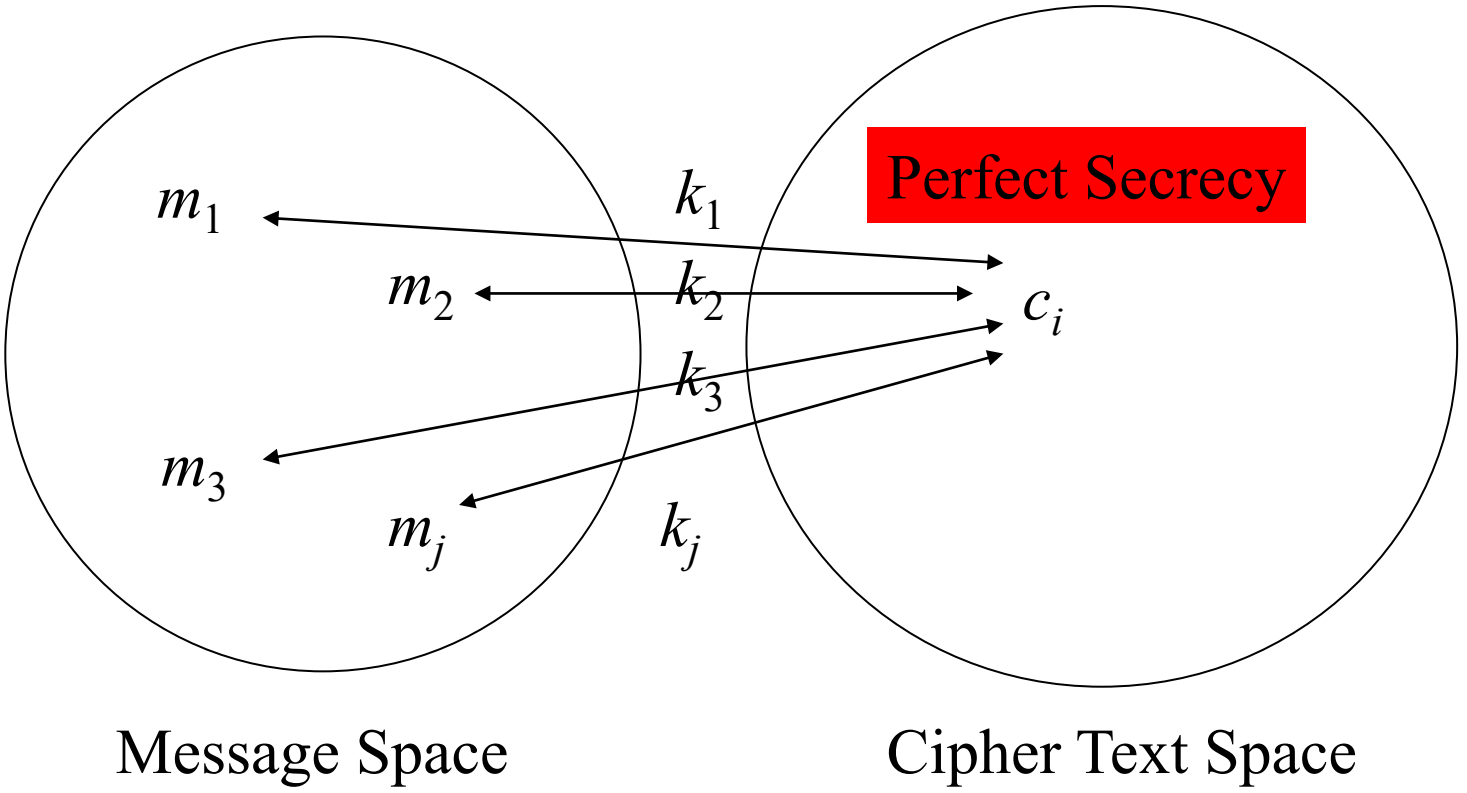
Information Theory and Cryptography

- Shannon's theory of cryptography was first published in 1949 and forms the basis of all theoretical work on cryptography until the introduction of public key cryptography in 1975.
- Here we are not concerned with the resources that the eavesdropping cryptanalyst has but only with whether he has sufficient information available to find the plain text message and/or the key.
- It is assumed that each key is used only once to encipher a single message, which could be, however, of variable length.

Perfect Secrecy

- Shannon defined perfect secrecy if the message and cipher text are statistically independent, i.e.
  $P(M|C) = P(M)$ or equivalently $P(C|M) = P(C)$.
- Alternatively $H(M|C) = H(M)$
  - no information is gained from the cipher text.
- Theorem (Shannon):
  - For prefect secrecy, there must be as many different keys as possible messages.
- *Proof:* Let C be some cipher text with non-zero probability $P(C)$. In order to satisfy the requirements of perfect secrecy, for each possible message m_j, there must be a key which maps m_j to C. But these keys must all be different since if the same key maps different messages to the same cipher text, then decryption is not possible.

Prefect Secrecy - Proof



One Time Pad

- It turns out that we can actually achieve perfect secrecy meeting the bound that the key length equals the message length. In this case

$$c_j = m_j + k_j \bmod 26$$

- where k_j is the jth symbol in the key stream and is independent of all previous or future key symbols.
- Since k_j is completely random, then it is clear that c_j is random as well and that

$$P(c_j|m_j) = P(c_j).$$

- Decryption is easy once we know all the k_j's.
- The problem with the one time pad is that of Key management, however, it is used in practice, in areas where security is vital.
- It is the typical crypto system used by spies.

Unicity Distance

- The true (entropy) rate of a language, r, is defined as the average number of information bits contained in each character and is expressed for a message of length N by

$$r = \frac{H(X)}{N}$$

  where $H(X)$ is the message entropy.
- For English (and large N), $r \approx 1.0$ to 1.5 bits/symbol.
- The absolute r' is defined by

$$r' = \log_2 L$$

  where L is the number of symbols in the language. For English, r' = $\log_2 26$ = 4.7 bits/symbol.
- The redundancy D of a language is defined as $D = r' - r$

- (For English $D = 3.2$ bits/symbol)
- The **Unicity distance** U is defined as the average amount of plaintext symbols whose cipher text must be seen so that there is only one possible key and plaintext. $\therefore$ For a perfect secrecy system, $U = \infty$.
- In general, an exact value for U is hard to find for a particular cryptosystem. Using the notion of a random cipher, Shannon estimated that

$$U = \frac{H(K)}{D}$$

- If we assume that the keys are chosen randomly from the key space K, then $H(K)$ is the number of bits in the key, where

$$H(K) = \log_2 |K|$$

- The Unicity distance is just an indication of the effectiveness of a cryptosystem, since this is independent of the system itself (depends only on $\log_2 |K|$ and D).

- For instance, for the simple substitution cipher

$$H(K) = \log_2 26! = 88.4 \text{ bits/symbol}$$

$$\Rightarrow U = \frac{88.4}{3.2} = 27.6$$

- It is well known that cryptograms of this order can be solved.
- However, for **DES**, which is much better than the above cipher, $U = 8.6$.
- So U is not a very good measurement after all.
- The only thing that we can say is that having a cryptosystem with large unicity distance gives theoretical assurance of security.
- However, a small Unicity distance does not necessarily indicate a weak cryptosystem.
- The Unicity distance also indicates that if we remove the redundancy from the source, the cipher text is harder to break.
- If the source-encoded data contains no redundancy, then $U = \infty$, and the cryptosystem may be called ideal, since any amount of cipher text does not help to deduce the key.

Homophonic Coding

Homophonic coding uses this idea by reducing the redundancy in the source by adding '*homophones*' to the source so that the source statistics appear more flat.

Consider a two-symbol source with

$$p(a) = \tfrac{3}{4} = 0.75 = 0.11$$
$$p(b) = \tfrac{1}{4} = 0.25 = 0.01$$

Suppose that a is allocated homophones (or copies) a_1 and a_2, with probability $p(a_1) = \tfrac{1}{2}$, $p(a_2) = \tfrac{1}{4}$ when a is emitted from the source, and b has a single homophone b_1, $p(b_1) = \tfrac{1}{4}$.

Hence

$p(a_1) = ½$	0
$p(a_2) = ¼$	10
$p(b_1) = ¼$	11

Notice that the homophones' probability are all in the form $1/2^n$.

Therefore a Huffman (or Shannon) code may be used to remove the redundancy. In this case D = 0 and U = ∞.

