INTRODUÇÃO À CRIPTOGRAFIA

Terminologia

Criptografia ≡ Arte de escrita secreta, convencional, por meio de sinais, cifras e abreviaturas. (gr.: *kripto* = secreto, *grapho* = escrita)

Criptograma ≡ Texto redigido em escrita cifrada.

Criptógrafo ≡ Equipamento de criptografia (cod./decod.).

Criptoanalista ≡ Profissional especialista em criptografia.

Cifragem ≡ Codificação de mensagens criptográficas.

Decifragem ≡ Decodificação de mensagens criptográficas.

Criptografia em Sistemas de Comunicação

Segurança e privacidade das informações transmitidas através dos sistemas de comunicação de dados. Proteção de informações (confidenciais) entre origem e destino.

Aplicações em Sistemas de Comunicação

- Segurança em redes de computadores.
- Segurança em transmissões de dados ponto a ponto (modems, rádio digital).
- Sigilo em transações bancárias.
- Autenticação e assinatura eletrônicas.
- Serviços por assinatura (Pay-TV, etc.).
- Internet

Medidas de Segurança em Redes

- **Associada a cada link de comunicação:** Processo de codificação/decodificação é realizado entre cada 2 nós.

  Vantagem: A identificação do originador e do destinatário são protegidas.

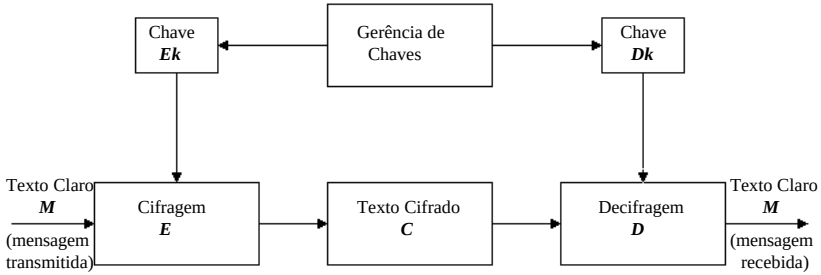
  Desvantagem: Os nós devem ter acesso restrito.

- **Ponto a Ponto:** Processo de codificação/decodificação é realizado somente pelo gerador/destinatário.

  Vantagem: Um usuário pode usar um esquema de cifragem sem afetar os demais.

  Desvantagem: As identificações do originador e do destinatário não podem estar cifradas.

Conceitos Básicos da Cifragem de Dados



Processo de Cifragem: A função da chave de cifragem E_k é realizar alguma operação matemática sobre o texto em claro, de modo a transformá-lo em texto cifrado.

$$C = E_k(M)$$

onde

$M \equiv$ texto em claro,
$E_k \equiv$ chave de cifragem
$C \equiv$ texto cifrado.

Processo de Decifragem: O algoritmo de cifragem precisa ser inversível, isto é, deve existir uma chave de decifragem D_k tal que o texto cifrado submetido a esta é convertido em texto em claro.

$$M = D_k(C)$$

Tipos de Sistemas de Cifragem

- **Sistema Criptográfico Simétrico:** $E_k = D_k$.

- **Sistema Criptográfico Assimétrico:** $E_k \neq D_k$.

Características Desejáveis:

- O algoritmo de cifragem/decifragem deve ser simples de implementar.

- O tempo de cifragem/decifragem deve ser relativamente pequeno, permitindo altas taxas de transmissão dos dados.

- tempo necessário para a quebra das chaves deve ser proibitivamente grande quando comparado ao valor da informação que está sendo protegida.

A teoria de obtenção do par de chaves sem autorização do proprietário é denominada *CRIPTOANÁLISE*.

Sistemas Criptográficos Simétricos

Cifragem por Transposição: Rearranjo de cada caractere do texto em claro de modo a produzir o texto cifrado.

- **Transposição Reversa:**

texto em claro: *SEGURANÇA DE UMA REDE LOCAL*

texto cifrado: *AÇNARUGES ED AMU EDER LACOL*

- **Transposição por Padrão Geométrico:**

Ex.: Arranjo bidimensional ou matriz (3 x 4).

texto em claro: *CRIPTOGRAFIA*

matriz 3 x 4:

1	2	3	4
C	*R*	*I*	*P*
T	*O*	*G*	*R*
A	*F*	*I*	*A*

texto cifrado pela leitura das colunas 2 4 1 3:

ROFPRACTAIGI

- **Transposição de Colunas:** O texto em claro é colocado segundo a direção vertical. As colunas são rearranjadas segundo uma ordem preestabelecida e o texto cifrado é obtido pela leitura dos caracteres na direção horizontal.

Ex.: matriz 5 x 6, arranjo {3,5,2,4,6,1}.

M: *CIFRAGEM POR MATRIZ BIDIMENSIONAL*

C	*G*	*R*	*I*	*I*	*I*
I	*E*	*M*	*Z*	*M*	*O*
F	*M*	*A*	*B*	*E*	*N*
R	*P*	*T*	*I*	*N*	*A*
A	*O*	*R*	*D*	*S*	*L*

R	*I*	*G*	*I*	*I*	*C*
M	*M*	*E*	*Z*	*O*	*I*
A	*E*	*M*	*B*	*N*	*F*
T	*N*	*P*	*I*	*A*	*R*
R	*S*	*O*	*D*	*L*	*A*

C: *RIGIIC MMEZOI AEMBNF TNPIAR RSODLA*

Para uma matriz com *n* colunas, estas podem ser arranjadas de *n*! maneiras diferentes. Para o exemplo, existem 6! = 720 possibilidades.

- **Permutação Periódica:** Constitui um método simples e eficiente que realiza a permutação dos caracteres do texto em claro de período *d*. Se *f* é a função de permutação de um bloco de *d* caracteres, então, a chave de cifragem depende de *f* e do período *d*.

Para o texto em claro:

$$M = m_1, m_2, \cdots, m_d, m_{d+1}, \cdots, m_{2d}, \cdots$$

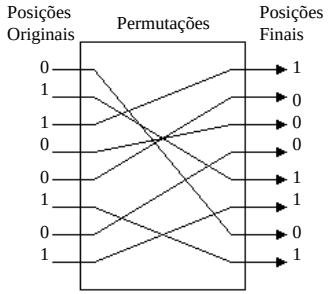
onde m_i são os caracteres. O criptograma resultante é dado por:

$$C = E_k(M) = m_{f(1)}, m_{f(2)}, \cdots, m_{f(d)}, m_{d+f(1)}, \cdots m_{d+f(d)}, \cdots$$

Ex.: *d* = 5; *f*(*i*) = {3,5,1,4,2}; *M* = "*FLAMENGO CAMPEÃO*"

C = "*AEFML OANCG EOMÃP*"

- Permutação Periódica Binária:



d = 8; *f*(*i*) = {3,5,4,7,2,8,1,6}

Cifragem por Substituição: Troca de caracteres do texto em claro por outros caracteres, de forma reversível.

- **Substituição Simples:** Cada caractere da mensagem é trocado por um caractere correspondente; o mapeamento escolhido associa o criptograma à mensagem.

- **Substituição Homofônica:** Cada caractere da mensagem é cifrado em uma variedade de caracteres; é usado um mapeamento que associa criptogramas diferentes a uma única mensagem.

- **Substituição Polialfabética:** São usados múltiplos alfabetos de cifragem; é usado um mapeamento que associa um único criptograma a uma dada mensagem, porém, este mapeamento pode variar com o tempo.

- **Substituição Polígrama:** Este método permite a substituição arbitrária de um grupo de caracteres do texto em claro simultaneamente. Os esquemas de cifragem por substituição, em geral, utilizam este método, pois, tem a vantagem de destruir a freqüência de caracteres.

Ex.: Cifrador de *César* (substituição simples)

mapeamento:

$$\begin{aligned} A &\rightarrow D \\ B &\rightarrow E \\ &\vdots \\ Y &\rightarrow B \\ Z &\rightarrow C \end{aligned}$$

M = "*JULIUS CESAR*" $\rightarrow$ *C* = "*MXOLXV FHVDU*"

- Para deslocamentos iniciais do alfabeto diferentes de 3, obtém-se outros esquemas de substituição.

