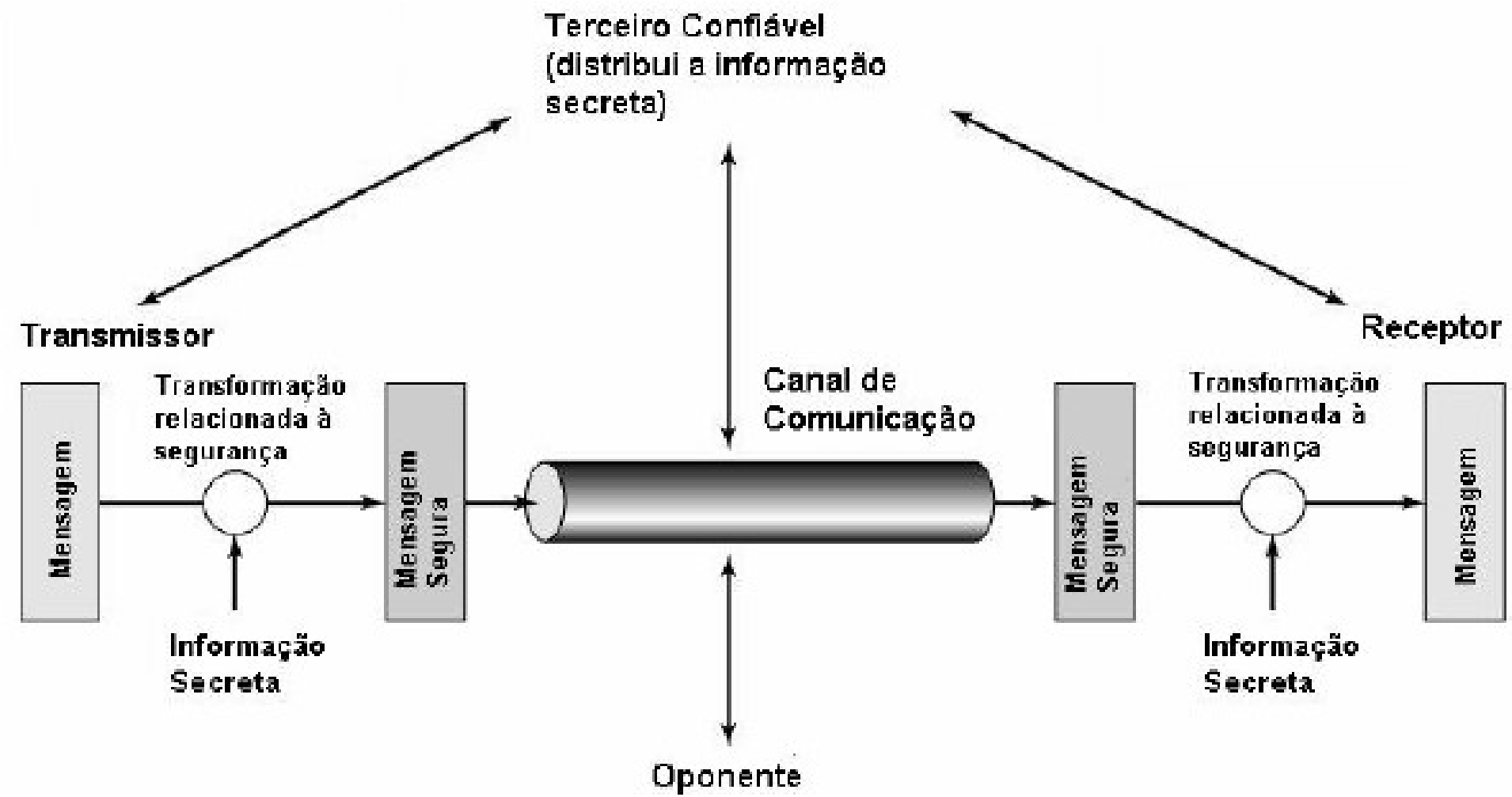
Introdução
Modelo de Segurança

- Modelo de segurança em Comunicação de Dados



Introdução
Criptografia - Conceitos

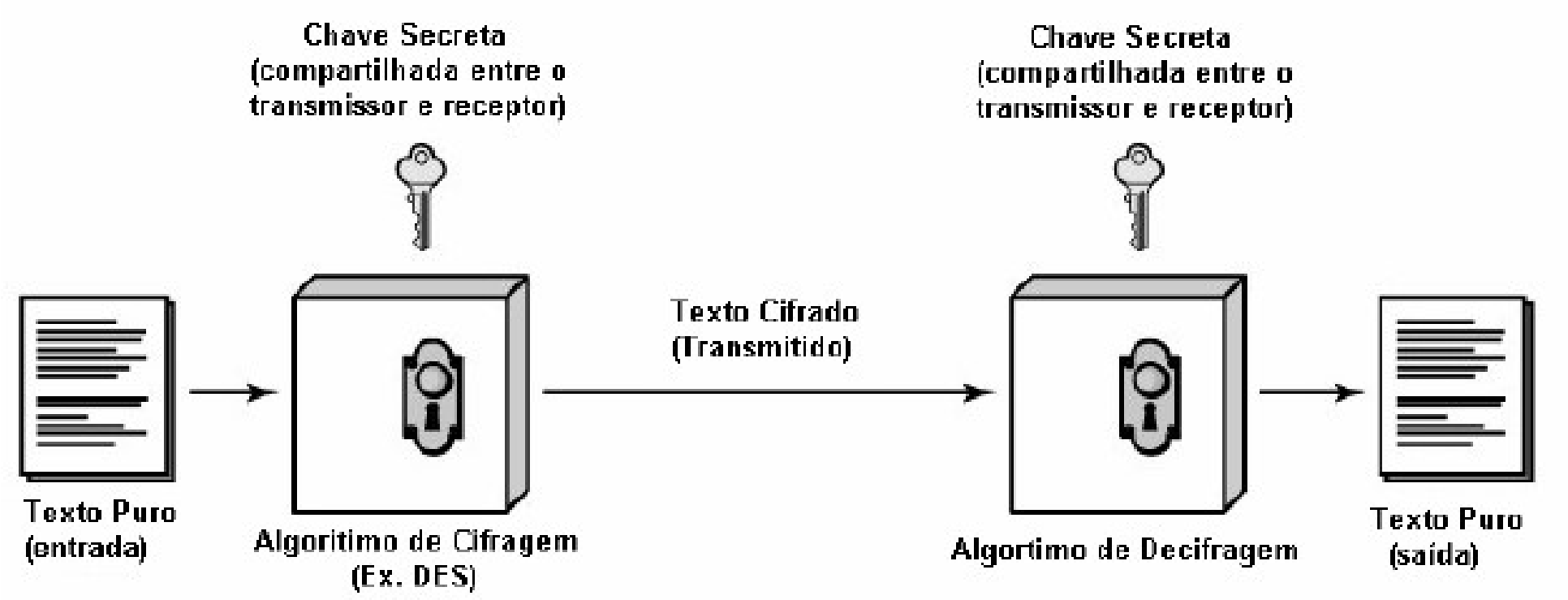
- Criptologia
  - Consiste no estudo das comunicações seguras
  - Disciplina que reúne os conhecimentos e técnicas necessários à criptoanálise e à criptografia
- Criptografia
  - Conjunto de princípios e técnicas empregada para cifrar uma mensagem a fim de torná-la secreta para os que não tenham acesso às convenções combinadas
  - Trata do desenvolvimento de algoritmos para cifrar e decifrar uma mensagem
- Criptoanálise
  - Conjunto de técnicas e métodos para a decifragem de uma mensagem escrita de sistema desconhecido

Introdução
Criptografia - Conceitos

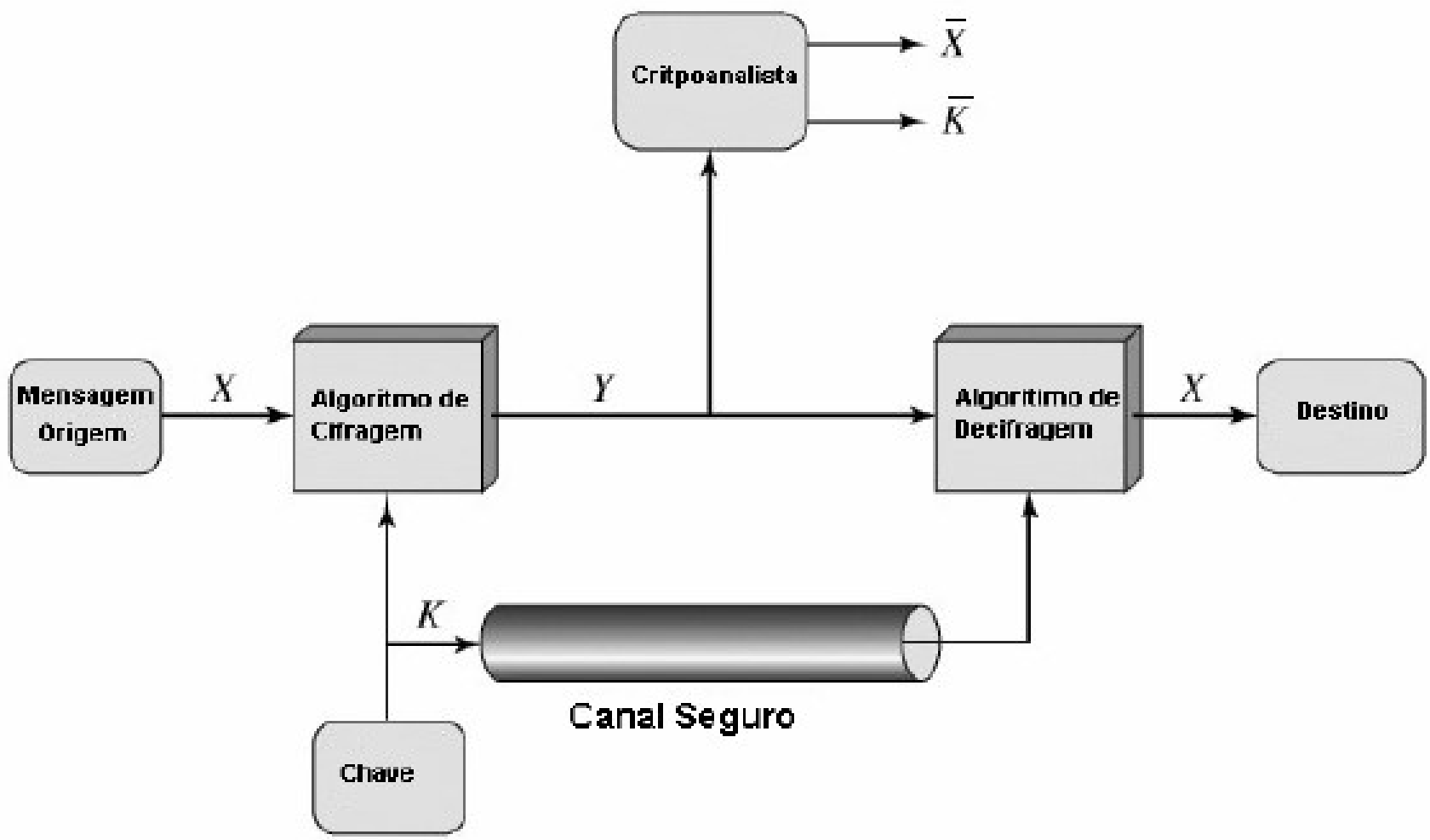
- Elementos de um sistema criptográfico
  - Texto Puro (Plaintext)
    - A mensagem original que pode ser entendida pelas partes envolvidas. Utilizado como entrada no algoritmo de cifragem
  - Algoritmo de Cifragem (Encryption Algorithm)
    - Algoritmo que realiza transformações sobre o texto puro
  - Chave Secreta (Secret Key)
    - Também utilizada pelo algoritmo de cifragem.
    - Valor independente do texto plano e do algoritmo
    - A cada chave utilizada um novo resultado será produzido pelo algoritmo de cifragem
  - Texto Cifrado (Ciphertext)
    - A mensagem produzida como saída do algoritmo de cifragem. Esta mensagem não pode ser compreendida
    - Depende do texto puro e da chave utilizada
    - Para uma dada mensagem duas diferentes chaves produzirão diferentes textos cifrados
  - Algoritmo de Decifragem (Decryption Algorithm)
    - Realiza o processo inverso daquele executado pelo algoritmo de cifragem
    - Utiliza como texto cifrado como saída e juntamente com a chave, produz a mensagem original (texto puro)