The coded message may now be encrypted using a simple substitution cipher on all binary strings of a certain length. An ideal cipher will be achieved in this case.

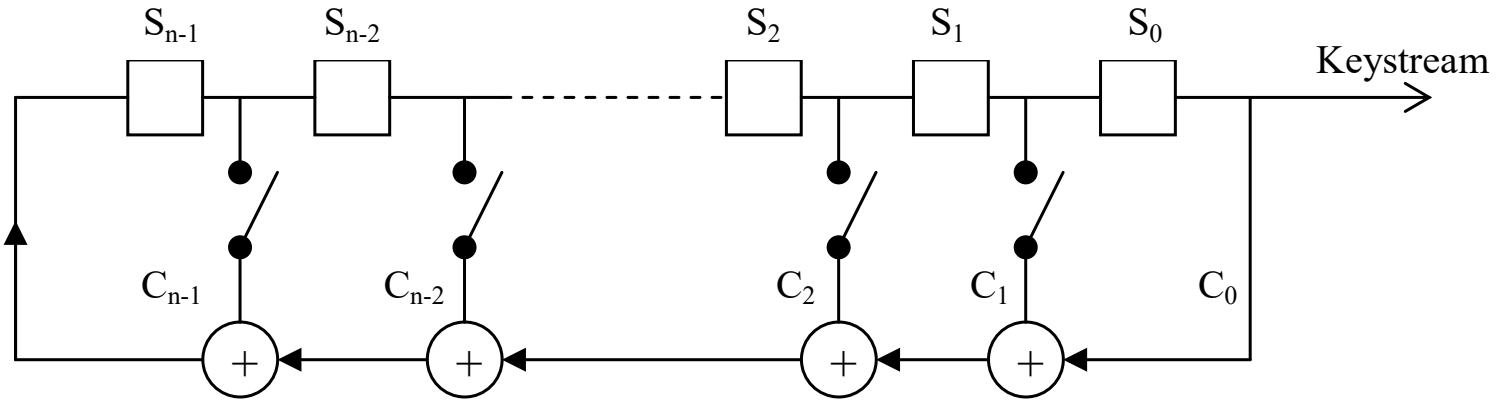
- The general procedure is as follows:
  - Expand each symbol probability as a binary fraction.
  - Allocate one homophone for each 1 in the expansion, with the corresponding probability.
  - Encode the set of homophones using perfect compression (with Huffman or Shannon Code).
  - To encode a source symbol choose one homophone according to its probability.
- In general the binary expansion of the probability would yield an infinite number of homophones. In practice only a finite number is chosen (neglecting those with probability less than some threshold value). This results in $D \neq 0$.
- In addition natural languages are much more complex than a simple first order model and are very difficult to model. Hence this method doesn't work really well. (Note that here the source is expanded, even though $D = 0$, since redundancy is removed by adding randomness)

Diffusion and Confusion

- Shannon realised that it was impractical to realise perfect secrecy or even ideal ciphers. So instead he came up with two methods how to make the life of the cryptanalyst a lot harder.
  - *Diffusion*: The source statistics is averaged out amongst a number of cipher text symbols.
    - Therefore in a block cipher, each bit in the output block should depend on each input bit to the block.
  - *Confusion*: Making the relationship between the key and cipher text complicated.
    - Therefore in a block cipher, each bit of the output block should be dependent on each bit of the key.

Stream Ciphers from LFSR's

- A pseudo random sequence may easily be generated using Linear Feedback Shift Registers (LFSR).



- The key consists of the initial state of the registers S_0, S_1, …. S_{n-1} together with the state of the feedback taps (C_0, C_1, …. C_{n-1}). Usually $C_0 = 1$. Therefore the key consists of $2n$-1 bits.

Stream Cipher Implementation

- If the feedback taps form a primitive polynomial, then the period of the generated sequence would be 2^n-1.
- Unfortunately this system is vulnerable to a known plain text attack. (Cryptosystem could be broken by solving n linear equations in *n* unknowns).
- However this system is still used in practice by introducing some non-linearity.
- Note that if the pseudo-random sequence is truly random, this would have been the same as the one time pad.

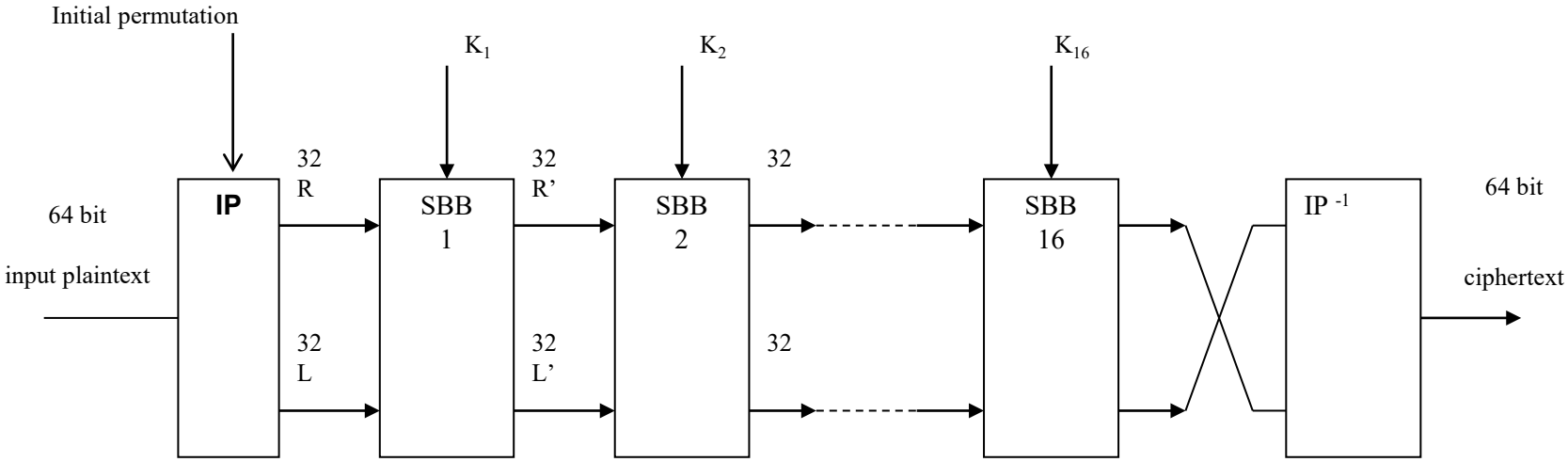
Block Ciphers

- In block ciphers, both diffusion and confusion are employed. Here *n*-bits of plain text are encrypted using a key into (usually) an *n*-bit block of cipher text.
- One problem with block cipher as compared with a stream cipher is that of error propagation. Should an error occur in a block, the whole block is lost, whereas in a stream cipher, the error would be localized.

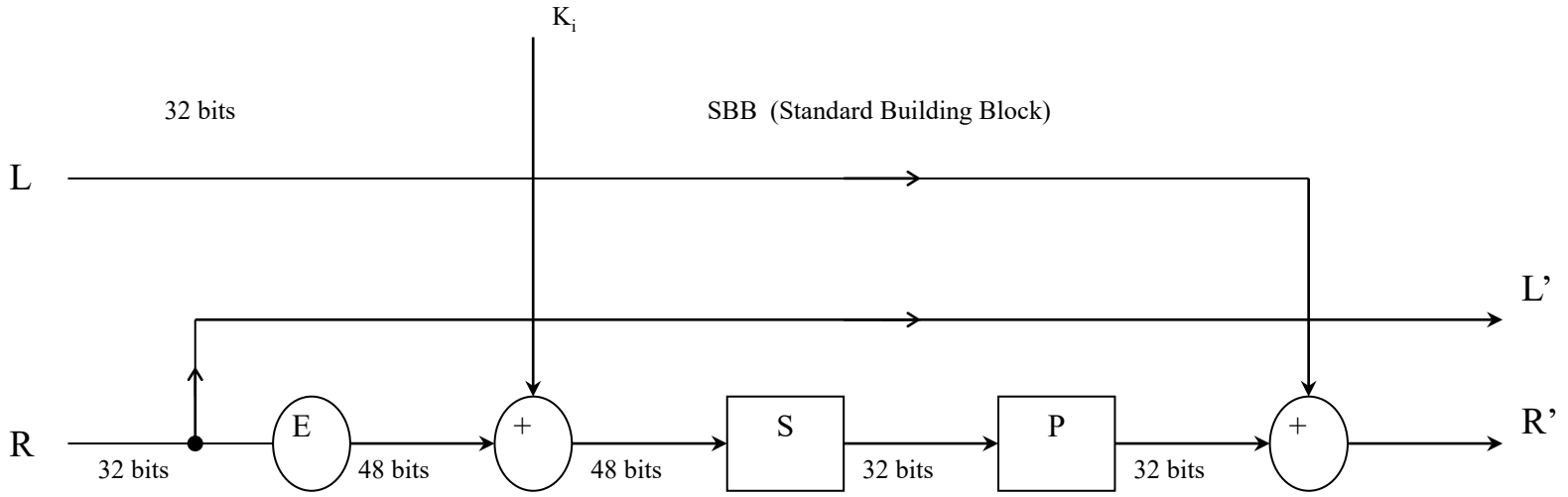
DES

- A well known example of a block cipher is the **Data Encryption Standard** (DES) adopted by the National Bureau of Standards (NBS) in January 1977, based on the Lucifer cipher developed by IBM in the early 70's.
- DES consists of 64 bit blocks and a 64 bit key, of which only 56 bits are effectively used, since every 7 bits there is a parity bit.