- A análise estatística do criptograma quebra, facilmente, a chave deste esquema de cifragem.

Ex.: Substituição homofônica

- As letras do alfabeto são mapeadas em números de 00 a 99.
- Cada letra é associada a mais de um número.

A	17	19	34
B	03	10	55
C	06	22	31
D	18	21	92
E	67	77	84
F	35	56	66
G	14	30	88
H	41	59	83
I	93	94	99

J	27	39	75
K	05	29	97
L	07	44	76
M	72	73	90
N	02	09	15
O	01	11	23
P	33	62	91
Q	08	24	81
R	36	53	69

S	00	70	71
T	37	58	64
U	04	12	98
V	16	20	47
W	25	45	51
X	38	61	65
Y	13	49	80
Z	26	50	95

M = "*PRESERVE A NATUREZA*"

C = "*33 36 67 00 77 69 20 84 34 02 19 37 12 53 77 50 17*"

- Para uma dada mensagem existem vários criptogramas associados.
- O primeiro uso da cifragem homofônica ocorreu na correspondência entre o Duque de Mântua e Simeone de Crema em 1401.

Ex.: Cifrador de *Vigenère* (polialfabético)

- A chave *K* é especificada por uma sequência de caracteres:

$$K = k_1 k_2 \cdots k_n$$

onde cada k_i forma um cifrador de César diferente.

K = "*CRIPT* "; *M* = "*TEXTO*"

$\Rightarrow$ *C* = "*VVFIH*"

- Para o caso em que a chave tem comprimento infinito, este cifrador é denominado de *Vernam*. Este sistema, apesar de ser ineficiente (tamanho da chave, sincronização), é bastante seguro, sendo utilizado na *linha vermelha* entre Washington e Moscou.

Ex.: Cifrador de *Playfair* (Polígramo)

- Foi inventado por um amigo de Lyon Playfair, Charles Wheatstone, e foi usada pelos ingleses na Segunda Guerra Mundial.

- A chave consiste de uma matriz 5 x 5, contendo todas as letras do alfabeto exceto o *J* :

H	*A*	*R*	*P*	*S*
I	*C*	*O*	*D*	*B*
E	*F*	*G*	*K*	*L*
M	*N*	*Q*	*T*	*U*
V	*W*	*X*	*Y*	*Z*

- Cada par de caracteres m_1m_2 é cifrado de acordo com as seguintes regras:

a) Se m_1 e m_2 estiverem em linhas e colunas diferentes, então c_1 e c_2 formam um retângulo em conjunto com m_1 e m_2. Além disso, m_1 está na mesma linha de c_1.

b) Se m_1 e m_2 estiverem na mesma linha, então os caracteres cifrados c_1 e c_2 são justamente os caracteres à direita de m_1 e m_2, onde a primeira coluna é considerada à direita da última.

c) Se m_1 e m_2 estiverem na mesma coluna, então os caracteres cifrados c_1 e c_2 são justamente os caracteres abaixo de m_1 e m_2, onde a primeira linha é considerada estar abaixo da última.

d) Se $m_1 = m_2$, a letra *X* é inserida no texto em claro para evitar este problema.

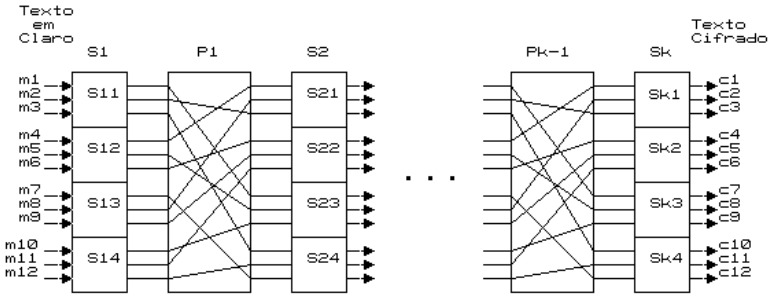
e) Se o texto em claro tiver um número ímpar de caracteres, a letra *X* é adicionada ao final do texto.

M = "*SOSSEGO*" $\rightarrow$ *SO-SS-EG-O* $\rightarrow$ *SO-SX-SE-GO*

C = "*RBRZHLQG*"

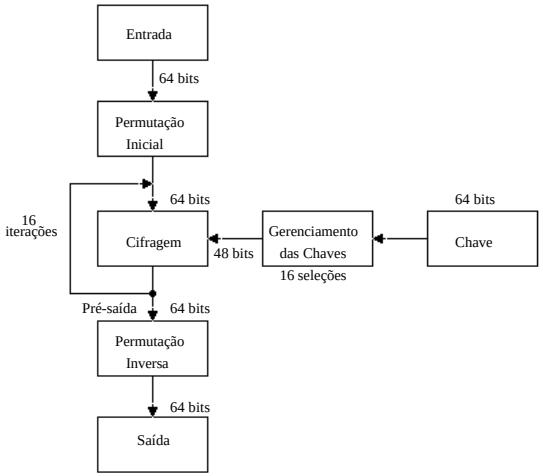
Cifragem de Produto: Envolve tanto permutações como substituições para produzir um criptograma. Uma cifragem de produto consiste na aplicação sucessiva de uma seqüência de *n* funções de cifragem $f_1, f_2, \ldots, f_n$, onde cada f_i pode ser uma cifragem de permutação *P*, ou uma cifragem de substituição *S*.

Ex.: Cifragem de produto aplicado a um conjunto de 12 bits, divididos em 4 blocos de 3 bits, cada um submetido a um processo de substituição. Os 12 bits resultantes são embaralhados por um bloco de permutação, sendo entregues novamente para uma fase de substituição.

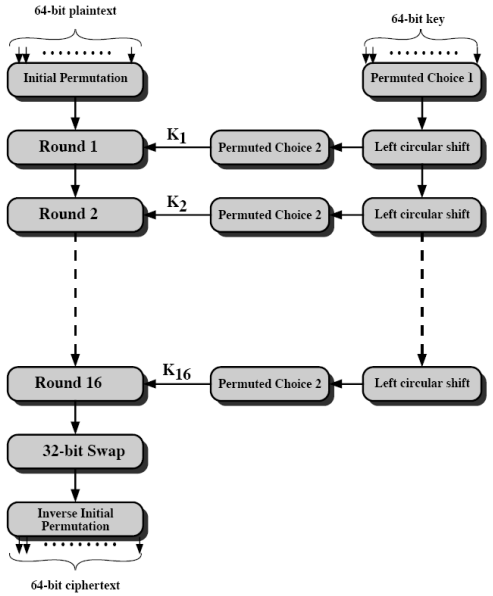


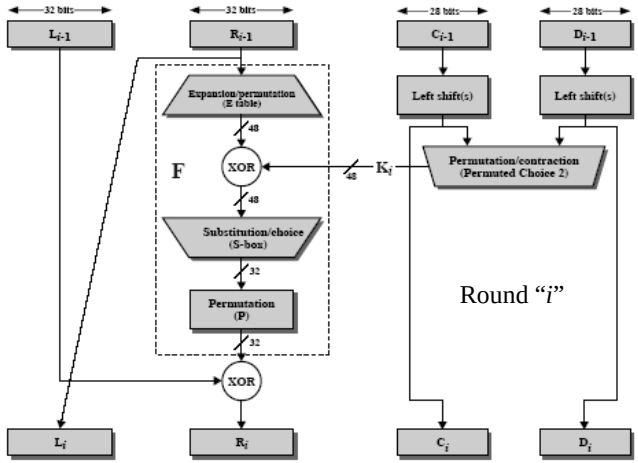
Padrão DES - "Data Encription Standard"

- Criado em 1971 pelo National Bureau of Standards dos EUA.
- Converte blocos de texto em claro de 64 bits em criptogramas também de 64 bits.
- Chave única (cifragem/decifragem) de 56 bits + 8 bits de paridade.
- Após uma permutação inicial, são realizadas 16 iterações de uma função que combina substituição e transposição (cifragem de produto). O criptograma é obtido após uma última permutação que é o inverso da permutação inicial.