Introdução
Criptografia Simétrica

- Modelo Convencional de Criptografia
- No modelo de criptografia simétrica a mesma chave é utilizada tanto no receptor quanto no transmissor



Introdução
Sistema Criptográfico



Introdução
Sistema Criptográfico

- Criptoanálise
  - Força Bruta – Exemplos de esforço necessário para descoberta da chave secreta

Tamanho Chave (Bits)	Número de Chaves Possíveis	Tempo necessário (Considerando 1 teste/μs)	Tempo necessário (Considerando 10^6 testes/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
168	$2^{168} = 3.7 \times 10^{50}$	2^{167} μs = 5.9×10^{36} years	5.9×10^{30} years
26 characters (permutation)	$26! = 4 \times 10^{26}$	2×10^{26} μs = 6.4×10^{12} years	6.4×10^{6} years

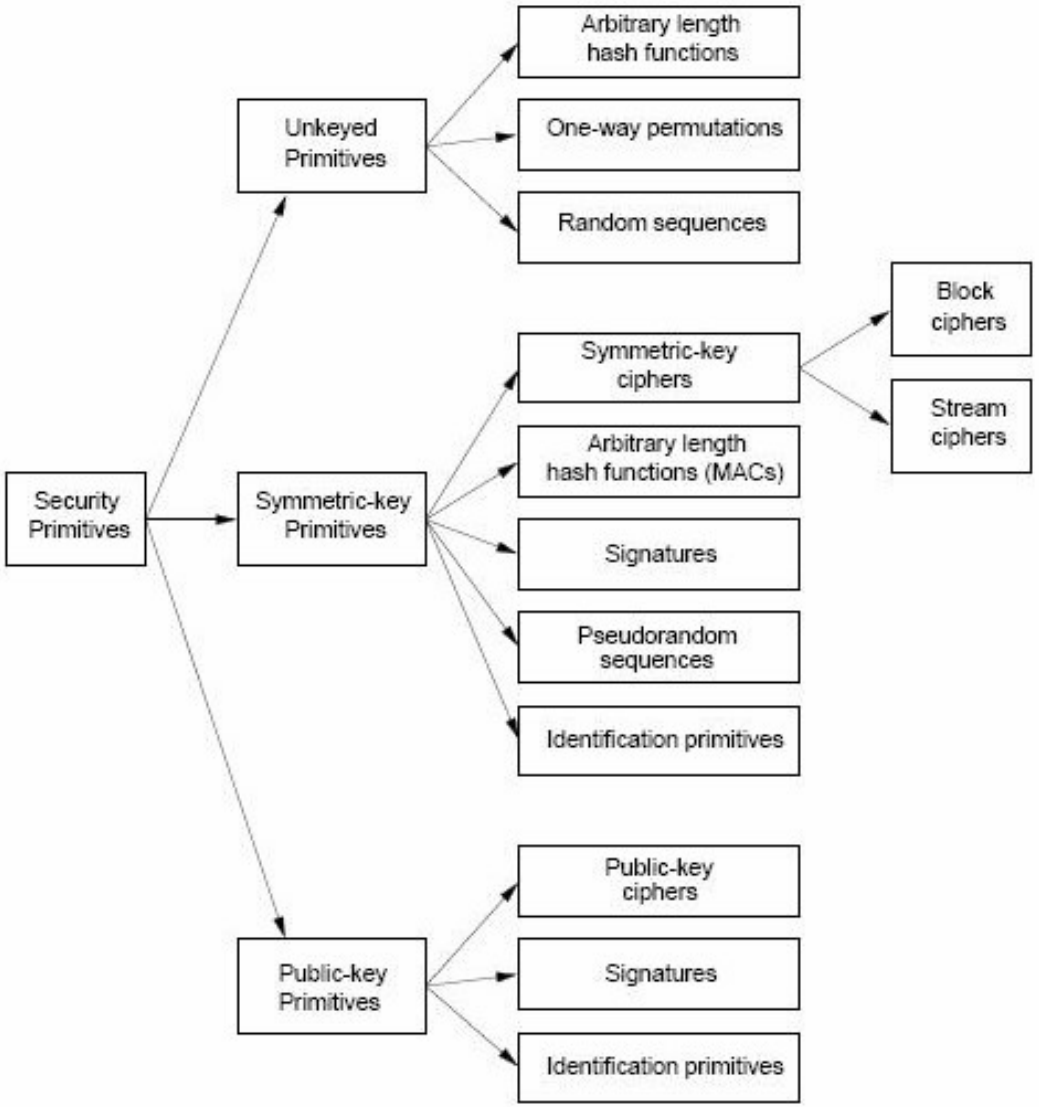
Introdução
Sistema Criptográfico

- Texto Puro
  - $X = [X_1, X_2, ..., X_M]$, possui m elementos
  - A mensagem é escrita em um alfabeto que pode ser as 26 letras maiúsculas ou alfabeto binário {0,1}
- Chave
  - $K = [K_1, K_2, ..., K_J]$, possui j elementos
  - A chave pode ser gerada por um terceiro e distribuída através de um canal seguro
- Texto Cifrado
  - $Y = [Y_1, Y_2, ..., Y_N]$, possui n elementos
  - $Y = E(X,K)$
- Processo de Decifragem
  - $X = D(Y,K)$
  - Um oponente que observa Y, mas não tem acesso a K ou X, porém conhece os algoritmos E e D, pode estimar a mensagem $\overline{X}$ assim como estimar a chave $\overline{K}$, permitindo desta forma o acesso a futuras mensagens

Introdução
Sistema Criptográfico - Tipos

- Um sistema criptográfico pode ser caracterizado de 3 diferentes formas:
- Quanto ao tipo de operações
  - Substituição – Um elemento da mensagem é mapeado em um outro elemento
  - Transposição – Os elementos são reorganizados na mensagem a fim de gerar o texto cifrado
  - Produto – Neste caso o sistema utiliza vários estágios de transposição e substituição
- Quanto a número de chaves utilizadas
  - Simétrico – Utiliza uma única chave tanto para o transmissor quanto para o receptor
  - Assimétrico – Neste caso transmissor e receptor utiliza diferentes chaves. Também conhecido como de chave pública
  - Sem Chaves – Não utiliza chaves
- Quanto à maneira que o texto puro é processado (Chaves Simétricas)
  - Cifradores de Bloco (Block chiphers) – Processa um bloco de elementos de cada vez
  - Cifradores Contínuos (Stream Chiphers) – Produz uma saída continua e a medida que um elemento entra no sistema, outro elemento é produzido na saída

Taxonomia de Técnicas Criptográficas



Conceitos Básicos
Substituição de César

- Alfabeto

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
A	B	C	D	E	F	G	H	I	J	K	L	M	N	0	P	Q	R	S	T	U	V	W	X	Y	Z

- Operação de Cifragem
  - O caractere cifrado da posição i, será calculado da seguinte forma:
    - Ci = (i+K) mod 26
    - Sendo i a posição no caractere na mensagem $0 \leq i \leq 25$
- Decifrar
  - O caractere original da posição i, será calculado da seguinte forma:
    - Ci = (i-K) mod 26
    - Sendo i a posição no caractere no texto cifrado $0 \leq i \leq 25$
  - O valor K é a chave utilizada $0 \leq K \leq 25$
  - Neste algoritmo existe a possibilidade de uso de 26 diferentes chaves

Conceitos Básicos
Substituição de César - Exemplo

- Alfabeto

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
A	B	C	D	E	F	G	H	I	J	K	L	M	N	0	P	Q	R	S	T	U	V	W	X	Y	Z

- Seja M a mensagem “ATACAR” e K = 3
- Desta forma teremos:
  - **M = ATACAR, será cifrada da seguinte forma:**
  - **C = DWDFDU**
  - Para i = 0 tem-se: C0 = (0+3) mod 26 = 3 mod 26 = 3, logo o novo caracter encontra-se na posição 3, ou seja, “D”

Algoritmo DES Funcionamento

- O algoritmo DES possui duas etapas principais:
  - Expansão da Chave
    - A partir da chave inicial de 56 bits, 16 novas chaves de 48 bits são obtidas
    - Cada chave será utilizada em uma etapa o algoritmo
  - Processo de Cifragem
    - Substituição e Permutas da mensagem com as chaves obtidas no passo anterior
- Especificação
  - http://www.itl.nist.gov/fipspubs/fip46-2.htm
  - A especificação do DES foi perdeu sua validade em Maio de 2005

Algoritmo DES Funcionamento

- Mensagem = $0123456789ABCDEF_H$

```
1000000001000000000200000000030000000004000000000500000000060006 4
1234567812345678123456781234567812345678123456781234567812345678
0000000100100011010001010110011111000100110101011110011011110111 1
```