DES Structure



- 16 sub keys K_1, K_2, …, K_{16} are deterministically generated from the key K, each sub key being 48-bits wide and a sub string of the 56-bit key.
- The initial permutation IP and its inverse IP^{-1} are of no encryption significance.
- The permuted plaintext is split into a left block of 32 bits and a right block of 32bits. The right block becomes the left block input of the next *Standard Building Block* (SBB). The right block R is diffused and extended using E to make it 48 bits.



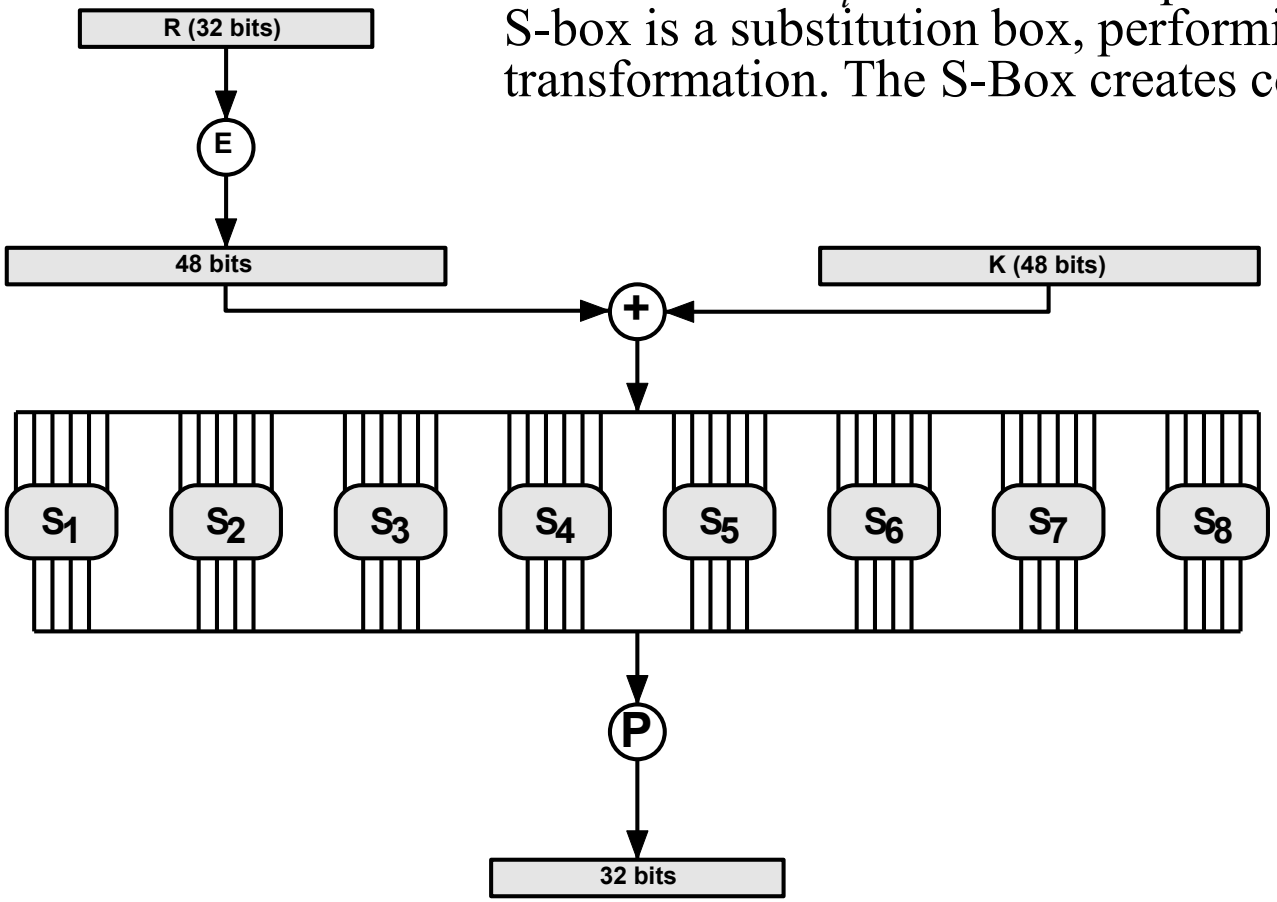
Expansion / Permutation Box (E)

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

- This expands the number of bits from 32 to 48 by repeating the bits shown shaded in yellow.
- The other bits are permuted, reading from left to right, top to bottom.

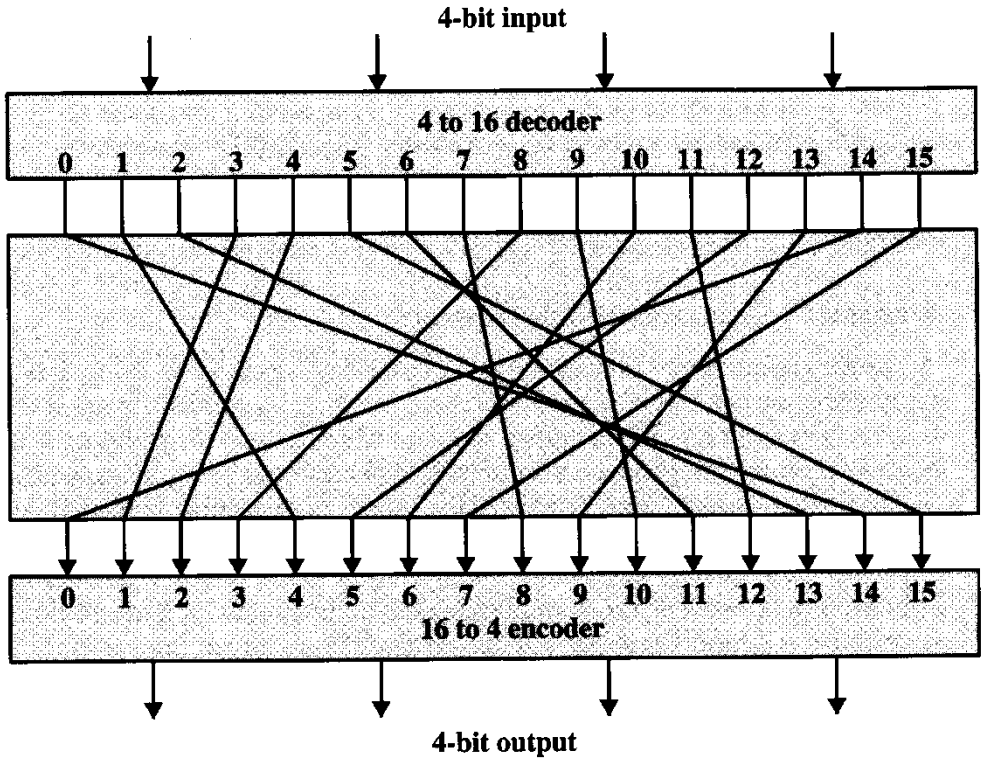
S-Box (S)

- The output from the expansion box is then modulo-2 summed with K_i and used as input to the S-Boxes. An S-box is a substitution box, performing a non-linear transformation. The S-Box creates confusion.