A figura abaixo ilustra o as funções do sistema DES em forma de diagrama de blocos:





Processo de Cifragem DES

- Inicialmente é realizada uma permutação inicial (IP) sobre um bloco de 64 bits de entrada T, fornecendo $T_0 = \mathrm{IP}(T)$.

Posição original dos bits	Permutação Inicial IP — Nova posição dos bits							
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

- O bloco permutado T_0 é dividido em 2 blocos de 32 bits: L_0 e R_0, onde

$$L_0 = t_{58}t_{50}\cdots t_{16}t_8$$
$$R_0 = t_{57}t_{49}\cdots t_{15}t_7$$

- A primeira iteração consiste em realizar a seguinte operação de substituição:

$$L_1 = R_0$$
$$R_1 = L_0 \oplus f(R_0, K_1)$$

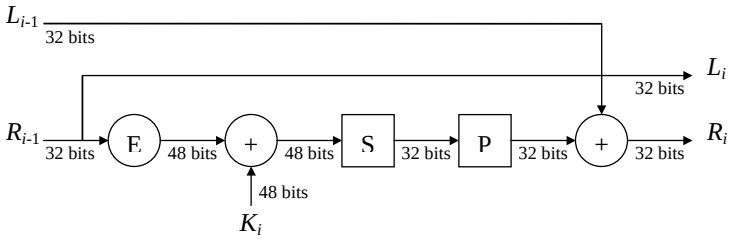
onde o símbolo $\oplus$ representa a operação soma módulo-2 (ou-exclusivo) bit a bit, e K_1 é uma chave de 48 bits escolhida a partir da chave geral de 56 bits.

- As demais 15 iterações seguem a mesma filosofia da primeira, isto é

$$L_i = R_{i-1}$$
$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$$

onde $1 \le i \le 16$.

- A função $f(R_{i-1}, K_i)$ denota a relação funcional composta por uma tabela de expansão de 32 para 48 bits, *E-table*, uma caixa de substituição, *S-box*, e uma tabela de permutação, *P-table*.



- O bloco R_{i-1} é permutado e expandido de 32 para 48 bits, gerando o bloco $E(R_{i-1})$, conforme a tabela a seguir. Note que alguns bits se repetem.

Posição final dos bits	E-Table: Posição inicial dos bits					
1-6	32	1	2	3	4	5
7-12	4	5	6	7	8	9
13-18	8	9	10	11	12	13
19-24	12	13	14	15	16	17
25-30	16	17	18	19	20	21
31-36	20	21	22	23	24	25
37-42	24	25	26	27	28	29
43-48	28	29	30	31	32	1

- Em seguida é feita a adição módulo-2 entre $E(R_{i-1})$ e K_i, e o resultado é organizado em 8 blocos de 6 bits:

$$E\left(R_{i-1}\right) \oplus K_i = B_1 B_2 \cdots B_8$$

- Cada bloco B_j de 6 bits é transformado em um bloco de 4 bits, usando a função de substituição S_j.

  Considere o bloco $B_1 = b_1\ b_2\ b_3\ b_4\ b_5\ b_6$ usado para selecionar S_1. O inteiro correspondente aos bits b_1 e b_6 selecionam uma linha na tabela, enquanto que o inteiro correspondentes aos bits b_2, b_3, b_4, e b_5 selecionam uma coluna na tabela.

Por exemplo, se $B_1 = 110001$, então S_1 retornará o valor na linha 3 e coluna 8, que neste exemplo é o inteiro 5, que é representado pela seqüência binária 0101. Assim,

$$S_j(B_j) = S_j^{b_1 b_6}(b_2 b_3 b_4 b_5)$$

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

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

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

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

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

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

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

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

- A seguir os 8 blocos de 4 bits são justapostos, resultando em um único bloco de 32 bits que é, então, permutado segundo a *P-Table*.

Posição original dos bits	**P-Table** Nova posição dos bits			
1-4	16	7	20	21
5-8	29	12	28	17
9-12	1	15	23	26
13-16	5	18	31	10
17-20	2	8	24	14
21-24	32	27	3	9
25-28	19	13	30	6
29-32	22	11	4	25

- Finalmente, o resultado é somado à L_{i-1} para se obter R_i.

- Todo o procedimento anterior de permutação e substituição é repetido por 16 vezes, e ao final, após a obtenção de $L_{16}R_{16}$, que totaliza 64 bits, é realizada a permutação final IP^{-1}, que é justamente a inversa de IP.

Posição original dos bits	**Permutação Final IP^{-1}** Nova posição dos bits							
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

Seleção de Chaves Intermediárias

- As chaves intermediárias K_i de 48 bits são derivadas da chave mestra K de 64 bits, que na realidade dispõe de 56 bits úteis, mais 8 bits de paridade nas posições 8, 16, … , 64.

  - O bloco *Permuted choice*-1, PC-1, realiza uma verificação dos bits de paridade, e caso haja confirmação, estes são eliminados, restando os 56 bits efetivos da chave mestra.

  - Então, o bloco de 56 bits sofre o processo de permutação PC-1, dividindo-o em dois blocos de 28 bits, C_0 e D_0, segundo a tabela:

Posição original dos bits	**PC-1** Nova posição dos bits da chave							
1-7	57	49	41	33	25	17	9	**Bloco**
8-14	1	58	50	42	34	26	18	**Esquerdo**
15-21	10	2	59	51	43	35	27	C_0
22-28	19	11	3	60	52	44	36	
1-7	63	55	47	39	31	23	15	**Bloco**
8-14	7	62	54	46	38	30	22	**Direito**
15-21	14	6	61	53	45	37	29	D_0
22-28	21	13	5	28	20	12	4	

- Os blocos *Left-shift* realizam um número de deslocamentos à esquerda, dependendo do índice i da chave parcial a ser obtida, nos blocos C_{i-1} e D_{i-1}. Como resultado são obtidos os novos blocos C_i e D_i.

Número da iteração	**Número de deslocamentos à esquerda**
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

- Cada bloco justaposto C_iD_i, é permutado através da tabela PC-2, fornecendo, assim, cada uma das chaves parciais K_i.

Posição original dos bits	**PC-2** Nova posição dos bits das chaves					
1-6	14	17	11	24	1	5
7-12	3	28	15	6	21	10
13-18	23	19	12	4	26	8
19-24	16	7	27	20	13	2
25-30	41	52	31	37	47	55
31-36	30	40	51	45	33	48
37-42	44	49	39	56	34	53
43-48	46	42	50	36	29	32

Resumo do Algoritmo DES

1. Especifique 56 bits, que adicionados a 8 bits de paridade formam a chave mestra K.

2. Construa a partir de K as 16 chaves intermediárias: $K_1, K_2, \ldots, K_{16}$.

3. Receba um bloco de 64 bits a ser cifrado (decifrado).

4. Usando a permutação inicial IP, obtenha os blocos L_0 e R_0.

5. Faça $i = 1$.

6. Para $1 \leq i \leq 16$, expanda R_{i-1} para 48 bits através da seleção E.

7. Faça $B = E(R_{i-1}) \oplus K_i$. (Para a decifragem use K_{17-i})

8. Divida B em blocos de 6 bits e use as funções S_i, de modo a gerar um novo bloco A de 32 bits.

9. Faça a permutação $P(A)$.

10. Obtenha $R_i = P(A) \oplus L_{i-1}$.

11. Defina $L_i = R_{i-1}$.

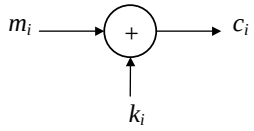
12. Faça $i = i + 1$. Se $i \leq 16$, volte ao passo 6. Senão, continue.