- Chave(K) = $133457799BBCDFF1_H$

```
1000000001000000000200000000030000000004000000000500000000060006 4
1234567812345678123456781234567812345678123456781234567812345678
0001001100110100101011101111001100110111011110011011111111100011   K
```

- Os bits da chave serão agrupados em grupos de 8 bits e o oitavo bit será removido (bit de paridade), desta forma a chave terá apenas 56 bits
- Na figuras cada caractere está sendo representado por 4 bits

Algoritmo DES - Funcionamento Expansão da Chave

- A chave K de 64 bits é permutada de acordo com a tabela (PC1) a fim de obter a chave X, conforme mostrado a seguir

PC1 =

57	49	41	33	25	17	09
01	58	50	42	34	26	18
10	02	59	51	43	35	27
19	11	03	60	52	44	36
63	55	47	39	31	23	15
07	62	54	46	38	30	22
14	06	61	53	45	37	29
21	13	05	28	20	12	04

- Pela tabela acima o bit de número 57 na chave K, torna-se o bit de numero 1 na chave X
- O bit 49 torna-se o segundo da chave X ; o bit 41 o terceiro; e assim sucessivamente, sendo que o último bit da chave X será o bit 4 da chave K
- Desta forma a chave X ficará, após este processo, da seguinte forma:

```
10000000010000000002000000000300000000040000000005000000000600064

12345678123456781234567812345678123456781234567812345678123456781234567812345678

11110000110011001010101011111010101010110011001111100011111    (X; 56 bits)
```

Algoritmo DES - Funcionamento Expansão da Chave

- A chave X será divida ao meio e cada uma terá 28 bits. A chave da esquerda será chamada C_0 e a chave da direita D_0

```
10000000010000000002000000000300000000040000000005000000006000 64
12345678123456781234567812345678123456781234567812345678123456 78
11110000110011001010101011111010101010110011001111000111 1    (X; 56 bits)
```

- Após a divisão teremos:

```
10000000010000000002000000000300000000040000000005000000006000 64
12345678123456781234567812345678123456781234567812345678123456 78
11110000110011001010101111    (28 bits) → C0
01010101011001100111100011 11 (28 bits) → D0
```

Algoritmo DES - Funcionamento Expansão da Chave

- A partir dos valores da chave da esquerda C0 e da chave da direita D0 serão criadas 16 chaves Cn; 1 ≤ n ≤ 16
- Para i = 1 até 16 tem-se que:
  - $Ci = C_{i-1} << \sigma(i)$ // << = rotação circular à esquerda de 1 ou 2 bits
  - $Di = D_{i-1} << \sigma(i)$ // << = rotação circular à esquerda de 1 ou 2 bits
  - $K_i = (C_i, D_i)$ // conjunto de bits à esquerda e à direita concatenados
  - $CH_i = PC2(Ki)$
- A função σ (i) receber um número de 1 a 16 e retorna 1 ou 2, da seguinte forma

$$\sigma(i) = \begin{cases} 1 \; se \; i = 1,2,9,16 \\ 2 \; se \; i \neq 1,2,9,16 \end{cases}$$

Algoritmo DES - Funcionamento Expansão da Chave

- No exemplo anterior teremos

```
10000000010000000002000000000300000000040000000005000000006000064
12345678123456781234567812345678123456781234567812345678123456781234567812345678
11110000110011001010101111    (28 bits) → C0
01010101011001100111100011111 (28 bits) → D0
```

- Cálculo de C1 e D1

```
10000000010000000002000000000300000000040000000005000000006000064
12345678123456781234567812345678123456781234567812345678123456781234567812345678
11110000110011001010101011111    (28 bits) → C1 = C0 << 1 ; σ(1)
10101010110011001111100011110    (28 bits) → D1 = D0 << 1 ; σ(1)
```

Algoritmo DES - Funcionamento Expansão da Chave

- Cálculo de C_1 e D_1

```
1000000001000000000200000000030000000004000000000500000000060006 4
12345678123456781234567812345678123456781234567812345678123456781234567812345678
1110000110011001010101011111    (28 bits) → C1 = C0 << 1 ; σ(1)
1010101011001100111100011110    (28 bits) → D1 = D0 << 1 ; σ(1)
```

- Cálculo $K_1 = (C_1, D_1)$

```
1000000001000000000200000000030000000004000000000500000000060006 4
12345678123456781234567812345678123456781234567812345678123456781234567812345678
111000011001100101010101111111010101011001100111100011110          K1
```

Algoritmo DES - Funcionamento
Expansão da Chave

```
10000000010000000002000000000300000000040000000005000000000600064
12345678123456781234567812345678123456781234567812345678123456781234567812345678
11100001100110010101010111111010101011001100111100011110          K1
```

- A partir da chave K_1 será feita a permuta PC2, para o cálculo da Chave CH_1

PC2 =

14	17	11	24	01	05
03	28	15	06	21	10
23	19	12	04	26	08
16	07	27	20	13	02
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Pela tabela ao lado o bit de número 14 na chave K_1, torna-se o bit de numero 1 na chave CH_1
O bit 17 torna-se o segundo da chave CH_1; o bit 11 o terceiro; e assim sucessivamente, sendo que o último bit da chave CH_1 será o bit 32 da chave K_1
A chave CH1 terá 48 bits

```
10000000010000000002000000000300000000040000000005000000000600064
12345678123456781234567812345678123456781234567812345678123456781234567812345678
000110110000001011101111111111000111000001110010          CH1
```

Algoritmo DES - Funcionamento Expansão da Chave

- Utilizando-se o mesmo método a partir de C_1 e D_1, pode-se calcular C_2 e D2 e então é feito o cálculo da CH_2
- Neste exemplo teremos os seguintes valores para C_i e D_i

```
C0 = 11110000110011001010101011111 D0 = 0101010101100110011110001111
C1 = 11100001100110010101010111111 D1 = 1010101011001100111100011110
C2 = 11000011001100101010101111111 D2 = 0101010110011001111000111101
C3 = 00001100110010101010111111111 D3 = 0101011001100111100011110101
C4 = 00110011001010101011111111100 D4 = 0101100110011110001111010101
C5 = 11001100101010101111111110000 D5 = 0110011001111000111101010101
C6 = 00110010101010111111111000011 D6 = 1001100111100011110101010101
C7 = 11001010101011111111100001100 D7 = 0110011110001111010101010110
C8 = 00101010101111111110000110011 D8 = 1001111000111101010101011001
```

Algoritmo DES - Funcionamento Expansão da Chave

- Continuação dos valores de C_i e D_i

```
C09 = 010101010111111110001100110 D09 = 001111000111101010101011001 1
C10 = 010101011111111000110011001 D10 = 111100011110101010110011000
C11 = 010101111111100011001100101 D11 = 110001111010101011001100111
C12 = 010111111110001100110010101 D12 = 000111101010101100110011111
C13 = 011111111000110011001010101 D13 = 011110101010101100110011110 0
C14 = 111111100011001100101010101 D14 = 111010101010110011001111000 1
C15 = 111110001100110010101010111 D15 = 101010101011001100111100011 1
C16 = 111100011001100101010101111 D16 = 010101010110011001111000111 1
```

Algoritmo DES - Funcionamento Expansão da Chave

- A partir dos valores de C_i e D_i será possível calcular as chaves K_i e também as chaves que serão utilizadas em cada uma das etapas do DES
- Neste exemplo teremos os seguintes valores para as chaves CH_i
- As chaves CH_i possuem 48 bits

```
CH1 = 000110 110000 001011 101111 111111 000111 000001 110010
CH2 = 011110 011010 111011 011001 110110 111100 100111 100101
CH3 = 010101 011111 110010 001010 010000 101100 111110 011001
CH4 = 011100 101010 110111 010110 110110 110011 010100 011101
CH5 = 011111 001110 110000 000111 111010 110101 001110 101000
CH6 = 011000 111010 010100 111110 010100 000111 101100 101111
CH7 = 111011 001000 010010 110111 111101 100001 100010 111100
CH8 = 111101 111000 101000 111010 110000 010011 101111 111011
```

Algoritmo DES - Funcionamento Expansão da Chave

- Continuação dos valores para das chaves CH_i

```
CH09 = 111000 001101 101111 101011 111011 011110 011110 000001
CH10 = 101100 011111 001101 000111 101110 100100 011001 001111
CH11 = 001000 010101 111111 010011 110111 101101 001110 000110
CH12 = 011101 010111 000111 110101 100101 000110 011111 101001
CH13 = 100101 111100 010111 010001 111110 101011 101001 000001
CH14 = 010111 110100 001110 110111 111100 101110 011100 111010
CH15 = 101111 111001 000110 001101 001111 010011 111100 001010
CH16 = 110010 110011 110110 001011 000011 100001 011111 110101
```

Algoritmo DES – Funcionamento Processo de Cifragem

- Esta etapa utilizará o seguinte algoritmo
  - MP = IP(M) //permuta da mensagem utilizando o vetor IP
  - (L0,R0) = MP //Divisão de MP na parte esquerda e Direita
  - Para i de 1 até 16 tem-se que:
    1. $L_i = R_{i-1}$ //Inicialmente Li e Ri-1 possuem 32 bits
    2. $R_i = ET(R_{i-1})$ //Permuta usando vetor ET (Expansion Table); saída 48 bits
    3. $R_i = CH_i$ **XOR** $\mathbf{R_i}$ //Operação de OU-EXCLUSIVO entre CH_i e R_i obtido no passo 2
    4. $R_i = SB[R_i]$ //Permuta utilizando as Caixas-S (S-BOX), saída 32 bits
    5. $R_i = P(R_i)$ //Permuta utilizando P(Permutation Function), saída 32 bits
    6. $R_i = L_{i-1}$ **XOR** $\mathbf{R_i}$ //Operação de OU-EXCLUSIVO entre Li-1 e Ri obtido em 5
  - C = (R16,L16) //O texto cifrado é a seqüência de bits invertida
  - C = IP-1(C) //Finalmente é feita a permuta inversa IP-1 do texto cifrado

Algoritmo DES
Processo de Cifragem - Exemplo

- Neste exemplo abaixo é novamente mostrado a mensagem(M) que será criptografada utilizando o DES
- M = $0123456789ABCDEF_H$

```
1000000001000000002000000003000000004000000005000000006000 64
1234567812345678123456781234567812345678123456781234567812345678
0000000100100011010001010110011111000100110101011110011011110 1111      M
```

- A partir da chave inicial (K) foram obtidas as 16 chaves que serão agora utilizadas
- K = $133457799BBCDFF1_H$

Algoritmo DES - Funcionamento
Processo de Cifragem - Exemplo

- Inicialmente será feita a permuta na mensagem IP(M) - M = $0123456789ABCDEF_H$

```
1000000001000000000200000000030000000004000000000500000000060006 4
1234567812345678123456781234567812345678123456781234567812345678
0000000100100011010001010110011111000100110101011110011011110111 1111    M
```