S-Boxes

S_1															
	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13



- The first and last bit of the input to the S-Box define the row number to use.
- The middle 4 bits define the column number to use.
- Example: the input 011000 indicates row 0 and column 12. So the output is 5 (0101).

Definition of S-Boxes

S_1

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

S_2

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

S_3

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

S_4

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

S_5

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

S_6

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

S_7

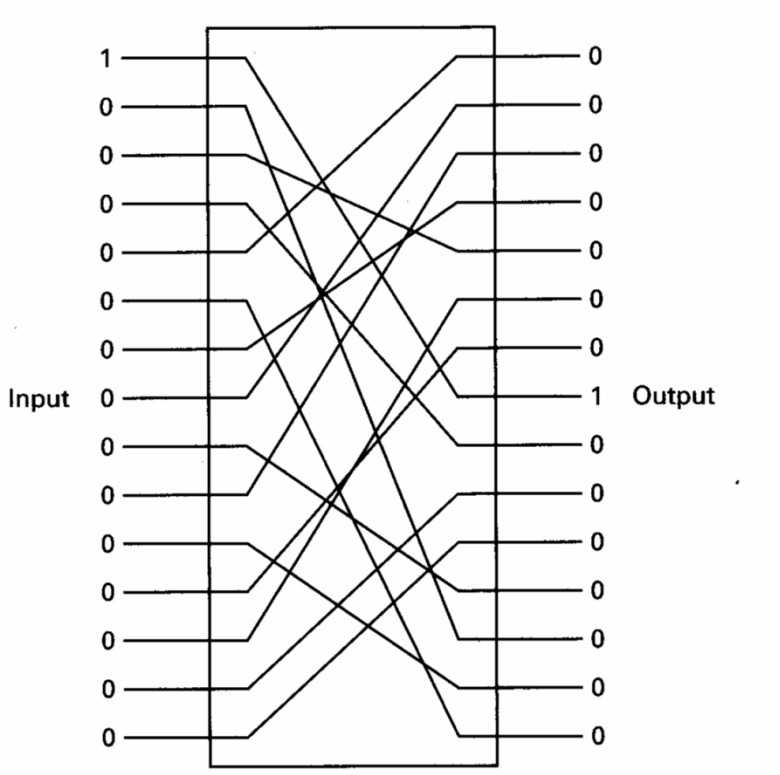
4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S_8

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

Permutation Box

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



- The output of the S-Box is then permuted using the P-Box (Diffusion).
- The input sequence $x_1x_2\ldots x_{32}$ is permuted as $x_{16}x_7\ldots x_{25}$.
- The permutation box performs diffusion.
- The S-Box performs confusion

Avalanche Effect

- The DES algorithm exhibits the desirable property of the Avalanche Effect
- This is a direct result of diffusion and confusion
- A single bit change in the plaintext results in many bit changes in the ciphertext
- A single bit change in the key results in many bit changes in the ciphertext

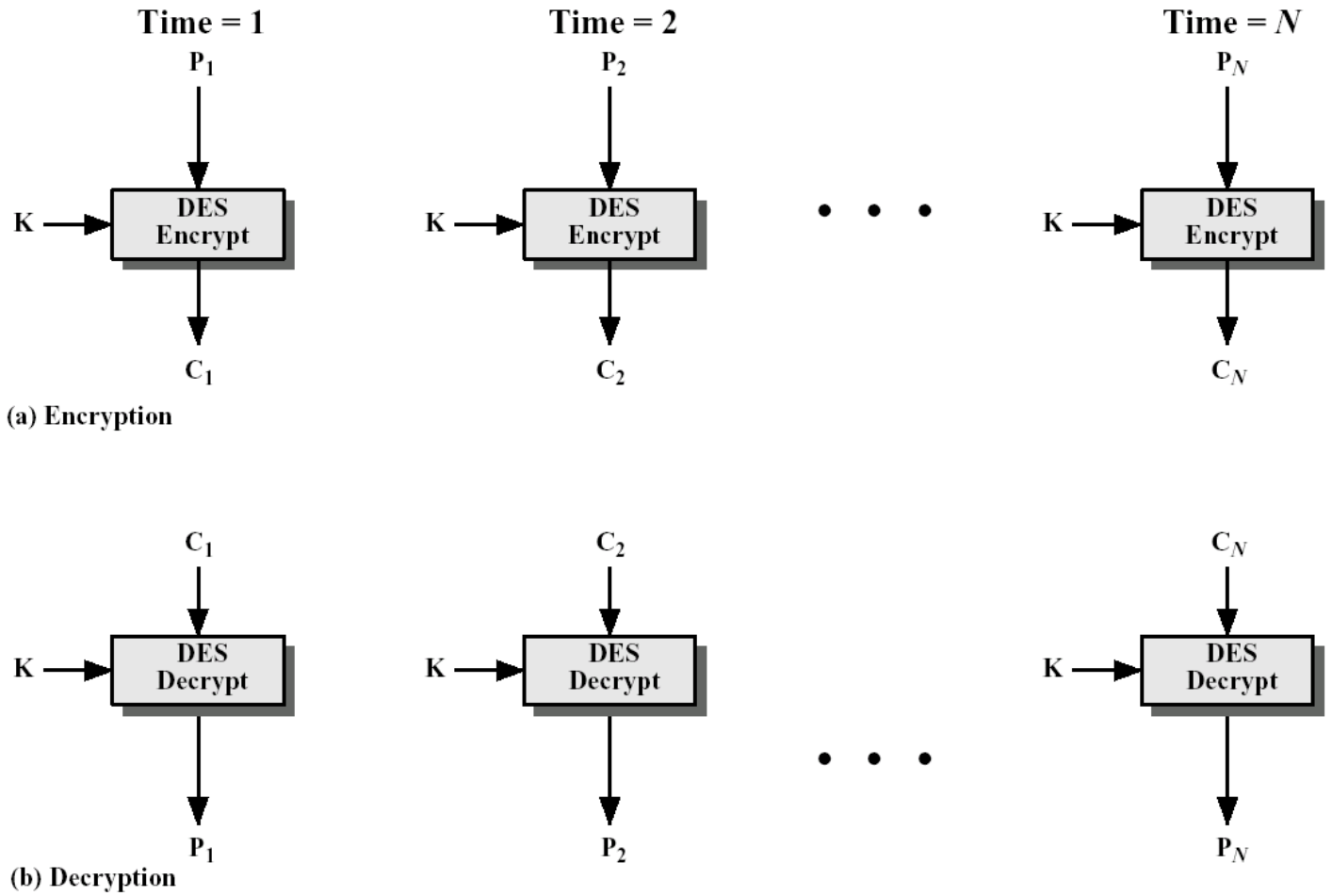
Decryption

- To decrypt, the same algorithm is used but the key sequence that is used in the standard building block (SBB) is taken in the reverse order.

Block Cipher Modes of Operation

- Electronic Codebook (ECB)
  - Typically used for secure transmission of single values
- Cipher Block Chaining (CBC)
  - General-purpose block oriented transmission
- Cipher Feedback (CFB)
  - General-purpose stream-oriented transmission
- Output Feedback (OFB)
  - Stream-oriented transmission over noisy channels