13. Use a inversa da permutação inicial IP^{-1}, gerando o criptograma C de 64 bits.

14. Enquanto houver dados a serem cifrados (decifrados), retorne ao passo 3.

Criptografia Seqüencial

- Conceitualmente, a criptografia seqüencial ideal constitui-se em um cifrador de *Vernam* binário, onde cada bit de entrada m_i é modificado através de uma operação de adição módulo-2, com uma sequência aleatória de comprimento infinito k_i.

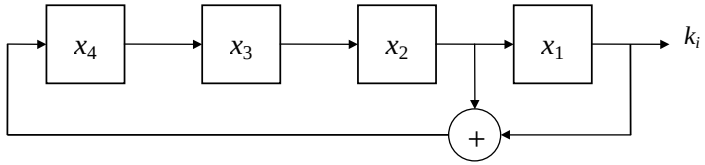


- No entanto, a geração de seqüências verdadeiramente aleatórias, sincronizadas no transmissor e receptor, não é fácil de se implementar na prática. Uma solução alternativa é a utilização de seqüências pseudo-aleatórias (PN).

- Geradores PN são facilmente construídos através de registros de deslocamento binários. Um gerador de 2^n - 1 bits, requer n estágios (flip-flops).

- Se uma seqüência PN é implementada através de um registro de deslocamento de 50 estágios, comandado a uma taxa de *clock* de 1 MHz, a seqüência resultante se repetirá a cada 2^{50} - 1 microsegundos, ou seja, a cada 35 anos. Com o advento dos circuitos integrados LSI, tornou-se possível usar 100 estágios ou mais, o que faz com que a seqüência se repita apenas a cada 4×10^{16} anos!

Ex: Para o registro de deslocamento linear abaixo, qual é a seqüência de saída resultante, considerando o estado inicial (x_4, x_3, x_2, x_1) = (1 0 0 0)?



Solução:

A sequência de estados de saída é obtida a partir da recorrência:

$$x_k(i) = x_{k+1}(i-1) \qquad k = 1,2,3$$
$$x_4(i) = x_1(i-1) \oplus x_2(i-1)$$

Logo,

i	x_4	x_3	x_2	$x_1 = k_i$
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	1	0	0	1
5	1	1	0	0
6	0	1	1	0
7	1	0	1	1
8	0	1	0	1
9	1	0	1	0
10	1	1	0	1
11	1	1	1	0
12	1	1	1	1
13	0	1	1	1
14	0	0	1	1
15	0	0	0	1
1	1	0	0	0

Segurança do Cifrador Seqüencial PN

O esquema criptográfico que usa um registro de deslocamento linear para gerar a seqüência chave é bastante vulnerável a ataques.

Um criptoanalista precisa de apenas $2n$ bits de texto em claro e, do texto cifrado correspondente, para determinar as conexões de realimentação, o estado inicial do registro, e toda a seqüencia do código. Em geral, $2n$ é um número muito pequeno em relação à 2^n -1.

Para o nosso exemplo anterior, imagine que um criptoanalista não sabe nada sobre as conexões internas do registro de deslocamento, porém obtém $2n$ =8 bits da mensagem cifrada e o texto em claro correspondente, onde o bit mais a esquerda é o mais recente:

M = 0 1 0 1 0 1 0 1
C = 0 0 0 0 1 1 0 0

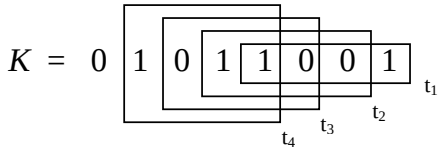
O criptoanalista soma as duas seqüências, módulo-2, e obtém um segmento da seqüência chave:

K = 0 1 0 1 1 0 0 1

Devido a estrutura linear do registro de deslocamento, sabemos que

$$g_4 x_4 \oplus g_3 x_3 \oplus g_2 x_2 \oplus g_1 x_1 = x_5$$

onde x_5 é o bit realimentado para a entrada, e g_i (= 0 ou 1) define se a i-ésima conexão de realimentação existe.



Então, podemos montar o sistema de equações, relacionado aos quatro instantes mostrados na figura acima:

$$g_4 \oplus g_1 = 1$$
$$g_4 \oplus g_3 = 0$$
$$g_3 \oplus g_2 = 1$$
$$g_4 \oplus g_2 \oplus g_1 = 0$$

cuja solução é $g_1 = 1$, $g_2 = 1$, $g_3 = 0$ e $g_4 = 0$, que corresponde exatamente as conexões do registro de deslocamento do nosso exemplo.

Para um sistema maior o criptoanalista pode montar o sistema de equações matricialmente:

$$\mathbf{x} = \mathbf{X}\mathbf{g}$$

e resolvê-lo por

$$\mathbf{g} = \mathbf{X}^{-1}\mathbf{x}$$

A inversão da matriz **X** requer, no máximo, n^3 operações e é facilmente realizada por um computador para qualquer valor razoável de n. Por exemplo, para $n = 100$, $n^3 = 10^6$, e supondo um tempo de 1 μs por operação, o computador levaria apenas 1 *segundo para quebrar o código*!

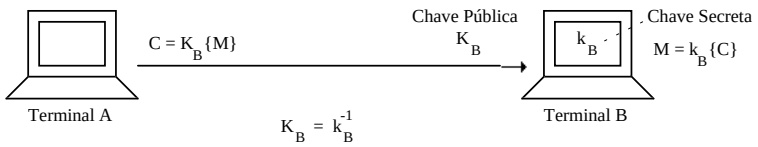
A vulnerabilidade do sistema é causada pela linearidade do sistema. A utilização de *realimentações não lineares* no registro de deslocamento torna a tarefa do criptoanalista muito mais *difícil* de ser realizada.

Sistemas Criptográficos Assimétricos

- Primeira proposta de Sistema Criptográfico Assimétrico ou de Chave Pública foi feita em 1976 por Diffie e Hellman.
- Sistemas Assimétricos $\rightarrow$ 2 chaves diferentes:

  Chave de Cifragem $\neq$ Chave de Decifragem
- A obtenção de uma chave a partir da outra é um problema difícil.

Cifragem e Decifragem



- Processo de Cifragem:

$$C = K_B\{M\}$$

- Processo de Decifragem:

$$M = k_B\{C\} = k_B\left\{K_B[M]\right\}$$

***Características necessárias para um sistema criptográfico assimétrico*:**

1. O cálculo do par de chaves deve ser simples.

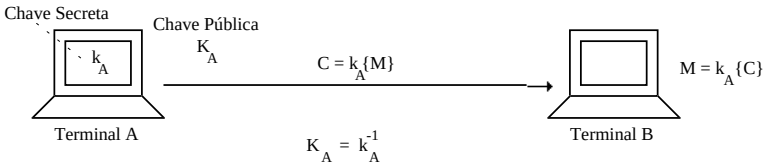
2. O transmissor A deve realizar a operação de cifragem facilmente, isto é, $C = K_B\{M\}$.

3. O receptor B deve realizar a operação de decifragem facilmente, isto é, $M = k_B\{C\}$.

4. A obtenção da chave secreta k_B a partir da chave pública K_B deve ser um problema de difícil solução.

5. Apesar do conhecimento do par (K_B, C), ainda assim a decodificação da mensagem M deve ser muito difícil.

Autenticação



- Processo de Autenticação:

$$C = k_A\{M\}$$

$$M = K_A\{C\} = K_A\{k_A[M]\}$$

→ Se a mensagem M decifrada pelo terminal B é *"coerente"*, significa que esta só pode ter sido gerada pelo terminal A, detentor da chave secreta.

Fundamentos Matemáticos

- **Teoria da Complexidade**

***Algoritmo*:** É uma seqüência finita de passos que resolvem algum problema.

***Problemas*:** Existem problemas solucionáveis e insolúveis.

Problema Solucionável $\leftrightarrow$ Algoritmo

***Complexidade*:** É medida pelo número de operações básicas necessárias no algoritmo para se obter a solução do problema.

***Problemas Solucionáveis*:** Realizáveis e Não Realizáveis.

Os *problemas não realizáveis* são aqueles que demandam recursos e/ou tempo de solução abusivamente grandes para se obter a solução.