- A partir mensagem M será feita a permuta IP, para o cálculo da Chave MP

IP =

58	50	42	34	26	18	10	02
60	52	44	36	28	20	12	04
62	54	46	38	30	22	14	06
64	56	48	40	32	24	16	08
57	49	41	33	25	17	09	01
59	51	43	35	27	19	11	03
61	53	45	37	29	21	13	05
63	55	47	39	31	23	15	07

Pela tabela ao lado o bit de número 58 na mensagem M, torna-se o bit de numero 1 na mensagem MP
O bit 50 torna-se o segundo em MP; o bit 42 o terceiro; e assim sucessivamente, sendo que o último bit da mensagem MP será o bit 07 da mensagem M
A saída terá 64 bits

```
1000000001000000000200000000030000000004000000000500000000060006 4

1234567812345678123456781234567812345678123456781234567812345678

1100110000000000110011001111111111110000101010101111000010101010    MP=IP(M)
```

Algoritmo DES - Funcionamento Processo de Cifragem - Exemplo

- A mensagem então será dividida em duas partes de 32 bits cada. Sendo L0 os 32 primeiros bits, ou seja os bits à esquerda e R0 os bits à direita

```
10000000010000000002000000000300000000040000000005000000000600064
12345678123456781234567812345678123456781234567812345678123456781234567812345678
11001100000000001100110011111111111100001010101011110000101010101    MP
11001100000000001100110011111111                                      LO
11110000101010101111000010101010                                      R0
```

- A seguir são realizadas as 16 iterações do algoritmo
- Neste caso, será mostrada a apenas primeira iteração

Algoritmo DES - Funcionamento
Processo de Cifragem - Exemplo

- L1 será igual à seqüência R0

```
10000000010000000002000000000300000000040000000005000000006000064

12345678123456781234567812345678123456781234567812345678123456781234567812345678
11110000101010101111000010101010                R0
11110000101010101111000010101010                R0

11110000101010101111000010101010 (L1)           //L1 = R0
```

- Será realizada e expansão de R0 usando a permuta ET (Expansion Table)

ET =

32	01	02	03	04	05
04	05	06	07	08	09
08	09	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	01

Pela tabela ao lado o bit de número 32 em R0, tornar-se-á o bit de numero 1 em R1
O bit 01 tornar-se-á o segundo em R1; o bit 02 o terceiro; e assim sucessivamente, sendo que o último bit em R1 será o bit 01
Neste caso o R1 resultante terá 48 bits

```
011110 100001 010101 010101 011110 100001 010101 010101    R1 = ET(R0)
```

Algoritmo DES - Funcionamento Processo de Cifragem - Exemplo

- Para I = 1; sendo esta a primeira iteração. Neste caso será realizada um operação de OU-EXCLUSIVO (XOR) entre CH_i, neste caso será entre CH_1 e R_1

```
100000 000100 000000 020000 000003 000000 000400 000000050000000000600064

123456 781234 567812 345678 123456 781234 567812 345678123456781234 5678
011110 100001 010101 010101 011110 100001 010101 010101      R1
000110 110000 001011 101111 111111 000111 000001 110010      CH1
011000 010001 011110 111010 100001 100110 010100 100111      R1= CH1 XOR R1
```

- A próximo passo fará uso das caixas S
- Estas caixas farão redução de 48 bits para 32 bits, em R_i

Algoritmo DES - Funcionamento Processo de Cifragem – S-Boxes

- As Caixas-S (S-Boxes) são definidas da seguinte forma:

S_1 =

i	j 00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
0	14	04	13	01	02	15	11	08	03	10	06	12	05	09	00	07
1	00	15	07	04	14	02	13	01	10	06	12	11	09	05	03	08
2	04	01	14	08	13	06	02	11	15	12	09	07	03	10	05	00
3	15	12	08	02	04	09	01	07	05	11	03	14	10	00	06	13

S_2 =

	15	01	08	14	06	11	03	04	09	07	02	13	12	00	05	10
	03	13	04	07	15	02	08	14	12	00	01	10	06	09	11	05
	00	14	07	11	10	04	13	01	05	08	12	06	09	03	02	15
	13	08	10	01	03	15	04	02	11	06	07	12	00	05	14	09

S_3 =

	10	00	09	14	06	03	15	05	01	13	12	07	11	04	02	08
	13	07	00	09	03	04	06	10	02	08	05	14	12	11	15	01
	13	06	04	09	08	15	03	00	11	01	02	12	05	10	14	07
	01	10	13	00	06	09	08	07	04	15	14	03	11	05	02	12

S_4 =

	07	13	14	03	00	06	09	10	01	02	08	05	11	12	04	15
	13	08	11	05	06	15	00	03	04	07	02	12	01	10	14	09
	10	06	09	00	12	11	07	13	15	01	03	14	05	02	08	04
	03	15	00	06	10	01	13	08	09	04	05	11	12	07	02	14

Algoritmo DES - Funcionamento Processo de Cifragem – S-Boxes

- As Caixas-S (S-Boxes) são definidas da seguinte forma:

S_5 =

i	j 00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
0	02	12	04	01	07	10	11	06	08	05	03	15	13	00	14	09
1	14	11	02	12	04	07	13	01	05	00	15	10	03	09	08	06
2	04	02	01	11	10	13	07	08	15	09	12	05	06	03	00	14
3	11	08	12	07	01	14	02	13	06	15	00	09	10	04	05	03

S_6 =

	12	01	10	15	09	02	06	08	00	13	03	04	14	07	05	11
	10	15	04	02	07	12	09	05	06	01	13	14	00	11	03	08
	09	14	15	05	02	08	12	03	07	00	04	10	01	13	11	06
	04	03	02	12	09	05	15	10	11	14	01	07	06	00	08	13

S_7 =

	04	11	02	14	15	00	08	13	03	12	09	07	05	10	06	01
	13	00	11	07	04	09	01	10	14	03	05	12	02	15	08	06
	01	04	11	13	12	03	07	14	10	15	06	08	00	05	09	02
	06	11	13	08	01	04	10	07	09	05	00	15	14	02	03	12

S_8 =

	13	02	08	04	06	15	11	01	10	09	03	14	05	00	12	07
	01	15	13	08	10	03	07	04	12	05	06	11	00	14	09	02
	07	11	04	01	09	12	14	02	00	06	10	13	15	03	05	08
	02	01	14	07	04	10	08	13	15	12	09	00	03	05	06	11

Algoritmo DES - Funcionamento Processo de Cifragem – S-Boxes

- Todas as S-Boxes são utilizadas uma iteração da seguinte forma:

```
011000 010001 011110 111010 100001 100110 010100 100111     R1= CH1 XOR R1
  B1     B2     B3     B4     B5     B6     B7     B8
```

- Cada grupo de seis bits em R_i nos dará um endereço numa caixa S diferente.
  - **S1(B1)S2(B2)S3(B3)S4(B4)S5(B5)S6(B6)S7(B7)S8(B8)**
- Um número de 4 bits estará localizado neste endereço. Este número de 4 bits irá substituir os 6 bits originais.
- Desta forma cada oito grupos de 6 bits são transformados em oito grupos de 4 bits (as saídas de 4 bits das caixas S) num total de 32 bits.
- Para cada conjunto S_K (B_K) teremos:
  - O primeiro e o último bit de B_K serão obtidos.
  - A partir destes dois bits será obtido um número (i).
  - Este número será a i-ésima linha da S-BOX S_K
  - Os quatro bits centrais de B_K serão obtidos. Estes bits representa um número entre 0 e 15 e será o número j
  - Este numero será a j-ésima coluna da S-BOX S_K

Algoritmo DES - Funcionamento Processo de Cifragem – S-Boxes

- Voltando ao exemplo, na primeira iteração teremos:

```
011000 010001 011110 111010 100001 100110 010100 100111      R1= CH1 XOR R1
  B1     B2     B3     B4     B5     B7     B7     B8
```

- Como temos:
  - S1(B1)S2(B2)S3(B3)S4(B4)S5(B5)S6(B6)S7(B7)S8(B8)
  - **B1 = 011000 , logo i = 00 e j = 1100; ou seja i = 0 e j = 12**
  - Na caixa S1 temos na linha 0 e coluna 12 o valor 5
  - logo a saída S1(B1) será igual a 5 ou seja: 0101
  - Repete-se este processo para cada S-BOX
  - Ao final teremos o seguinte resultado para R1

```
011000 010001 011110 111010 100001 100110 010100 100111      R1= CH1 XOR R1
  B1     B2     B3     B4     B5     B7     B7     B8
0101   1100   1000   0010   1011   0101   1001   0111        R1 = SB[R1]
S1B1   S2B2   S3B3   S4B4   S5B5   S6B6   S7B7   S8B8
```

Algoritmo DES
Processo de Cifragem - Exemplo

- Será feita a permuta P sobre o valor R1

```
1000 0000 0100 0000 0002 0000 0000 0300000000040000000005000000006000064
1234 5678 1234 5678 1234 5678 1234 5678123456781234567812345678123456781234567812345678
0101 1100 1000 0010 1011 0101 1001 0111                    R1 = SB[R1]
```

- Será realizada permuta P sobre o valor R1

P =

16	07	20	21
29	12	28	17
01	15	23	26
05	18	31	10
02	08	24	14
32	27	03	09
19	13	30	06
22	11	04	25

Pela tabela ao lado o bit de número 16 em R1, tornar-se-á o bit de numero 1 em
R1 O bit 07 tornar-se-á o segundo em R1; o
bit 20 o terceiro; e assim sucessivamente, sendo que o último bit em R1 será o bit 25
Neste caso o R1 terá 32 bits

```
1000 0000 0100 0000 0002 0000 0000 0300000000040000000005000000006000064
1234 5678 1234 5678 1234 5678 1234 5678123456781234567812345678123456781234567812345678
0010 0011 0100 1010 1010 1001 1011 1011                    R1 = P(R1)
```

Algoritmo DES - Funcionamento Processo de Cifragem - Exemplo

- Para o cálculo final de R1 será feito um XOR entre R1 e L0, conforme mostrado abaixo:

```
1000 0000 0100 0000 0002 0000 0000 0300000000040000000005000000006000064
1234 5678 1234 5678 1234 5678 1234 5678123456781234567812345678123456781234567812345678
0010 0011 0100 1010 1010 1001 1011 1011                R1 = P(R1)
1100 1100 0000 0000 1100 1100 1111 1111                L0
1110 1111 0100 1010 0110 0101 0100 0100                R1 = R1 XOR L0
```