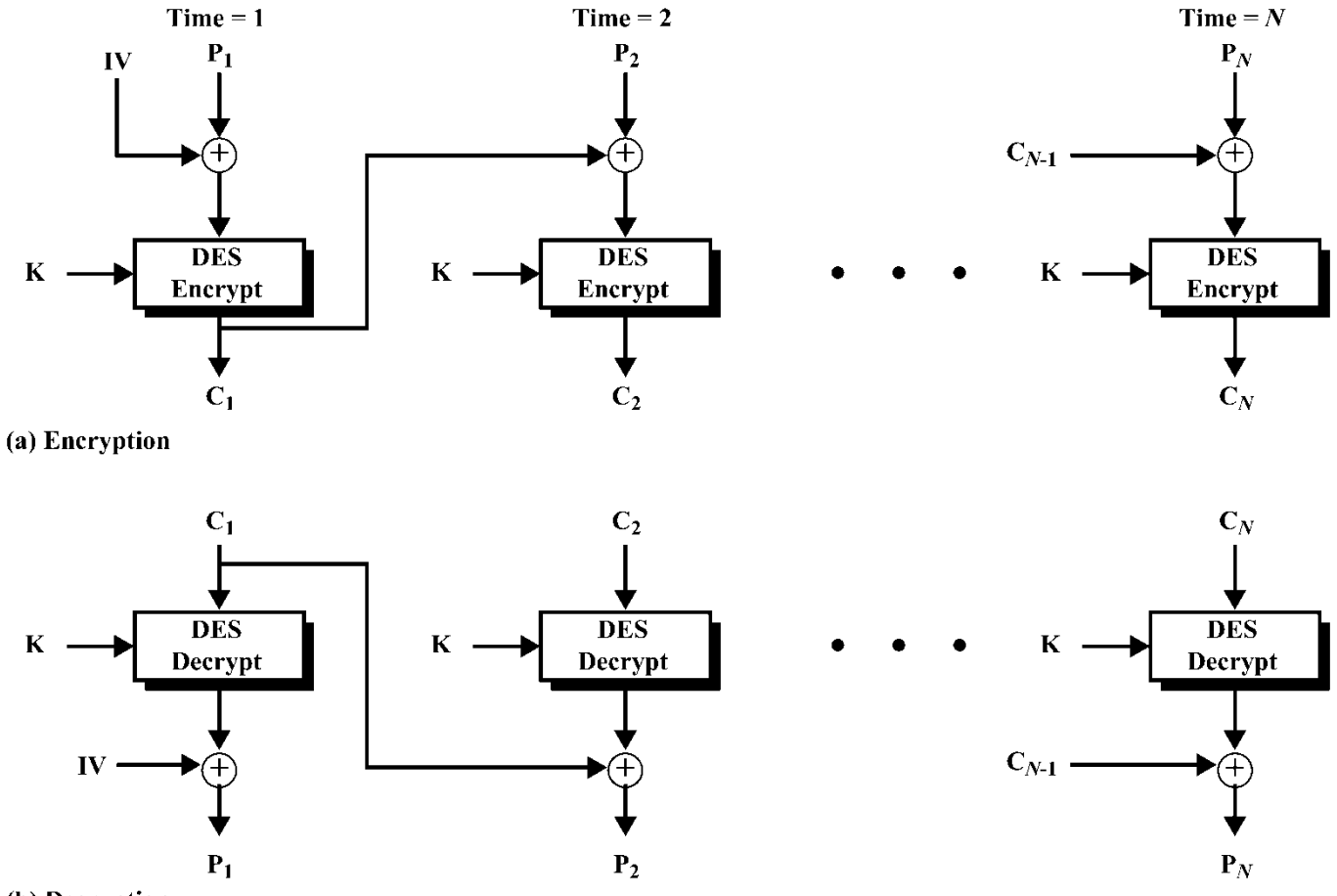
Electronic Codebook Mode



ECB Characteristics

- Ideal for short amount of data, such as the encryption of a DES key
- The same 64-bit block of plaintext will give the same ciphertext
- Vulnerable to attack if the message is long and there are repetitive 64-bit blocks (consider for instance using 8-bit ASCII)