Tipos de Complexidade dos Problemas Solucionáveis

- ***Polinomial* (P)**

A função de complexidade é um polinômio de n:

$$t(n) = a\ n^b$$

- ***Não Polinomial* (NP)**

A função de complexidade não é um polinômio de n:

$$t(n) = a\ b^n$$

$\rightarrow$ Complexidade NP >> Complexidade P para um dado n.

Problemas Matemáticos com Interesse em Criptografia

- ***Multiplicação***

Dados 2 números inteiros. Qual o valor de seu produto N?

→ Complexidade P.

- ***Fatoração***

Dado um número inteiro N. Descobrir os seus fatores.

→ Complexidade NP.

O melhor algoritmo apresenta complexidade obtida por

$$t(n) = \exp\left[\sqrt{\ln(n) \cdot \ln[\ln(n)]}\right]$$

Algoritmos básicos para a fatoração de inteiros:

- Curva Elíptica de Lenstra
- Classes de Grupos de Schnorr-Lenstra
- Sieve Linear de Schroeppel
- Sieve Quadrático de Pomerance
- Sieve da Lista de Resíduos de Coppersmith, Odlyzko e Schroeppel
- Fração Contínua de Morrison Brilhart

Introdução à Teoria dos Números

- ***Números Primos***

  - p > 1 ∈ **Z** é primo se e somente se 1 e p são os seus únicos divisores.
  - Os dez primeiros números primos: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29.
  - A quantidade de números primos é infinita e a freqüência de ocorrência cai à medida que n → ∞.

- ***MDC (Máximo Divisor Comum)***

O maior divisor em comum de dois inteiros *a* e *b*, mdc(*a*,*b*), vale *d*, onde *d* é o maior inteiro que satisfaz:

$$d \mid a \quad \text{e} \quad d \mid b\,.$$

Se mdc(*a*,*b*) = 1 → *a* e *b* são *primos entre si*.

- ***MMC (Mínimo Múltiplo Comum)***

O menor múltiplo em comum de dois inteiros *a* e *b*, mmc(*a*,*b*), vale *m*, onde *m* é o menor inteiro que satisfaz:

$$a \mid m \quad \text{e} \quad b \mid m\,.$$

- ***Teorema Fundamental da Teoria dos Números***

Qualquer inteiro pode ser fatorado na forma

$$\prod_i p_i^{e_i} \qquad e_i \in \mathbf{N}$$

Sejam $a = \prod_i p_i^{e_i}$ e $b = \prod_i p_i^{f_i}$, então

$$\text{mdc}(a,b) = \prod_i p_i^{\min(e_i, f_i)}$$
$$\text{mmc}(a,b) = \prod_i p_i^{\max(e_i, f_i)}$$
$$\text{mdc}(a,b) \cdot \text{mmc}(a,b) = a \cdot b$$

Ex.: Calcule o mdc e o mmc de 21 e 14.

Solução:

Fatorando os dois inteiros, temos

$$a = 21 = 2^0 \cdot 3^1 \cdot 7^1$$
$$b = 14 = 2^1 \cdot 3^0 \cdot 7^1$$

Logo,

$$\text{mdc}(21,14) = 2^0 \cdot 3^0 \cdot 7^1 = 7$$
$$\text{mmc}(21,14) = 2^1 \cdot 3^1 \cdot 7^1 = 42$$
$$\text{mdc}(21,14) \cdot \text{mmc}(21,14) = 294 = a \cdot b$$

- ***Algoritmo de Euclides***

Sejam a e $b \in \mathbf{N}$, $b \geq a$. Temos que

$$d \mid a \text{ e } d \mid b \quad \leftrightarrow \quad d \mid a \text{ e } d \mid (b - a)$$

então,

$$\text{mdc}(a,b) = \text{mdc}(a, b-a).$$

Como

$$b = q \cdot a + r \qquad 0 \leq r \leq a,$$

onde q é o *quociente* da divisão de b por a, e r é o *resto*. Logo,

$$\text{mdc}(a,b) = \text{mdc}(a,r)$$

Ex.: Calcule o mdc(220, 33) usando o algoritmo de Euclides.

Solução:

Aplicamos o algoritmo de Euclides sucessivamente, até encontrarmos o resto da divisão igual a zero:

$$\frac{220}{33} = 6 + \frac{22}{33}$$
$$\frac{33}{22} = 1 + \frac{11}{22}$$
$$\frac{22}{11} = 2 \qquad \Rightarrow \qquad \text{mdc}(220, 33) = 11$$

- ***Aritmética Modular***

Diz-se que a é congruente à b módulo n, isto é

$$a \equiv b \bmod n$$

se e somente se, para algum k inteiro

$$a = b + k\,n$$

ou seja,

$$n \mid (a - b)$$

Ex.:

$$29 = 2 + 9 \cdot 3 \qquad \Rightarrow \qquad 29 \equiv 2 \bmod 9$$

e, portanto

$$9 \mid (29\text{-}2).$$

- Se $a \equiv b \bmod n$, b é denominado *resíduo* de a módulo n.
- O conjunto de resíduos módulo n formam um anel comutativo para o qual as propriedades associativa, comutativa e distributiva são válidas:

$$(a \pm b) \bmod n \equiv (a \bmod n \pm b \bmod n) \bmod n$$
$$(a * b) \bmod n \equiv (a \bmod n * b \bmod n) \bmod n$$

Ex.: O número 1.234.567.890 é divisível por 9?

Solução:

Note que a soma dos dígitos do número 1.234.567.890 é igual a 45, cuja soma também é divisível por 9. A explicação da *Regra dos Noves-Fora* reside no fato que representando um número por seus dígitos decimais

$$(a_{m\text{-}1}\ a_{m\text{-}2}\ \ldots\ a_0) \bmod 9$$

então,

$$a = \left(a_{m-1}10^{m-1} + a_{m-2}10^{m-2} + \cdots + a_1 10 + a_0\right) \bmod 9.$$

Através das propriedades, obtemos que

$$a \equiv \left[\left(a_{m-1} \bmod 9\right) \cdot \left(10^{m-1} \bmod 9\right) + \cdots + \left(a_1 \bmod 9\right) \cdot \left(10 \bmod 9\right) + \left(a_0 \bmod 9\right)\right]$$

Notando que

$$10^n \bmod 9 \equiv 1,$$

então,

$$a \equiv \left(a_{m-1} + a_{m-2} + \cdots + a_1 + a_0\right) \bmod 9.$$

Desta forma, verifica-se que qualquer número cuja soma dos dígitos é um múltiplo de 9 é, também, divisível por 9.

Exercício:

Crie uma regra para saber se o número 137.295.132.482 é divisível por 11.

- ***Exponenciais***

As exponenciais são funções relativamente fáceis de serem calculadas.

Ex.:

$$3^{12} \bmod 7 \equiv \left(3^2 \bmod 7\right)^6 \bmod 7 \equiv 2^6 \bmod 7 \equiv 1 \bmod 7$$

- ***Logaritmos***

Ao contrário das exponenciais, os logaritmos são problemas difíceis de serem resolvidos.

Ex.: Encontre x para as expressões modulares.

a) $3^x \equiv 4 \bmod 13$.

Como

$$3^0 = 1,\ 3^1 = 3,\ 3^2=9,\ 3^3=27;$$

não existe solução.

b) $2^x \equiv 3 \bmod 13$

Como

$$2^0 = 1,\ 2^1 = 2,\ 2^2 = 4,\ 2^3 = 8,\ 2^4 = 16 \equiv 4;$$

portanto $x = 4$.

- ***Cálculo de Inversas***

Ao contrário da aritmética comum dos números inteiros, a aritmética modular *às vezes* tem inversa, isto é

$$(a\,x) \bmod n \equiv 1, \qquad a \text{ e } x \in \{0,1,\ldots,n-1\}$$

Ex.: $(3x) \bmod 10 \equiv 1$

Por tentativas podemos obter que $x = 7$.