- Ao final deste processo temos o cálculo de R1 e L1 para a primeira iteração

```
1000 0000 0100 0000 0002 0000 0000 0300000000040000000005000000006000064
1234 5678 1234 5678 1234 5678 1234 5678123456781234567812345678123456781234567812345678

1111 0000 1010 1010 1111 0000 1010 1010                L1 = = R0
1110 1111 0100 1010 0110 0101 0100 0100                R1 = R1 XOR L0
```

Algoritmo DES - Funcionamento Processo de Cifragem - Exemplo

- A partir de L1 e R1 deve ser realizado todo o processo novamente (1 ≤i ≤16)

```
1000 0000 0100 0000 0002 0000 0000 0300000000040000000005000000000600064
1234 5678 1234 5678 1234 5678 1234 5678123456781234567812345678123456781234567812345678

1111 0000 1010 1010 1111 0000 1010 1010          L1 = = R0
1110 1111 0100 1010 0110 0101 0100 0100          R1 = R1 XOR L0
```

- Para i igual a 16 teremos

```
1000 0000 0100 0000 0002 0000 0000 0300000000040000000005000000000600064
1234 5678 1234 5678 1234 5678 1234 5678123456781234567812345678123456781234567812345678

0100 0011 0100 0010 0011 0010 0011 0100          L16 = = R15
0000 1010 0100 1100 1101 1001 1001 0101          R16 = R16 XOR L15
```

- O texto cifrado C será igual a (R16,L16), logo temos

```
1000000001000000000200000000030000000004000000000500000000060006 4
1234567812345678123456781234567812345678123456781234567812345678123456781234567812345678

0000101001001100110110011001010100001101000010001100100011010 0      C
```

Algoritmo DES
Processo de Cifragem - Exemplo

- Finalmente será feita a permutação inversa do texto Cifrado

```
10000000010000000002000000000300000000040000000005000000006000 64
12345678123456781234567812345678123456781234567812345678123456 78
00001010010011001101100110010101010000110100001000110010001101 00    C
```

- Permutação Inversa Utilizando IP-1

IP^{-1} =

40	08	48	16	56	24	64	32
39	07	47	15	55	23	63	31
38	06	46	14	54	22	62	30
37	05	45	13	53	21	61	29
36	04	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	02	42	10	50	18	58	26
33	01	41	09	49	17	57	25

```
1000010111101000000100110101010000001111000010101011010000000101    C = IP-1(C)

Logo para M = 0123456789ABCDEF teremos C = 85E813540F0AB405
```

Algoritmo DES
Processo de Decifragem

- O processo de decifragem utiliza o mesmo algoritmo da cifragem
- Neste caso porém as chaves são utilizadas na ordem reversa
- Esta etapa utilizará o seguinte algoritmo
  - MP = IP(C) //permuta do texto cifrado utilizando o vetor IP
  - (L0,R0) = MP //Divisão de MP na parte esquerda e Direita
  - Para i de 1 até 16 tem-se que:
    - $L_i = R_{i-1}$ //Inicialmente Li e Ri-1 possuem 32 bits
    - $R_i = ET(R_{i-1})$ //Permuta usando vetor ET (Expansion Table); saída 48 bits
    - $R_i = CH_{17-i}$ **XOR** $\mathbf{R_i}$ //Operação de OU-EXCLUSIVO entre CH_i e R_i obtido no passo 2
    - $R_i = SB[R_i]$ //Permuta utilizando as Caixas-S (S-BOX), saída 32 bits
    - $R_i = P(R_i)$ //Permuta utilizando P(Permutation Function), saída 32 bits
    - $R_i = L_{i-1}$ **XOR** $\mathbf{R_i}$ //Operação de OU-EXCLUSIVO entre Li-1 e Ri obtido em 5
  - P = (R16,L16) //O texto plano é a seqüência de bits invertida
  - P = IP-1(P) //Finalmente é feita a permuta inversa IP-1 do texto plano

Algoritmos de Bloco
DES - Segurança

- Como o algoritmo DES utiliza uma chave com 56 bits é possível existir 2^{56} chaves, o que inicialmente tornava inviável um ataque de força bruta ao DES
- Porém com o aumento de capacidade dos computadores o algoritmo tornou-se fraco
- Em 1998 o DES foi quebrado em 56 horas em um computador especialmente preparado
- Uma solução para este problema é executar o algoritmo DES mais de uma vez
- Uma proposta é o Triple DES que realiza uma Cifragem-Decifragem-Cifragem de uma mensagem desta forma é possível utilizar até 3 diferentes chaves para o algoritmo
- Uma outra possibilidade é o uso de um novo algoritmo, o que foi efetivamente realizado com o AES

Algoritmos de Bloco Triple DES (3DES)

- Neste caso o algoritmo utilizado é o DES, porém são utilizadas até três chaves
- Triples DES com duas chaves
  - Neste caso as seguintes operações são realizadas da seguinte forma
    - $C = E_{K1}(D_{K2}(E_{K1}(P)))$
    - $P = D_{K1}(E_{K2}(D_{K1}(c)))$
  - O custo de um ataque força-bruta no 3DES é da ordem de 2^{112}
  - Porem em 1981 foi feita uma proposta para realizar um ataque ao 3DES
- Triples DES com três chaves
  - Neste caso as seguintes operações são realizadas da seguinte forma
    - $C = E_{K3}(D_{K2}(E_{K1}(P)))$
    - $P = D_{K1}(E_{K2}(D_{K3}(C)))$
- O 3DES é utilizado em várias aplicações como por exemplo S/MIME (Secure/Multipurpose Internet Mail Extension)

AES (Advanced Encryption Standard)

- Em 1999 o NIST (National Institute of Standards and Technology) realizou uma revisão do padrão DES (FIPS PUB 46-3) e indicou que o DES deve ser utilizado apenas em sistemas legados e em seu lugar deveria ser utilizado o triple DES(3DES)
- Considerando o uso de 3 chaves o 3DES é forte o bastante, porém sua performance é bastante ineficiente para ser executado através de software
- Outro problema relacionado ao DES e ao 3DES é que ambos utilizam um bloco de 64 bits o que é ineficiente e reduz a segurança
- Em 1997 o NIST realizou uma chamada pública a fim de obter proposta para um novo padrão de criptografia avançado (Advanced Encryption Standard)
- O processo de avaliação foi finalizado em 2001. O algoritmo selecionado era conhecido como **Rijndael e** posteriormente ficou conhecido como **AES** (Advanced Encryption Standard)
- A tendência é que o 3DES seja substituído pelo AES, porém este processo deverá levar alguns anos e enquanto isto o 3DES continua como um algoritmo aprovado

Algoritmos de Bloco
AES Vantagens x Destavangatens

- Vantagens
  - Rápido em relação a outros algoritmos de bloco
  - Pode ser implementado em SmartCard utilizando pouco código e memória
  - Algumas operações são executadas em paralelo otimizando o seu desempenho
    - O uso das S-Box.
    - Expansão da chave enquanto são executadas operações que não dependem da mesma como ByteSub() e ShiftRow()
  - Como não utiliza operações aritméticas exige pouco processamento
  - Não reutiliza elementos já processados, apenas o estado (State) da rodada anterior
- Desvantagens
  - A operação inversa utiliza mais código e mais processamento, não sendo indicada para implementação em SmartCard
  - Em software a cifragem e a inserva utilizam diferentes códigos e tabelas, tornando a implementação um pouco mais complexa

Algoritmos de Bloco
AES x DES - Características

- No AES o comprimento do bloco e da chave são independentes
- É possível utilizar blocos e chaves de 128, 192 ou 256 bits conforme a tabela abaixo

PARÂMETROS				
	AES			DES (bits)
Tamanho da Chave (words/bytes/bits)	4/16/128	6/24/192	8/32/256	64
Tamanho do Bloco (words/bytes/bits)	4/16/128	4/16/128	4/16/128	64
Número de iterações	10	12	14	16
Tamanho da chave em cada Iteração (words/bytes/bits)	4/16/128	4/16/128	4/16/128	48
Tamanho da chave Expandida (words/bytes)	44/176	52/208	60/240	48

Algoritmos de Bloco
AES - Comparação

- A tabela abaixo mostra a comparação entre alguns algoritmos de bloco quanto ao seu funcionamento e as operações matemáticas utilizadas

Algoritmo	*Tamanho da Chave*	*Número de Rodadas*	*Operações Matemáticas*
AES	128, 192 ou 256 bits	10, 12, 14	XOR, S-Boxes fixas
DES	56 bits	16	XOR, S-Boxes fixas
3DES	112 ou 168 bits	48	XOR, S-Boxes fixas
IDEA	128 bits	8	XOR, adição, multiplicação
BLOWFISH	Variável até 448 bits	16	XOR, S-Boxes variáveis, adição
RC5	Variável até 2048 bits	Variável até 255	Adição, Subtração, XOR, rotação
CAST-128	40 até 128 bits	16	Adição, subtração, XOR, rotação, S-Boxes fixas

Algoritmos de Bloco
Modos de Operação

- O algoritmos de bloco possuem diferentes modos de operação
- Basicamente o modo de operação é uma técnica para aumentar a eficiência do algoritmo
- O NIST (FIPS 800-38A) definiu cinco diferentes modos de operação que cobrem todas as forma de operação dos algoritmos de bloco.
- Os modos de operação podem ser utilizados com qualquer algoritmo de bloco incluindo DES; TRIPLE DES e AES entre outros.