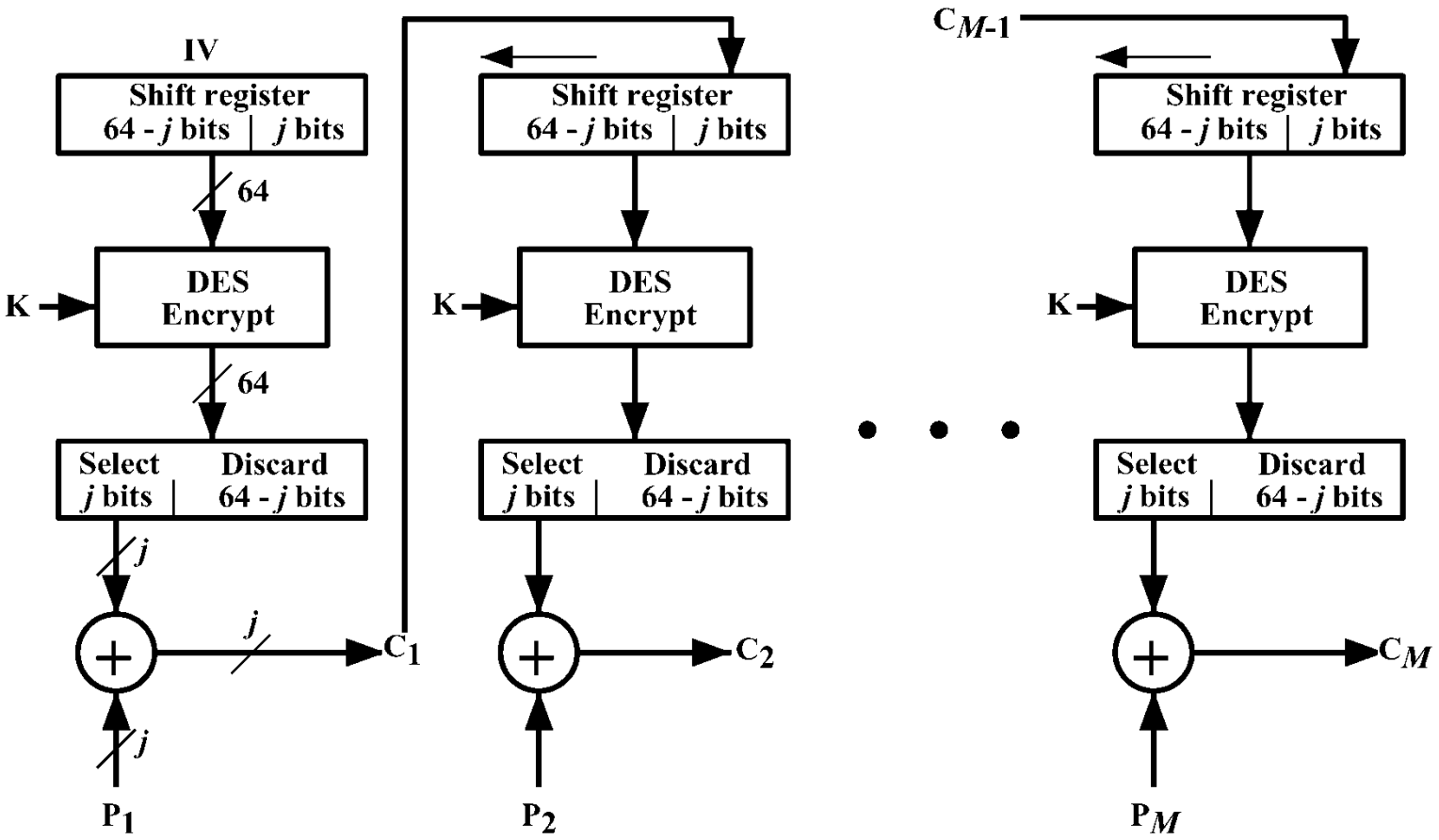
Cipher Block Chaining Mode



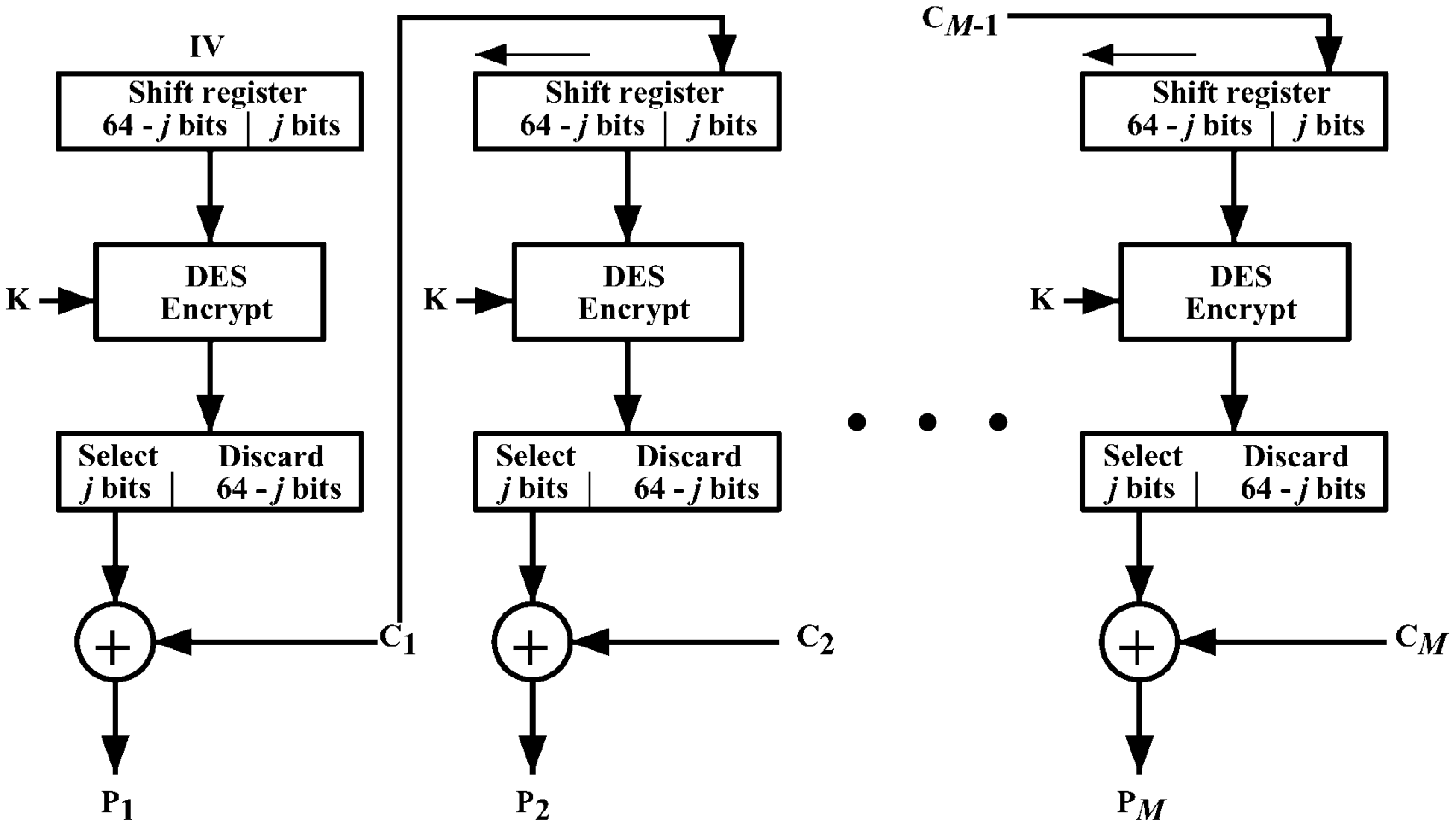
CBC Characteristics

- Same plaintext 64-bit blocks produce different ciphertext
- Can be used to encrypt long messages
- The initialisation vector (IV) must be known to both the sender and the recipient.
- The IV must be protected as well as the key, since it can be used to mount a known plaintext attack
- Problems with error propagation