- Se a e n são primos entre si, isto é, mdc(a, n) = 1, então

$$(a \cdot i) \bmod n \neq (a \cdot j) \bmod n, \qquad 0 \leq i \neq j < n$$

Assim, neste caso, (a.i) mod n é uma permutação de {0,1,…,n-1}.

Ex.: É fácil constatar que (3i) mod 7 = {0, 3, 6, 2, 5, 1, 4} para i = 0,…,6.

Quando o mdc(a, n) ≠ 1, isto não é verdadeiro:

(2i) mod 6 = {0, 2, 4, 0, 2, 4} para i = 0,…,5.

- Se mdc(a,n) = 1, existe um elemento a^{-1}, tal que

$$a\,a^{-1} \equiv 1 \bmod n$$

***Prova*:** O conjunto de resíduos de (ai) mod n é uma permutação de {0, 1,…, n-1}, de modo que sempre existe para um dado a, um elemento a^{-1} tal que aa^{-1} mod n.

- A *função de Euler*, $\phi(n)$, é definida como o número de elementos do conjunto completo de resíduos que é primo de n. Ao conjunto destes elementos denominamos de *conjunto reduzido de resíduos*.

Ex.: Determine o conjunto reduzido de resíduos e o valor da função de Euler para n = 10.

Solução:

Para n = 10, dos elementos {0, 1,…,9}, somente os elementos {1, 3, 7, 9} são primos de 10, constituindo, assim, o conjunto reduzido de resíduos e, neste caso, $\phi(10) = 4$.

- Se n é primo, então $\phi(n) = n - 1$.

Ex.: Para $n = 5$, dos elementos $\{0,1,\ldots,4\}$, todos são primos de 5, exceto o 0. Logo, $\phi(5) = 4$.

- ***Teorema Generalizado de Euler*:**

  Se $\text{mdc}(a, n) = 1$, então

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

- ***Teorema de Fermat*:**

  Se p é primo e $\text{mdc}(a, p) = 1$, então

$$a^{p-1} \bmod p \equiv 1$$

Algoritmos para se Obter Inversas

1. Se $\text{mdc}(a, n) \neq 1$, procure em todo o conjunto dos resíduos qual o elemento é o inverso, se é que ele existe.

2. Se $\phi(n)$ é conhecido e $\text{mdc}(a, n) = 1$, então através do teorema de Euler:

$$a^{\phi(n)} \bmod n \equiv 1 \quad \Rightarrow \quad a\, a^{\phi(n)-1} \bmod n \equiv 1$$

   e, obtemos o elemento inverso através da congruência

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

3. Se $\phi(n)$ é desconhecido podemos usar uma extensão do algoritmo de Euclides.

Ex.: Calcule x, tal que $5x \equiv 1 \bmod 23$

Solução:

a) Através do teorema de Euler:

$$x = 5^{-1} \equiv 5^{\phi(23)-1} \bmod 23$$
$$\phi(23) = 22 \quad (23 \text{ e' primo})$$
$$x \equiv 5^{21} \bmod 23 \equiv \left(5^7\right)^3 \bmod 23 \equiv \left(5^7 \bmod 23\right)^3 \bmod 23$$
$$x \equiv 17^3 \bmod 23 \equiv 14 \bmod 23$$

Logo, $x = 14$.

b) Através do algoritmo de Euclides:

$$3 = 23 - 4 \cdot 5$$
$$2 = 5 - 1 \cdot 3 = 5 - 1(23 - 4 \cdot 5) = 5 \cdot 5 - 1 \cdot 23$$
$$1 = 3 - 1 \cdot 2 = (23 - 4 \cdot 5) - (5 \cdot 5 - 1 \cdot 23) = 23 \cdot 2 + 5 \cdot (-9)$$

Logo,

$$5 \cdot (-9) \equiv 1 \bmod 23 \qquad \Rightarrow \qquad x = -9$$

Mas

$$-9 \equiv (-9 + 23) \bmod 23 = 14 \bmod 23$$

E, assim

$$x = 14.$$

Exercício:

Calcule x, tal que $3x \equiv 1 \bmod 17$, usando o teorema de Euler. Refaça o problema através do algoritmo de Euclides.

Algoritmo para Resolver Equações

- Para se obter x, tal que $ax \equiv b \bmod n$, primeiro resolva a inversa

$$ay \equiv 1 \bmod n,$$

e então, encontre

$$x \equiv (yb) \bmod n.$$

Ex.: $5x \equiv 9 \bmod 23$

Resolvemos primeiro

$$5y \equiv 1 \bmod 23 \implies y = 14 \quad \text{(exemplo anterior)}$$

Em seguida computamos

$$x \equiv (14 \cdot 9) \bmod 23 = 126 \bmod 23 \equiv 11 \bmod 23$$

E, assim, obtemos

$$x = 11.$$

- Se $\text{mdc}(a,n) = g$, e $g|b$, então a equação $ax \equiv b \bmod n$ tem g soluções:

$$x \equiv \left[\left(\frac{bx_o}{g}\right) \bmod \left(\frac{n}{g}\right) + \left(t\frac{n}{g}\right)\right] \bmod n \quad \text{para } t = 0,\ldots, g\text{-}1$$

onde x_o é solução de

$$\left(\frac{a}{g}\right)x \equiv 1 \bmod \left(\frac{n}{g}\right)$$

Caso contrário não há solução.

Ex.: Resolva a equação $9x \equiv 6 \bmod 12$.

Observamos que $g = \text{mdc}(9, 12) = 3$ e que $3|6$. Portanto existem 3 soluções. Precisamos resolver primeiro

$$3x_o \equiv 1 \bmod 4$$

que resulta em $x_o = 3$.

Assim, as 3 soluções são:

$$x = (6 \bmod 4) + 4t, \quad t = 0, 1, 2$$

isto é, $x = 2$, 6 e 10.

- Sejam $p_1, p_2, \ldots, p_r$ primos dois a dois. Defina $n = p_1 p_2 \ldots p_r$.

  Então,

$$f(x) \bmod n \equiv 0$$

  se e somente se

$$f(x) \bmod p_i \equiv 0 \qquad i = 1, \cdots, r.$$

  Portanto, neste caso, para resolver

$$ax \equiv b \bmod n$$

  é mais fácil resolver o sistema de congruências:

$$ax \equiv b \bmod p_i \qquad i = 1, \cdots, r.$$

- ***Teorema Chinês do Resto*:**

  Sejam $p_1, p_2, \ldots, p_r$ primos dois a dois. Defina $n = p_1 p_2 \ldots p_r$.

  Então o sistema de congruências

$$x \equiv x_i \bmod p_i \qquad i = 1, \cdots, r$$

  tem uma em comum no intervalo {0, 1, … , n - 1}, e existe um y_i, tal que

$$\left(\frac{n}{p_i}\right) y_i \equiv 1 \bmod p_i$$

  e

$$\left(\frac{n}{p_i}\right) y_i \equiv 0 \bmod p_j \qquad j \neq i \quad \text{e} \quad p_j \mid \frac{n}{p_i}$$

  De modo que

$$x = \left(\sum_{i=1}^{r} \frac{n}{p_i} y_i x_i\right) \bmod n$$

  satisfaz as congruências, pois

$$x = \frac{n}{p_i} y_i x_i \equiv x_i \bmod p_i$$

Ex.: Obtenha $7^{-1} \bmod 65$, ou seja, resolva $7x \equiv 1 \bmod 65$.