Algoritmos de Bloco
Modos de Operação

- Modos de Operação

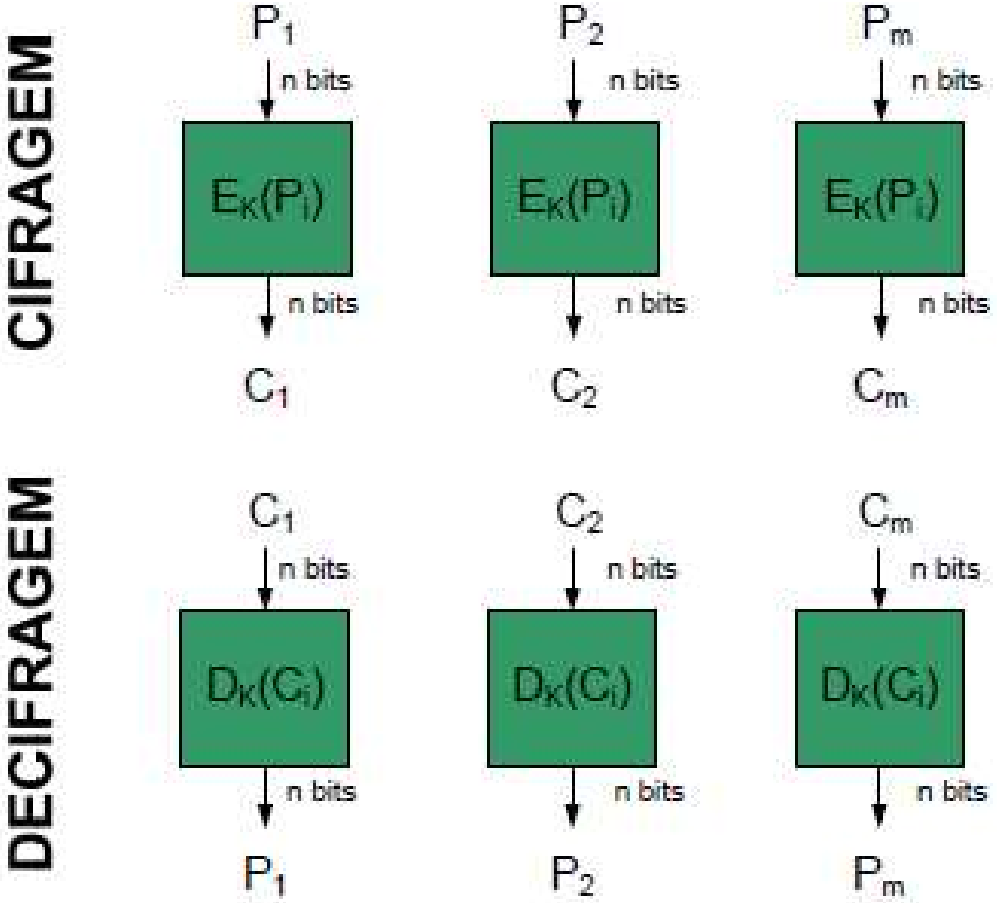
Modo	Descrição	Aplicações Típicas
Electronic Codebook (ECB)	Cada bloco de 64 bits do texto é cifrado de forma independente, utilizando a mesma chave. É o modo padrão de operação	Transmissão de valores únicos critografados (ex: Chave)
Cipher Block Chaining (CBC)	A entrada do algoritimo é o resultado de um XOR dos próximos 64 bits e dos últimos 64 bits cifrados	Transmissão de um bloco de dados, porém para a mesma chave, blocos iguais produzem saídas diferentes
Cipher Feedback Mode (CFB)	A entrada é processada com j bits de cada vez. Ao invés de utilizar os 64 bits (8 bytes). O último texto cifrado é utilizado como entrada para o algoritmo e fim de produzir um uma falsa saída que é então associada com uma parte de texto plano, através de uma operação XOR a fim de produzir o próximo texto a ser cifrado	Transmissão de uma stream de dados; Autenticação
Output Feedback Mode (OFB)	Semelhante ao CFB, porém a entrada para a próxima cifragem é a saída da cifragem anterior	Transmissão de uma stream de dados em um canal com ruídos (ex. Satélite)
Counter (CTR)	É realizada uma operação XOR com um contador cifrado. Este contador é incrementado a cada bloco subsequente	Transmissão de um bloco de dados; Útil para requisito de alta performance

Algoritmos de Bloco
Modos de Operação - ECB

- Electronic Codebook (ECB)
  - Os número total de bits a ser cifrado deve ser múltiplo do tamanho do bloco utilizado. Caso seja necessário deve ser feita a inserção de bits (padding)
  - Os blocos cifrados são independentes entre si
  - Blocos iguais no texto plano são repetidos na mensagem de forma cifrada e desta forma é mais fácil descobrir um padrão na criptografia, apresentando desta forma uma alta vulnerabilidade
  - Cifragem
    - $C_i = E_K(P_i)$ sendo i = 1, 2, …, m
  - Decifragem
    - $P_i = D_K(C_i)$ sendo i = 1, 2, …, m
  - Sendo:
    - P_i Texto Puro com n bits. O último bloco P_m pode sofrer inserção de bits
    - C_i Texto Cifrado com n bits
    - $E_K()$ Algoritmo de cifragem em bloco, utilizando a chave K
    - $D_K()$ Algoritmo de decifragem em bloco, utilizando a chave K

Algoritmos de Bloco
Modo de Operação - ECB

- Electronic Codebook (ECB)
  - $M = P_1 \,||\, P_2 \,||\, P_m$, sendo || a operação de concatenação de bits
  - $C = C_1 \,||\, C_2 \,||\, C_m$, sendo || a operação de concatenação de bits

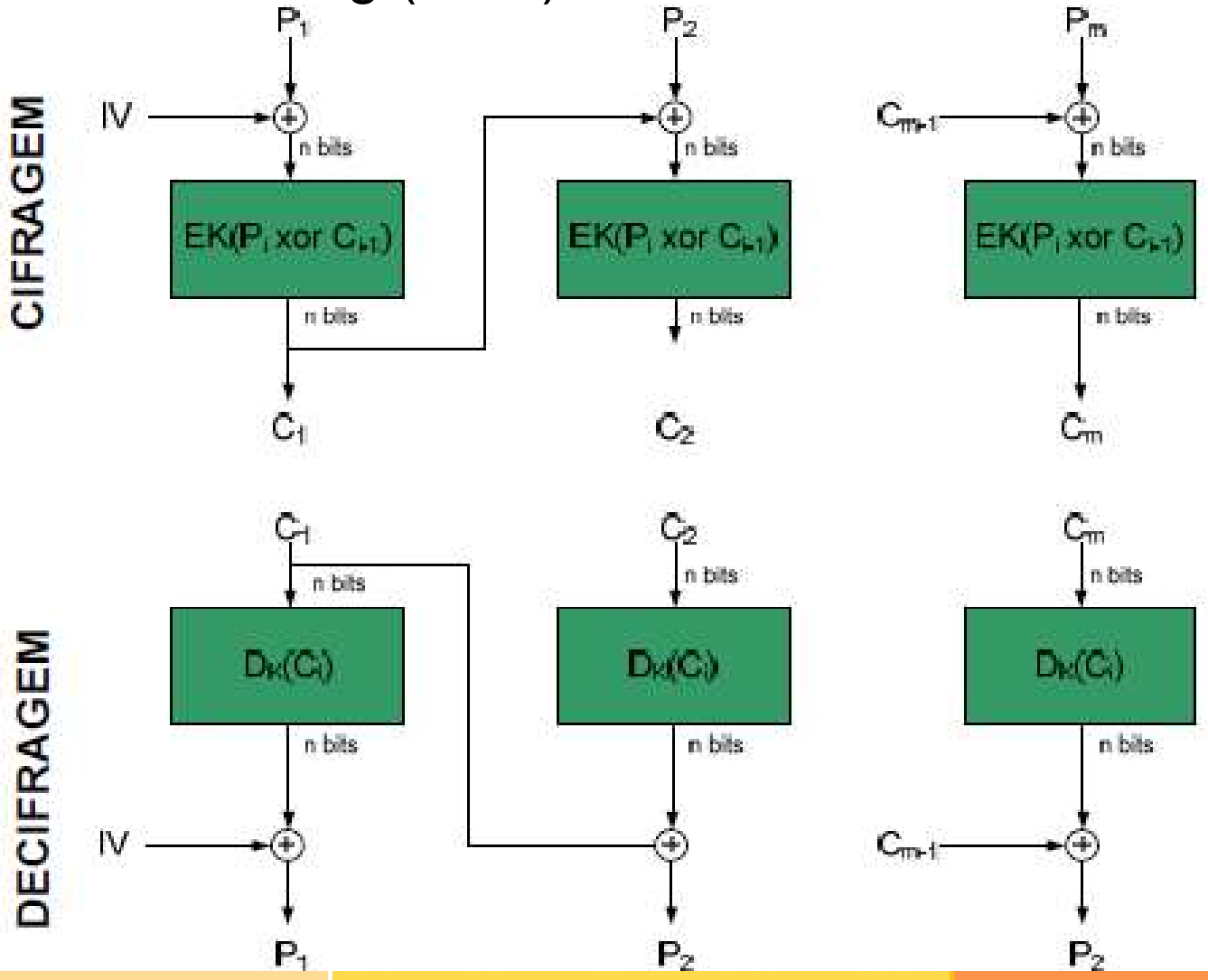


Algoritmos de Bloco
Modos de Operação - CBC

- Cipher Block Chaining (CBC)
  - A entrada do algoritmo é o texto plano associado ao texto cifrado do bloco anterior através de uma operação XOR
  - Desta forma os blocos são encadeados entre si e a saída do algoritmo é função não apenas do texto plano mas de todos os blocos cifrados anteriormente
  - A mesma chave é utilizada em todos os blocos
  - Neste caso há maior segurança, visto que o mesmo texto plano quando é cifrado produz diferentes resultados
  - No primeiro passo é necessário um vetor inicial (IV) que é então associado ao primeiro bloco do texto puro (P_1)
  - Cifragem
    - $C_0 = IV$
    - $C_i = E_K(P_i \oplus C_{i-1})$ sendo i = 1, 2, …, m
  - Decifragem
    - $C_0 = IV$
    - $P_i = D_K(C_i) \oplus C_{i-1}$ sendo i = 1, 2, …, m

Algoritmos de Bloco
Modos de Operação - CBC

- Cipher Block Chaining (CBC)



Algoritmos de Bloco
Modos de Operação - CFB

- Cipher Feedback Mode (CFB)
  - Este modo converte o algoritmo de bloco em um cifrador de fluxo (Stream)
  - Desta forma não é necessário acrescentar bits à mensagem original a fim de criar um número inteiro de blocos a fim de serem processados pelo algoritmo
  - Além disso é possível operar em tempo real, ou seja, se um fluxo de caracteres (8 bits) está sendo transmitido é possível cifrar cada caractere (8 bits)
  - Por exemplo, se o algoritmo utilizado for o DES, não é necessário montar blocos de 64 bits
  - No modo CFB as unidades de texto puro são encadeadas e desta forma o texto cifrado produzido é função de todo o texto precedente
  - Neste modo de operação apenas a função $E_K()$ é utilizada tanto para cifrar quanto para decifrar.