J-Bit Cipher Feedback Mode



(a) Encryption

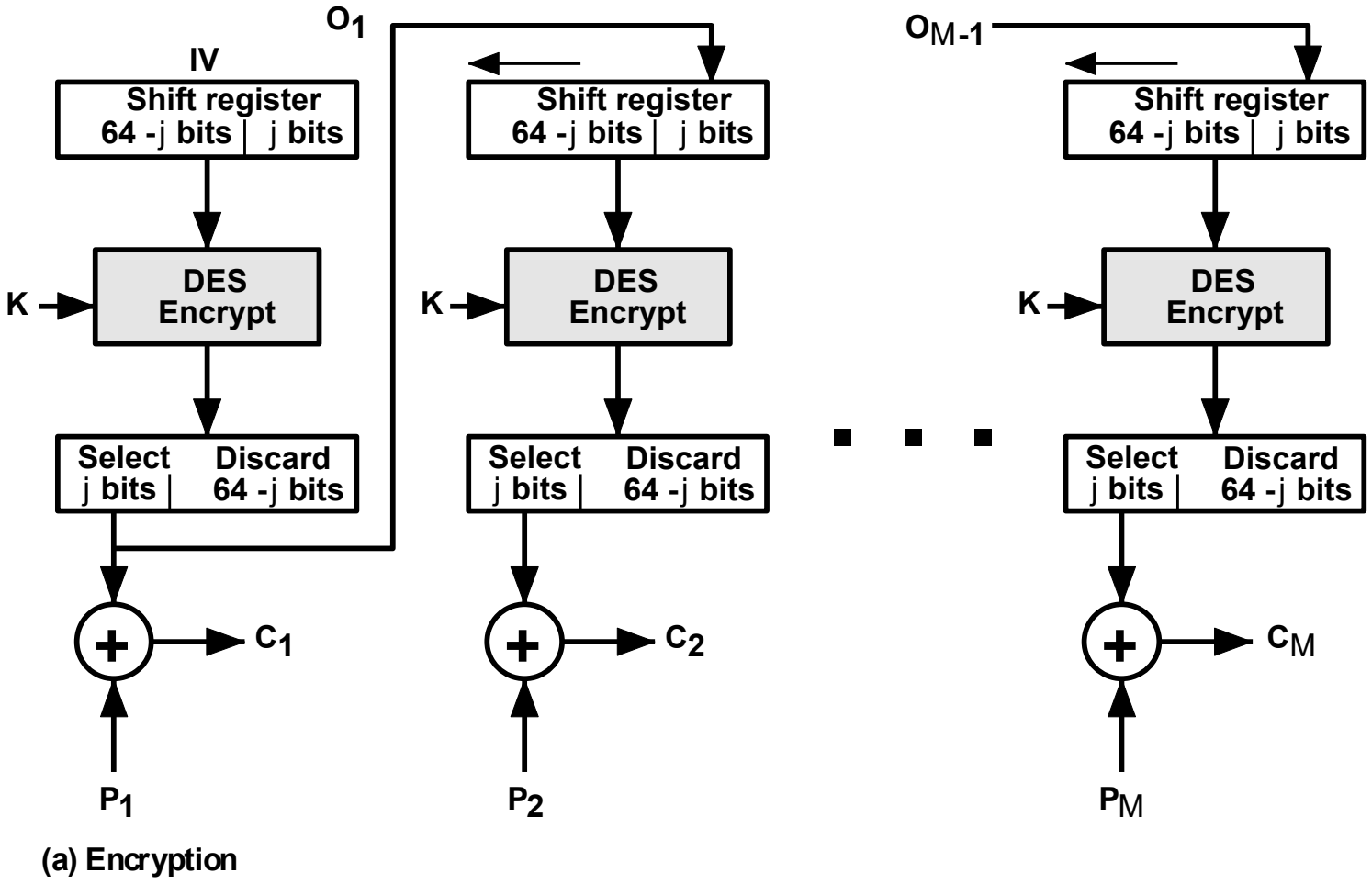


(b) Decryption

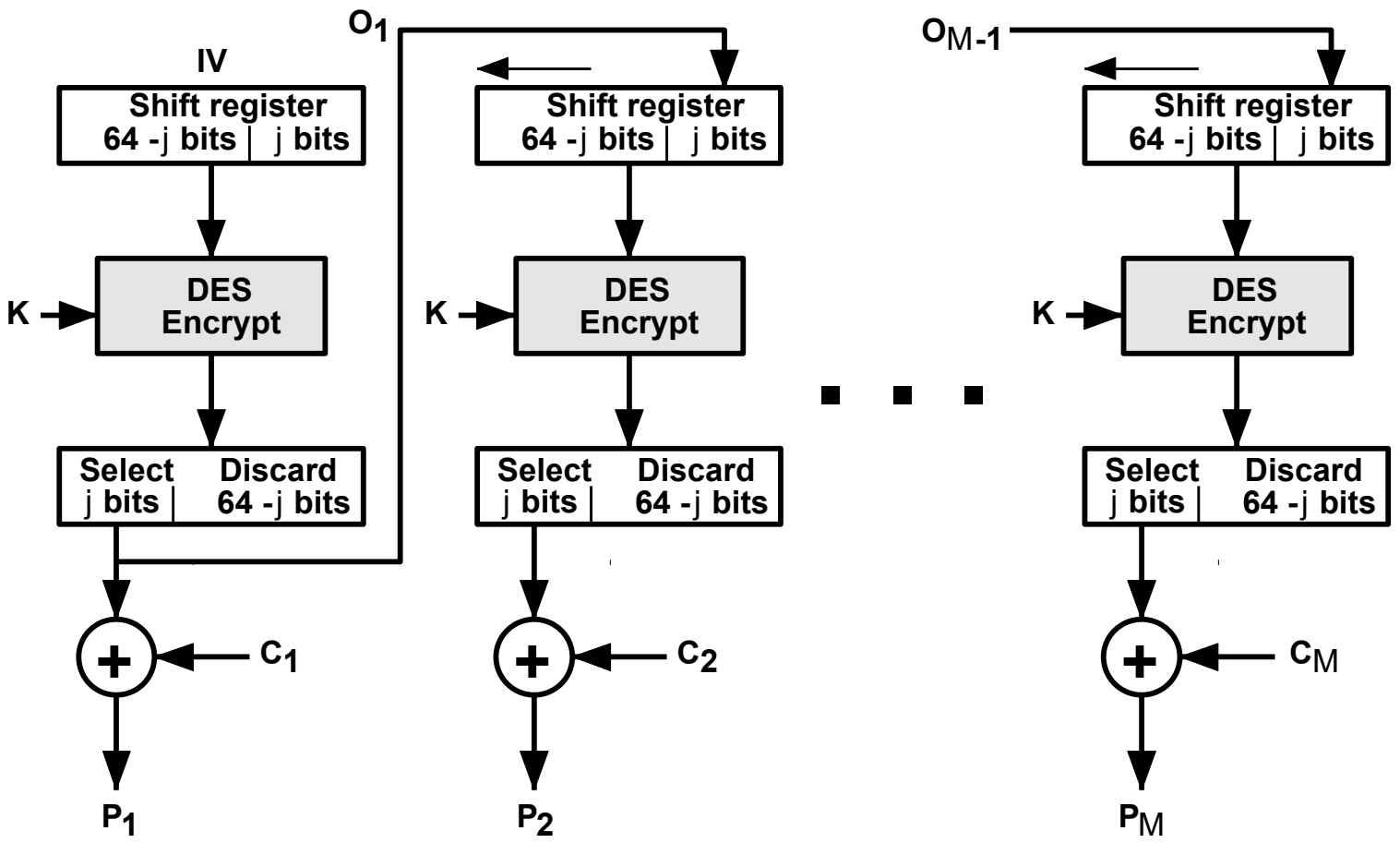
CFB Characteristics

- In this mode DES operates like a stream cipher, encoding *j*-bits at a time.
- Ciphertext is immediately available
- No need to pad last block with dummy bits
- Can be used to encode data continuously
- Since decoding of subsequent *j*-bit blocks depends on previous ciphertext, errors on the channel will result in error propagation

Output Feedback Mode



(a) Encryption



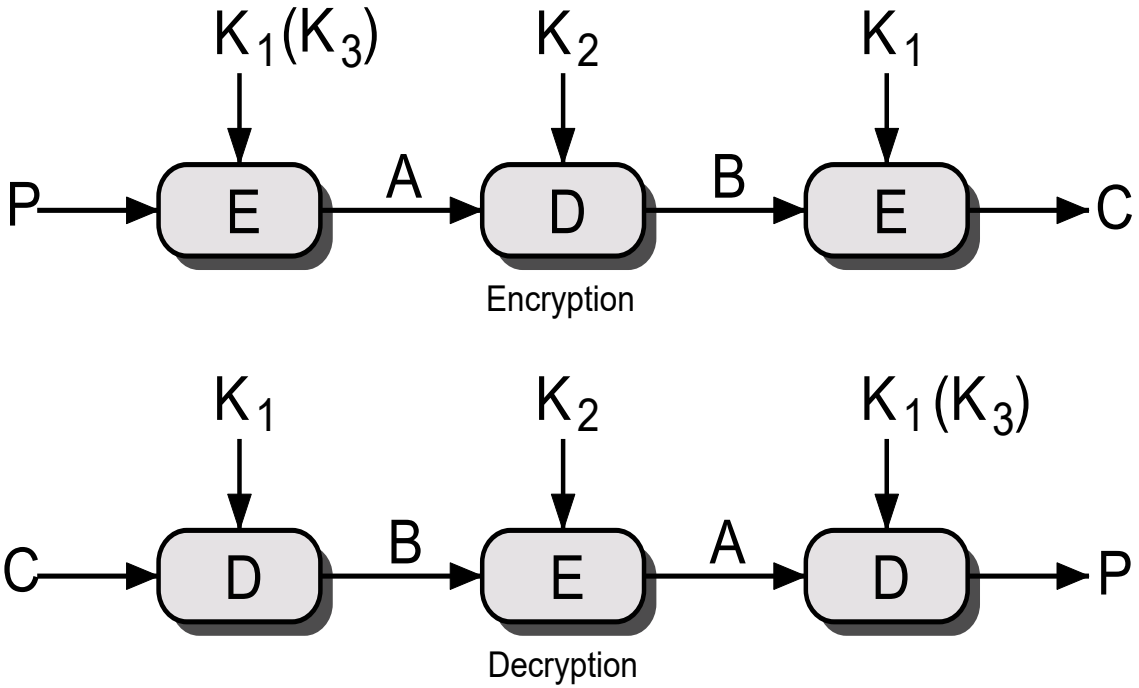
(b) Decryption

OFB Characteristics

- Similar to CFB except that it is the output of the encryption function that is fed back an not the ciphertext
- Bit errors in transmission do not propagate
- OFB is more vulnerable to a message stream modification attack than is CFB

Triple DES

- The problem with the DES algorithm is that the key is small and many people believe that the US government has a trapdoor and may easily break DES encrypted data.
- Triple DES solves partly this problem by effectively increasing the size of the key.



Other Symmetric Block Ciphers

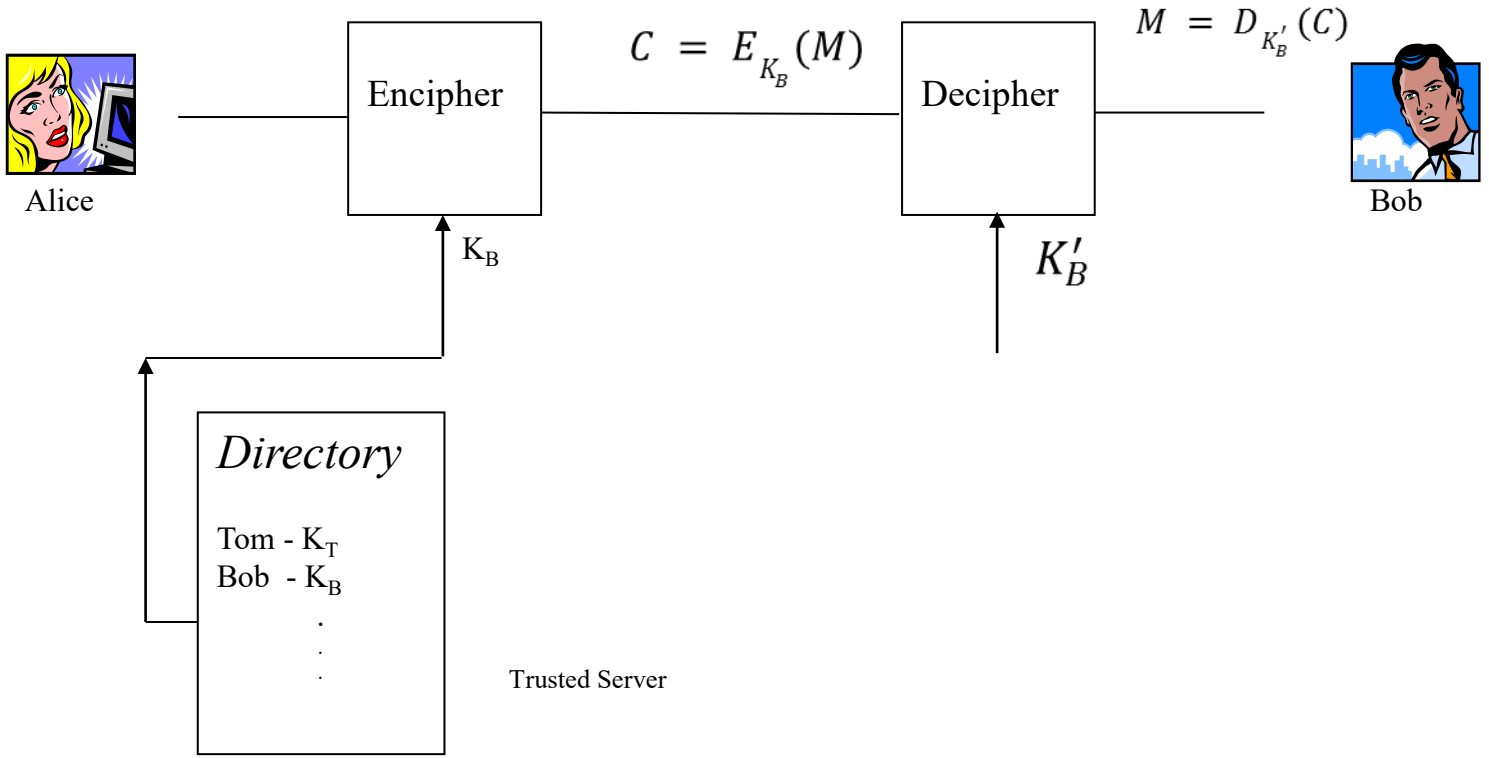
- International Data Encryption Algorithm (IDEA)
- Blowfish
- RC5
- Cast-128
- RC2

Speed Comparisons of Block Ciphers on a Pentium

Algorithm	Clock cycles per round	# of rounds	# of clock cycles per byte encrypted
Blowfish	9	16	18
RC5	12	16	23
DES	18	16	45
IDEA	50	8	50
Triple-DES	18	48	108

Public Key Cryptosystems

- This idea first appeared in a paper by Diffie Hellman in 1976.