Solução: Observamos que 65 possui dois fatores ($65 = 13 \cdot 5$) e, assim, temos que resolver:

$$7x \equiv 1 \bmod 5 \quad \Rightarrow \quad x_1 = 3$$
$$7x \equiv 1 \bmod 13 \quad \Rightarrow \quad x_2 = 2$$

Usando o teorema chinês do resto para descobrir qual x satisfaz as duas equações abaixo:

$$x = x_1 \equiv 3 \bmod 5$$
$$x = x_2 \equiv 2 \bmod 13$$

E obtendo em seguida y_1 e y_2:

$$\frac{65}{5} y_1 = 13 y_1 \equiv 1 \bmod 5 \quad \Rightarrow \quad y_1 = 2$$
$$\frac{65}{13} y_2 = 5 y_2 \equiv 1 \bmod 13 \quad \Rightarrow \quad y_2 = 40$$

Finalmente,

$$x = \frac{65}{5} x_1 y_1 + \frac{65}{13} x_2 y_2 = 13 \cdot 3 \cdot 2 + 5 \cdot 2 \cdot 8 = 28 \bmod 65$$

Portanto,

$$7^{-1} \equiv 28 \bmod 65$$

- ***O Problema da Mochila* (*Knapsack*)**

O problema da mochila é um problema clássico:

"*Uma mochila é preenchida com vários objetos de pesos diferentes e conhecidos. Dado o peso líquido total da mochila, determinar quais os objetos estão dentro da mochila sem abrí-la*."

Exercício:

Dados 6 objetos com pesos {1, 2, 5, 10, 22, 45} [Kg] e sabendo que o peso total da mochila é de 37 Kg. Quais os objetos que estão dentro da mochila?

O problema anterior é relativamente fácil de resolver por dois motivos:

1) O número de ítens é pequeno.
2) A soma dos pesos dos objetos mais leves é sempre menor que o próximo objeto mais pesado (vetor superaumentado).

A solução do problema da mochila, quando a condição 2 acima é satisfeita, pode ser sistematizada. Definimos um vetor de pesos

$$\mathbf{a} = a_1, a_2, \ldots, a_n$$

e, um vetor de dados binário

$$\mathbf{x} = x_1, x_2, \ldots, x_n$$

O peso da mochila é a soma de um subconjunto de pesos definido pela presença ou ausência dos diversos ítens, isto é

$$S = \sum_{i=1}^{n} a_i x_i = \mathbf{ax} \quad \text{onde} \quad x_i = 0,1$$

A solução do problema é obtida, começando com $x_n = 1$ se $S \geq a_n$, e continuando usando a seguinte regra:

$$x_i = \begin{cases} 1 & \text{se} \quad S - \sum_{j=i+1}^{n} x_j a_j \geq a_i \\ 0 & \text{caso contrario} \end{cases}$$

onde $i = n - 1, n - 2, \ldots, 1$.

Ex.: Dados **a** = 171, 197, 459, 1191, 2410, 4517 e S = 3798, encontre **x**.

Solução: Aplicando a regra anterior, obtemos

$S = 3798 < a_6 = 4517 \quad \Rightarrow \quad x_6 = 0$

$S - a_6 x_6 = 3798 - 4517 \cdot 0 = 3798 \quad > \quad a_5 = 2410 \quad \Rightarrow \quad x_5 = 1$

$3798 - (2410 + 0) = 1388 \quad > \quad a_4 = 1191 \quad \Rightarrow \quad x_4 = 1$

$3798 - (1191 + 2410 + 0) = 197 \quad < \quad a_3 = 459 \quad \Rightarrow \quad x_3 = 0$

$3798 - (0 + 1191 + 2410 + 0) = 197 \quad = \quad a_2 = 197 \quad \Rightarrow \quad x_2 = 1$

$3798 - (197 + 0 + 1191 + 2410 + 0) = 0 \quad < \quad a_1 = 171 \quad \Rightarrow \quad x_1 = 0$

Logo, o vetor de dados obtido é

$$\mathbf{x} = [010110]$$

Sistemas de Cifragem de Chave Pública

- **Desvantagem dos Sistemas Simétricos:** *Número grande de chaves a serem mantidas e administradas.*

Os *sistemas simétricos*, como o DES, são *satisfatórios* quando temos *comunicação ponto a ponto*.

Uma rede com *N usuários* requer $N(N-1)/2$ *pares de chaves diferentes*.

Cada *usuário deve armazenar N* - 1 *chaves* de cifragem / decifragem correspondentes a todos os demais usuários.

Problema de segurança no sigilo das chaves: - a partir de certo usuário, seria possível obter as chaves de acesso aos demais usuários.

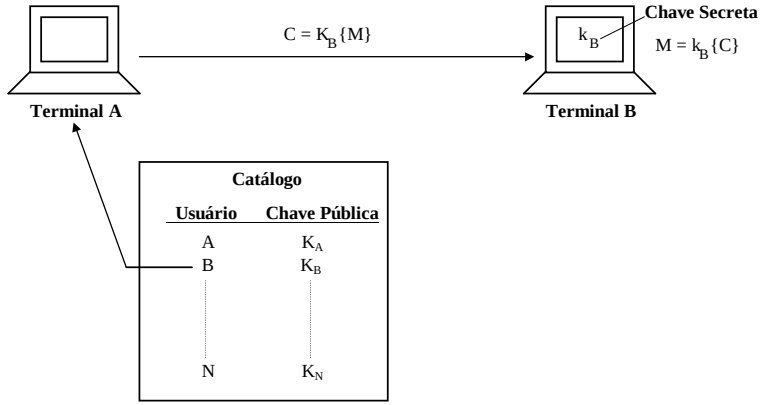
- V**antagem dos Sistemas Assimétricos:** *Gerência das chaves simplificada*.

Os sistemas criptográficos de chave pública se caracterizam por duas chaves diferentes: uma usada para a cifragem e outra para a decifragem.

A chave de cifragem é tornada pública através de um arquivo de acesso irrestrito na rede de comunicação. Cada usuário mantém apenas a sua chave de decifragem secreta.

A obtenção da chave de decifragem a partir da de cifragem é difícil.

O número necessário de chaves a ser armazenado no arquivo público (catálogo) é igual ao número N de usuários da rede, muito menor do que no sistema simétrico.



Esquema Criptográfico RSA

- Criado por *Rivest*, *Shamir* e *Adleman*.
- Utiliza dois problemas de difícil resolução: - fatoração e cálculo de logaritmos em corpos finitos.
- As mensagens, os criptogramas e as chaves pertencem ao conjunto de inteiros módulo N, onde $N = pq$ com p e q primos.
- **Cifragem:**

$$C = K_X\{M\} \equiv M^{K_X} \bmod N$$

- **Decifragem:**

$$M = k_X\{C\} \equiv C^{k_X} \bmod N$$

onde

K_X é a chave pública do usuário X, e
k_X é a chave secreta do usuário X.

- **Geração do Par de Chaves RSA**

Substituindo a expressão da mensagem cifrada

$$C = M^K \bmod N$$

na expressão de decifragem

$$M \equiv C^k \bmod N$$

obtemos

$$\left(M^K\right)^k \equiv M \bmod N \quad \Rightarrow \quad M^{kK-1} \equiv 1 \bmod N$$

Se escolhermos o par de chaves satisfazendo o teorema de Fermat, então, as chaves podem ser obtidas através de

$$kK \equiv 1 \bmod \phi(N)$$

para N formado pelo produto de dois números primos p e q, tal que

$$\phi(N) = (p-1)(q-1).$$

Também, pode-se mostrar que o par de chaves pode ser obtido, mais facilmente, computando-se

$$kK \equiv 1 \bmod \gamma(N)$$

onde

$$\gamma(N) = \text{mmc}(p-1, q-1)$$

pois, esta equação possui um módulo menor que a anterior.

Ex.: Obtenha um par de chaves para o esquema de cifragem RSA, com $p = 11$ e $q = 7$.

Solução:

Observando que p e q são primos e $N = pq = 77$, temos que

$$\gamma(77) = \text{mmc}(11-1, 7-1) = \text{mmc}(10, 6) = 30$$

Escolhendo, aleatoriamente, a chave pública como $K = 17$, obtemos a chave secreta resolvendo a equação

$$17k \equiv 1 \bmod 30$$

cuja solução pode ser obtida, por exemplo, pelo método de Euler:

$$k = 17^{\phi(30)-1} \bmod 30$$

como 30 consiste no produto dos fatores primos 2, 3 e 5, temos que

$$\phi(30) = (2-1)(3-1)(5-1) = 8$$

Logo,

$$k = 17^7 \bmod 30 \qquad \Rightarrow \qquad k = 23$$

Para o esquema RSA, o par $N = 77$, $K = 17$ será tornado público, enquanto que $k = 23$ será mantido em sigilo.