Algoritmos de Bloco
Modos de Operação - CFB

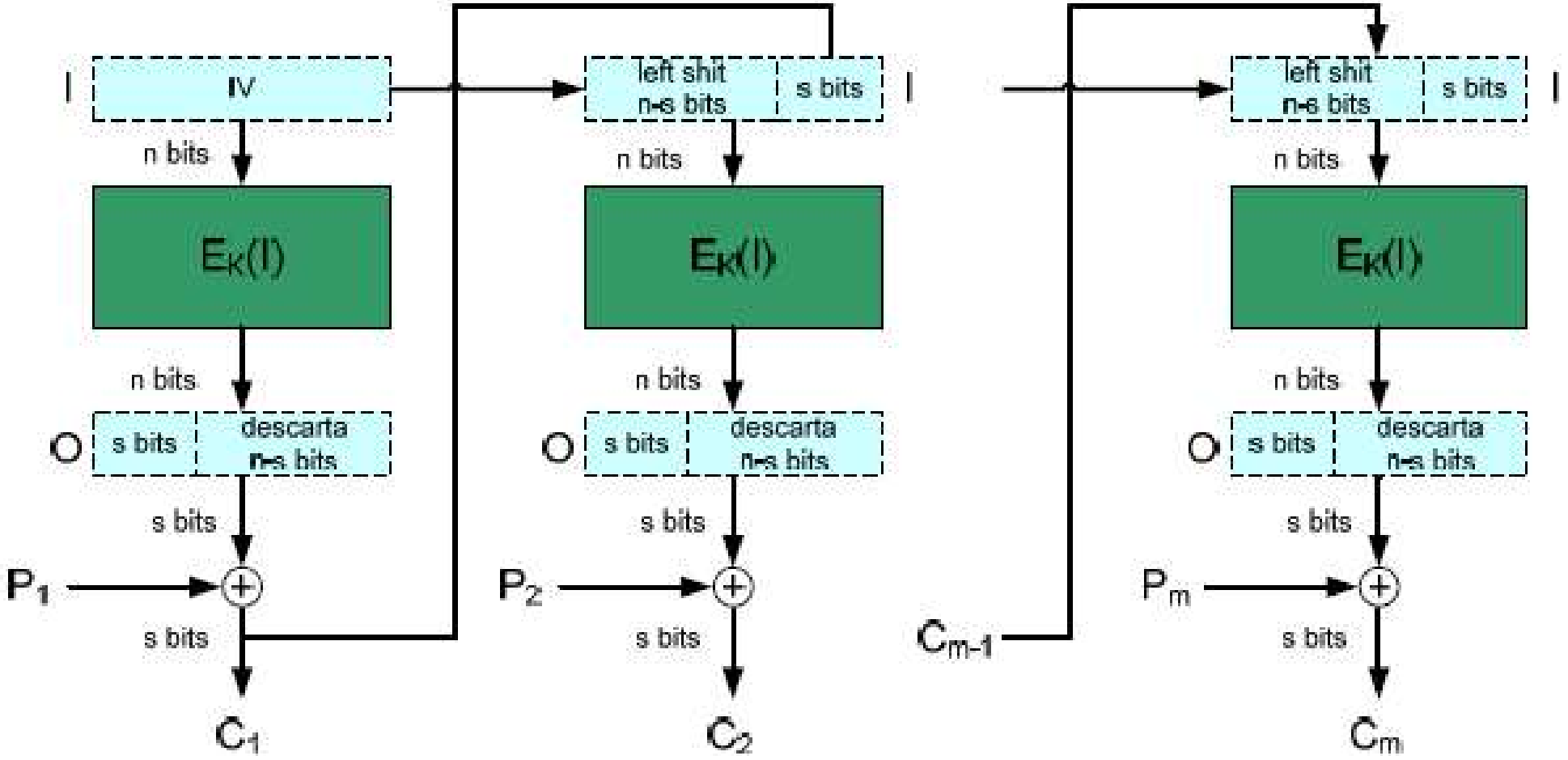
- Cipher Feedback Mode (CFB)
  - Cifragem
    - O tamanho do bloco do cifrador é n
    - s bits de uma mensagem que possui no total m bits serão cifrados
    - $I_0 = IV$; //entrada do cifrador (I_0) é inicializada com o vetor IV com n bits
    - Para $i = 1 \leq i \leq m$ temos:
    - $O_i = E_K(I_{i-1})$ //esta entrada é então cifrada (E_K) obtendo a saída O_i
    - $O_i = MSBs(O_i)$ //Somente os s bits mais significativos de Oi serão considerados
    - $C_i = P_i \oplus O_i$ //Operação XOR entre Oi com os s bits do texto plano (P_i)
    - A próxima entrada é calculada a partir dos n-s bits menos significativos da entrada anterior concatenados com os s bits cifrados
    - $I_i = LSB_{n-s}(I_{i-1}) \,||\, C_i$

Algoritmos de Bloco
Modos de Operação - CFB

- Cipher Feedback Mode (CFB)
  - Decifragem
    - O tamanho do bloco do cifrador é n
    - O texto plano (P_i) será recuperado a partir do texto cifrado (C_i)
    - $I_0 = IV$; //entrada do cifrador (I_0) é inicializada com o vetor IV com n bits
    - Para $i = 1 \leq i \leq m$ temos:
    - $O_i = E_K(I_{i-1})$ //esta entrada é então cifrada (E_K) obtendo a saída O_i
    - $O_i = MSBs(O_i)$ //Somente os s bits mais significativos de Oi serão considerados
    - $P_i = C_i \oplus O_i$ //Operação XOR entre Oi com os s bits do texto plano (P_i)
    - A próxima entrada é calculada a partir dos n-s bits significativos da entrada anterior concatenados com os s bits cifrados
    - $I_i = LSB_{n-s}(I_{i-1}) \,||\, C_i$

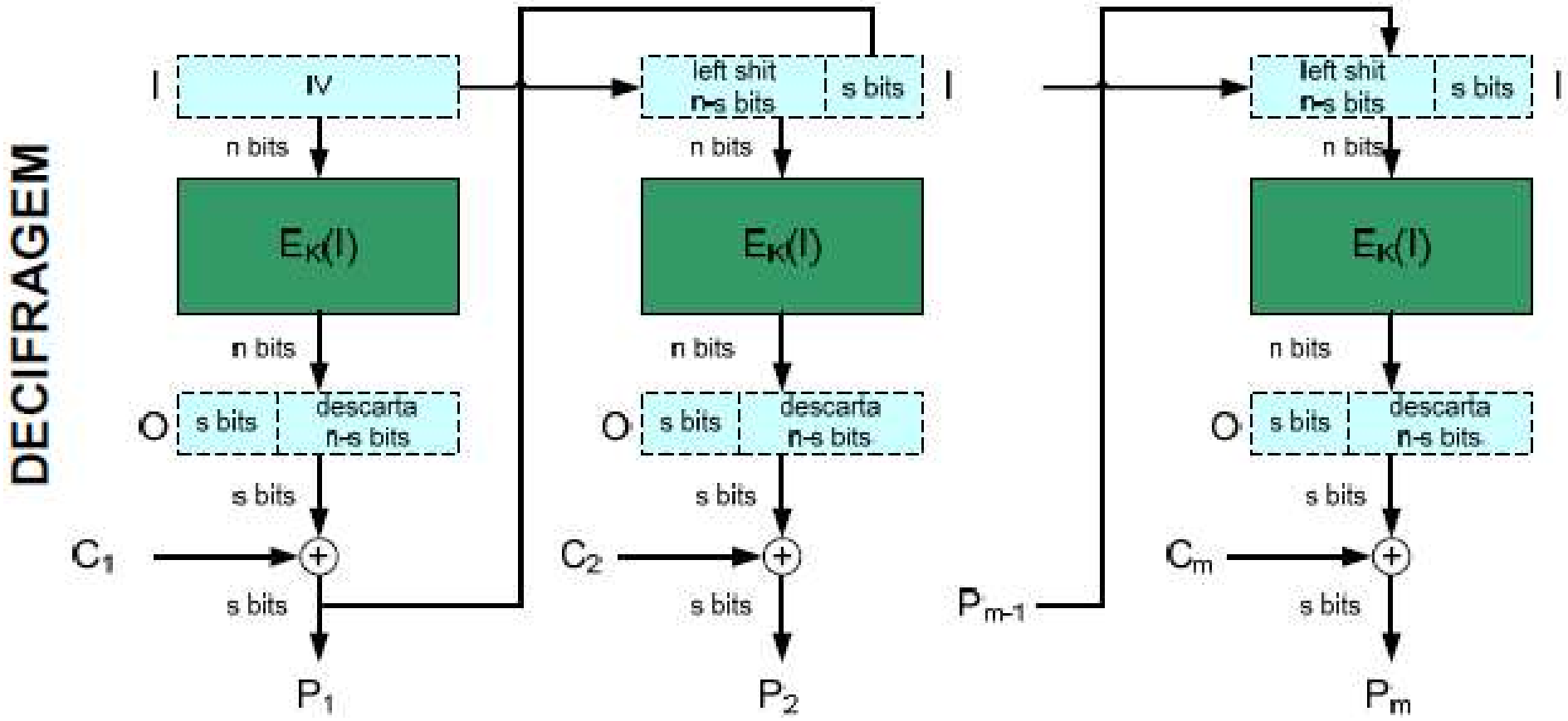
Algoritmos de Bloco
Modos de Operação - CFB

- Cipher Feedback Mode (CFB) - Cifragem



Algoritmos de Bloco
Modos de Operação - CFB

- Cipher Feedback Mode (CFB) - Decifragem



Algoritmos de Bloco
Modos de Operação - OFB

- Output Feedback Mode (OFB)
  - Este modo também converte o algoritmo de bloco em um cifrador de fluxo (Stream)
  - Seu funcionamento é similar ao modo CFB
  - A diferença básica é que ao invés de inserir na entrada seguinte o texto cifrado (C_i), os s bits são aqueles obtidos na saída anterior cifrada (O_{i-1})
  - Uma vantagem deste modo é que os bits incorretos na transmissão não são propagados, sendo útil em transmissões onde não é possível a retransmissão, como por exemplo, ondas de rádio
  - Se ocorrer um erro na transmissão de C_i, somente P_i será afetado, pois C_i não interfere no cálculo da entrada I_{i+1}.
  - Uma desvantagem é que este modo é mais vulnerável a um ataque de modificação quando comparado com o modo CFB.
  - O complemento de um bit no texto cifrado causa o complemento do bit correspondente no texto puro obtido. Este problema agrava-se quanto menor for o tamanho s