Trapdoor One Way Function

- In a public key cryptosystem, the encoding key is public domain. However the decoding key is secret. What we require is that K_B' is hard to compute from K_B. However knowing K_B' it is easy to compute K_B.
- This may be achieved by using a trap door one way function. A one way function is one for which $f(x)$ is easy to compute but its inverse is not. An example of a one-way function is

$$f(x) = x^5 + 12x^3 + 107x + 123$$

- In this example it is easy to compute $f(x)$ knowing x but it is hard to find x knowing $f(x)$.
- A trap door one way function is a one way function whose inverse is easily computed if certain features used to design the function are known.

Trap-Door One-Way Function

- The invertible function f_k is a trap-door one-way function if it satisfies the following 3 criteria
  - $Y = f_k(X)$ easy, if k and X are known
  - $X = f_k^{-1}(Y)$ easy, if k and Y are known
  - $X = f_k^{-1}(Y)$ infeasible, if Y is known but k is not known
- Three main types of trap-door one-way functions used:
  - Factoring
  - Discrete Logarithms
  - Elliptic Curves

The Rivest-Shamir-Adelman Scheme (RSA)

- Messages are represented as integers in the range (0, n-1). The public key e (encryption key) together with n are publicly known (n, e). The decryption key d is a secret.
- Encryption : $C = E(M) = (M)^e \bmod n$
- Decryption : $M = D(C) = (C)^d \bmod n$
- Both these operation are easy to compute and result in integers in the range (0, n-1)
- In the RSA scheme $n = pq$
- where p and q are two large prime numbers which are kept hidden.
- No known polynomial time algorithm is known to exist to factorise n into prime numbers, hence knowing n it is very difficult to compute p and q.

- Let $\phi(n) = (p-1)(q-1)$

  (this is known as Euler's totient function)
- It can be shown that for any integer x in the range $(0, n-1)$ and integer k,

$$x = x^{k\phi(n)+1} \bmod n$$

- $\therefore$ All arithmetic, except in the exponent, is done modulo n, while that in the exponent is done modulo $\phi(n)$.
- d is chosen to be a large random integer relatively prime to $\phi(n)$,

  i.e. gcd $[\phi(n), d] = 1$.
- Any prime number greater than the larger of (p,q) will suffice. The integer e where $0 < e < \phi(n)$, is found from the following relationship

$$ed \bmod \phi(n) = 1$$

  hence $$x = x^{ed} \bmod n$$

$$\Rightarrow \quad M^e \bmod n = C$$

$$\Rightarrow \quad C^d \bmod n = (M^e)^d \bmod n$$

$$\Rightarrow \quad = M^{ed} \bmod n$$

$$= M$$

- Hence decryption is easy, knowing d.
- If the cryptanalyst can factor n into p and q then it would be very easy to compute d.
- However factorisation is a very difficult problem.

Example

Let $p = 47$ and $q = 59$

$\therefore \quad n = p.q = 2773$

$\phi(n) = (p-1)(q-1) = 2668$

The parameter d is chosen to be relatively prime to $\phi(n)$ for example d=157.

Now, $e.d \mod \phi(n) = 1$

i.e. $157.e \mod 2668 = 1$

e may then be calculated as follows. First compute a series x_0, x_1, x_{i+1}, where

$$x_0 = \phi(n)$$
$$x_1 = d$$
$$x_{i+1} = x_{i-1} \mod x_i$$

until $x_k = 0$ is found, then $\gcd(x_0, x_1) = x_{k-1}$

For each x_i , compute numbers a_i, b_i such that :

$$x_i = a_i\, x_0 + b_i\, x_1$$

where :

$$x_i = x_{i\text{-}2} - y_{i\text{-}1}\, x_{i\text{-}1}$$

$$a_i = a_{i\text{-}2} - y_{i\text{-}1}\, a_{i\text{-}1}$$

$$b_i = b_{i\text{-}2} - y_{i\text{-}1}\, b_{i\text{-}1}$$

$$y_i = \left\lfloor \frac{x_{i-1}}{x_i} \right\rfloor$$

Now, if $x_{k\text{-}1} = 1$ then $e = b_{k\text{-}1}$

and if $b_{k\text{-}1}$ is a negative number, the solution is :

$$e = b_{k\text{-}1} + \phi(n)$$

i	x_i	a_i	b_i	y_i
0	2668	1	0	-
1	157	0	1	16
2	156	1	-16	1
3	1	-1	**17**	

$$x_0 = \phi(n) = 2668$$
$$x_1 = d = 157$$
$$x_i = a_i\, x_0 + b_i\, x_1$$
$$x_i = x_{i\text{-}2} - y_{i\text{-}1}\, x_{i\text{-}1}$$
$$a_i = a_{i\text{-}2} - y_{i\text{-}1}\, a_{i\text{-}1}$$
$$b_i = b_{i\text{-}2} - y_{i\text{-}1}\, b_{i\text{-}1}$$
$$y_i = \left\lfloor \frac{x_{i-1}}{x_i} \right\rfloor$$

$$x_2 = 2668 - 157 \times 16 = 156$$
$$x_3 = 157 - 156 \times 1 = 1$$
$$\therefore \quad e = \underline{\mathbf{17}}$$

Alternatively, if d was chosen to be 61, then :

i	x_i	a_i	b_i	y_i
0	2668	1	0	-
1	61	0	1	43
2	45	1	-43	1
3	16	-1	42	2
4	13	3	-131	1
5	3	-4	175	4
6	1	19	**-831**	

$\therefore e = 2668 - 831 =$ **1837**

Now, with $d = 157$, $e = 17$ and $n = 2773$, we encode as follows. For the plain text example :

ITS ALL GREEK TO ME

and encoding :

space	by	00
A	by	01
B	by	02
:	:	:
Z	by	26

Then the above is encoded as :

09 20 19 00 01 12 12 00 07 18 05 05
11 00 20 15 00 13 05 00

Since any 4 digit sequence is ≤ 2626 then we can encode 4 digits at a time.

Therefore, encrypting the first 4 digits:

$$\begin{aligned} C &= (0920)^e \bmod n \\ &= 920^{17} \bmod 2773 \\ &= (920^2)^8(920) \bmod 2773 \\ &= [(920^2 \bmod 2773)^8.920] \bmod 2773 \\ &= [(635)^8.920] \bmod 2773 \\ &= 948 \end{aligned}$$

Therefore the complete ciphertext is given by

0948 2342 1084 1444 2663 2390 0778 0774 0219 1655

The plain text is recovered by applying the decryption key :

$$M = (C)^{157} \bmod 2773$$

Therefore for the first message symbol:

$$\begin{aligned} (0948)^{157} \bmod 2773 &= \{[(948)^2 \bmod 2773]^{78}.948\} \bmod 2773 \\ &= [252^{78}.948] \bmod 2773 \\ &= 0920 \end{aligned}$$

Authentication

- Used when it is required to ensure that a particular encrypted message originated from a particular source.

$E_B[D_A(M)]$

$E_A[D_B\{E_B[D_A(M)]\}] = M$

Alice

Bob

And he knows that only Alice knows D_A

Public directory

Oscar

Is not able to decrypt since he does not have D_B

E_A

E_B

Encryption keys for Alice and Bob