Algoritmos de Bloco
Modos de Operação - OFB

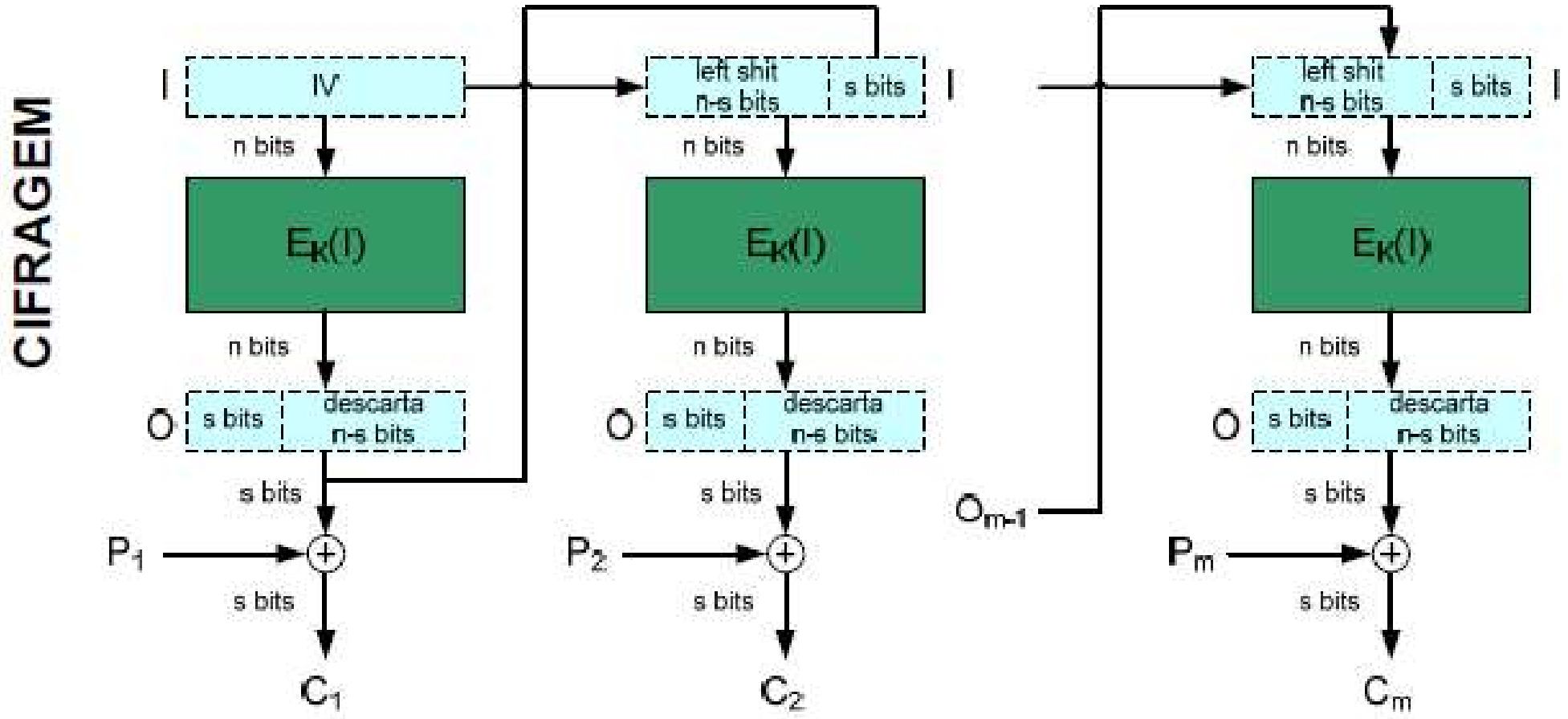
- Output Feedback Mode (OFB)
  - Cifragem
    - O tamanho do bloco do cifrador é n
    - s bits de uma mensagem que possui no total m bits serão cifrados
    - $I_0 = IV$; //entrada do cifrador (I0) é inicializada com o vetor IV com n bits
    - Para $i = 1 \leq i \leq m$ temos:
    - $O_i = E_K(I_{i-1})$ //esta entrada é então cifrada (EK) obtendo a saída Oi
    - $O_i = MSBs(O_i)$ //Somente os s bits mais significativos de Oi serão considerados
    - $C_i = P_i \oplus O_i$ //Operação XOR entre Oi com os s bits do texto plano (Pi)
    - A próxima entrada é calculada a partir dos n-s bits menos significativos da entrada anterior concatenados com os s bits obtidos na saída do cifrador
    - $I_i = LSB_{n-s}(I_{i-1}) \,||\, O_i$

Algoritmos de Bloco
Modos de Operação - OFB

- Output Feedback Mode (OFB)
  - Decifragem
    - O tamanho do bloco do cifrador é n
    - O texto plano (P_i) será recuperado a partir do texto cifrado (C_i)
    - $I_0 = IV$; //entrada do cifrador (I_0) é inicializada com o vetor IV com n bits
    - Para $i = 1 \leq i \leq m$ temos:
    - $O_i = E_K(I_{i-1})$ //esta entrada é então cifrada (E_K) obtendo a saída O_i
    - $O_i = MSBs(O_i)$ //Somente os s bits mais significativos de Oi serão considerados
    - $Pi = Ci \oplus Oi$ //Operação XOR entre O_i com os s bits do texto plano (P_i)
    - A próxima entrada é calculada a partir dos n-s bits menos significativos da entrada anterior concatenados com os s bits obtidos na saída do cifrador
    - $I_i = LSB_{n-s}(I_{i-1}) \,||\, O_i$

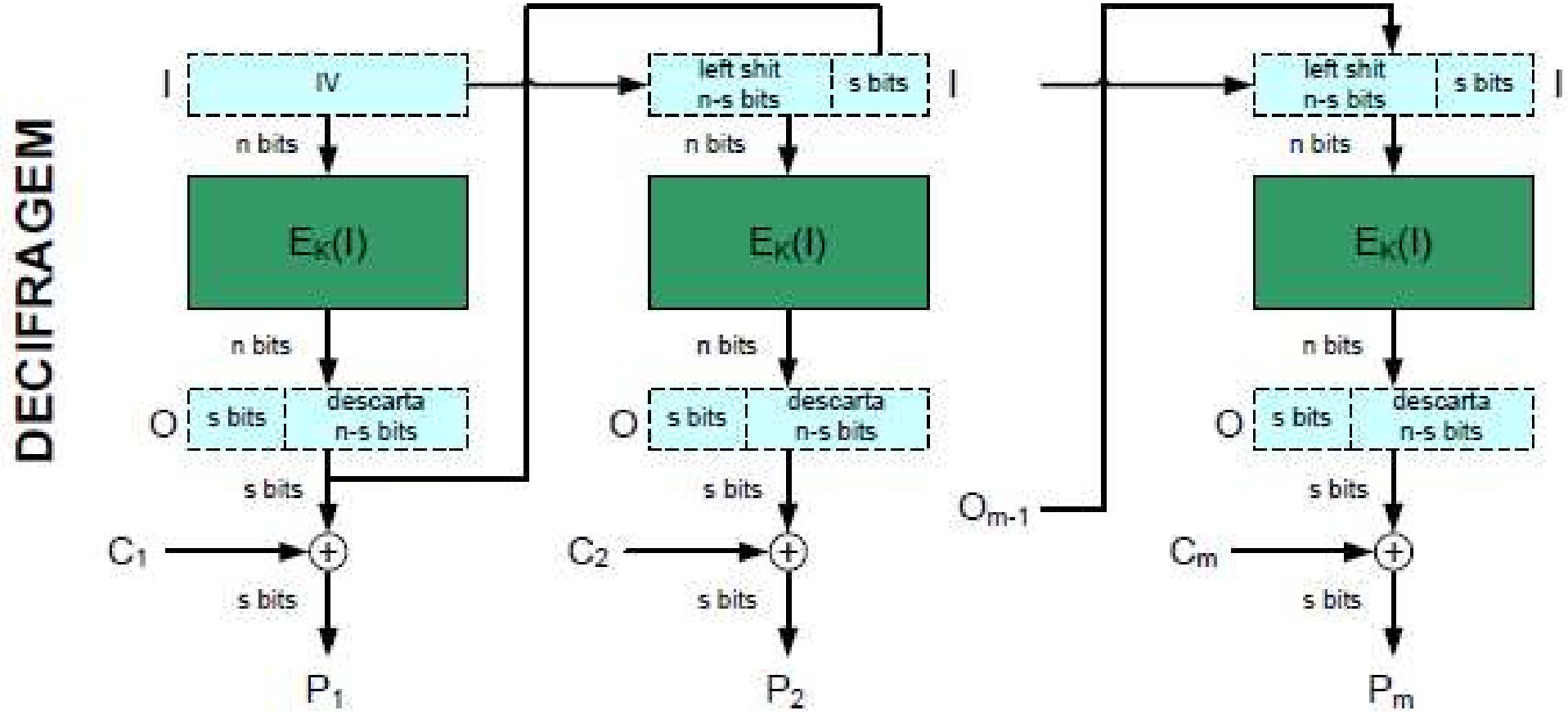
Algoritmos de Bloco
Modos de Operação - OFB

- Output Feedback Mode (OFB) - Cifragem



Algoritmos de Bloco
Modos de Operação - OFB

- Output Feedback Mode (OFB) - Decifragem



Algoritmos de Bloco
Modos de Operação - CTR

- Counter Mode (CTR)
  - Neste modo um contador, cujo tamanho em bits, deve ser igual ao tamanho do bloco é inicializado com um valor
  - Este contador é então cifrado e finalmente é realizada uma operação XOR do contador cifrado com o texto puro
  - Um requisito é que os valores do contador devem ser diferentes, neste caso, o contador é acrescido de 1 nos próximos blocos
  - Para decifrar o mesmo contador utilizado inicialmente é cifrado novamente é o resultado.
  - Finalmente é feita uma operação XOR entre o contador inicialmente cifrado e o último obtido. O resultado é o texto puro
  - Como não usa a técnica de encadeamento vários blocos podem ser cifrados e decifrados em paralelo
  - Assim como o CFB e OFB é um modo simples pois utilizada apenas o algoritmo de cifragem
  - Este modo poder ser utilizado por exemplo para cifrar o número de seqüência de uma transmissão utilizando IPSec

Algoritmos de Bloco
Modos de Operação - CTR

- Counter Mode (CTR)
  - Cifragem
    - I_1 = IV //Contador é inicializado
    - Para i = 1 ≤ i ≤ m temos:
    - $O_i = E_K(I_i)$ //O contador é então cifrado (E_K) obtendo a saída O_i
    - $C_i = P_i \oplus O_i$ //Operação XOR entre O_i com os bits do texto plano (P_i)
    - $I_i = I_{i+1}$
  - Decifragem
    - I_1 = IV //Contador é inicializado
    - Para i = 1 ≤ i ≤ m temos:
    - $O_i = E_K(I_i)$ //O contador é então cifrado (E_K) obtendo a saída O_i
    - $P_i = C_i \oplus O_i$ //Operação XOR entre O_i com os bits do texto cfrado (C_i)
    - $I_i = I_{i+1}$

Algoritmos de Bloco
Modos de Operação - CTR

- Counter Mode (CTR)