Ex.: Considere que o terminal A deseja enviar uma mensagem para o terminal B, cujo primeiro caractere é $M_1 = 33$, e a chave pública de B é aquela escolhida no exemplo anterior. Qual será o primeiro criptograma enviado por A? Mostre que o terminal B recupera a mensagem corretamente.

Solução:

O terminal para cifrar o primeiro caractere da mensagem realizará a operação:

$$C_1 = M_1^K = 33^{17} \bmod 77$$

Esta exponencial é relativamente fácil de se calcular:

$$\begin{aligned} 33^{17} \bmod 77 &\equiv \left[(33^4)^4 \cdot 33\right] \bmod 77 \equiv \left[\left(44^4 \bmod 77\right) \cdot 33\right] \bmod 77 \\ &\equiv (44 \cdot 33) \bmod 77 \equiv 66 \bmod 77 \end{aligned}$$

Logo, o criptograma enviado é $C_1 = 66$.

De forma similar, o terminal B resolve a exponencial com auxílio da chave secreta

$$M_1 = 66^{23} \bmod 77$$

recuperando $M_1 = 33$.

Exercício:

Considere um esquema RSA com $p = 5$ e $q = 7$.

a) Determine a chave pública K considerando que a chave secreta $k = 11$ foi escolhida previamente.

b) Calcule os criptogramas correspondentes à mensagem $M = \{17, 2, 6\}$, e mostre como o destinatário é capaz de recuperá-la.

Segurança do Esquema RSA

- ***Ataques Convencionais*:**

  1. Fatorar N. Após ter fatorado N, calcular $\gamma(N)$ e a chave secreta k.

  2. Conhecendo-se o par (K, C), determinar a mensagem M através de logaritmo.

  $\rightarrow$ Os dois problemas têm complexidade NP.

- ***Ataque Iterativo* (*Simmon e Norris*):**

  Conhecendo-se a tripla (N, K, C) é possível gerar a seqüência:

  $$C_i = C_{i-1}^K \bmod N \qquad i = 1, \ldots$$

  onde $C_0 = C$.

  Se existir um elemento C_j, $j \neq 0$ tal que $C_j = C$, então $M = C_{j-1}$.

  Se $p - 1$ e $q - 1$ contiverem grandes primos como fatores e N for um número grande, a probabilidade de quebra do código é pequena.

Ex.: Faça um ataque iterativo ao esquema RSA com $N = 77$, $K = 17$ e $C = 66$.

$$C_1 = C^K = 66^{17} \equiv 33 \bmod 77$$
$$C_2 = C_1^K = 33^{17} \equiv 66 \bmod 77$$

Como

$$C_2 = C$$
$$\Rightarrow \quad M = C_1 = 33$$

- ***Criptogramas Iguais às Mensagens***

No sistema RSA, algumas vezes o criptograma é igual à mensagem.

Ex: Compute o criptograma correspondente a $M = 2$, para $K = 31$ e $N = 77$.

$$C = 2^{31} \bmod 77 = 2$$

De fato, para $K = 31$, qualquer criptograma resulta igual à mensagem, caracterizando um caso extremo. Note também que os caracteres 0 e 1 sempre geram criptogramas idênticos às mensagens para qualquer N.

Para $K = 17$ e $N = 77$, existem nove criptogramas iguais às próprias mensagens. Isto é explicado porque, neste caso:

$$C = M^K \equiv M \bmod N$$

e, como $N = pq$, podemos reescrever:

$$M^K = M \bmod p$$
$$M^K = M \bmod q$$

Para K ímpar, ambas congruências apresentam pelo menos 3 soluções: {0, 1, -1}. Assim, existem 9 combinações que satisfazem simultaneamente às duas congruências anteriores.

Ex.: Descubra os 9 criptogramas idênticos às mensagens, que ocorrem para qualquer K ímpar, quando $N = 7 \cdot 11 = 77$.

Solução:

A primeira congruência, $M^K \equiv M \bmod 7$, tem como soluções $\{0, 1, 6\}$. A segunda, $M^K \equiv M \bmod 11$, fornece $\{0, 1, 10\}$. O conjunto das 9 mensagens que não são alteradas é obtido através do teorema chinês do resto:

$$\begin{aligned}
0 &= [0 \bmod 7, 0 \bmod 11] \\
56 &= [0 \bmod 7, 1 \bmod 11] \\
21 &= [0 \bmod 7, 10 \bmod 11] \\
22 &= [1 \bmod 7, 0 \bmod 11] \\
1 &= [1 \bmod 7, 1 \bmod 11] \\
43 &= [1 \bmod 7, 10 \bmod 11] \\
55 &= [6 \bmod 7, 0 \bmod 11] \\
34 &= [6 \bmod 7, 1 \bmod 11] \\
76 &= [6 \bmod 7, 10 \bmod 11]
\end{aligned}$$

- ***Número de Criptogramas Iguais às Mensagens*:**

Para o esquema RSA, para $N = pq$, e uma dada chave pública K, existirão σ criptogramas iguais às mensagens, obtido por:

$$\sigma = [1 + \text{mdc}(K-1, p-1)][1 + \text{mdc}(K-1, q-1)]$$

Como decorrência da expressão acima, deduzimos que se a chave pública K for escolhida da forma

$$K = \gamma(N) + 1 = \text{mmc}(p-1, q-1) + 1$$

então todos os criptogramas gerados serão iguais às mensagens, pois, neste caso

$$(p-1) | (K-1) \quad \text{e} \quad (q-1) | (K-1)$$

e, portanto, teremos

$$\sigma = (1 + q - 1)(1 + p - 1) = N.$$

Algumas conclusões podem ser tiradas para que não existam muitos criptogramas idênticos às suas mensagens:

- Se $K = 2$, ou se $K - 1$ for primo de $p - 1$ e de $q - 1$, isto é, $\text{mdc}(K - 1, p - 1) = \text{mdc}(K - 1, q - 1) = 1$, então existirão 4 mensagens iguais aos seus criptogramas.

- Se $K - 1$ tiver apenas o 2 em comum com $p - 1$ e $q - 1$, ou seja, $\text{mdc}(K - 1, p - 1) = \text{mdc}(K - 1, q - 1) = 2$, então teremos 9 mensagens idênticas aos criptogramas.

Conclusões sobre o Sistema RSA

- Se $p - 1$ e $q - 1$ tiverem grandes fatores primos em sua composição, evita-se o problema de existirem muitas mensagens iguais aos criptogramas.

- Se usarmos para p e q números com comprimento superior a 100 casas decimais, levando-se em conta o *poder computacional* atual, o sistema criptográfico RSA será bastante seguro.

  A complexidade de quebra do sistema RSA, equivale à dos problemas da fatoração e do logaritmo em corpos finitos:

$$t(N) = \exp\sqrt{\ln N \ln(\ln N)}$$

  Uma estimativa para o tempo necessário para a quebra do RSA em função do número de casas decimais de N e supondo que cada operação demore 0,1μs, é fornecida abaixo:

$\log_{10} N$	$t(N)$	tempo
50	$1{,}4 \times 10^{10}$	23,7 min.
75	$9{,}0 \times 10^{12}$	10,4 dias
100	$2{,}3 \times 10^{15}$	7,4 anos
129	$7{,}3 \times 10^{17}$	2.299 anos
200	$1{,}2 \times 10^{23}$	$3{,}8 \times 10^{8}$ anos
300	$1{,}5 \times 10^{29}$	$4{,}9 \times 10^{14}$ anos
500	$1{,}3 \times 10^{39}$	$4{,}1 \times 10^{24}$ anos

- *Não se deve desprezar a velocidade com que a tecnologia dos computadores avança*!

  Em 1977 *Rivest* anunciou na revista Scientific American a criação de um número RSA-129, que na época levaria 40 *quatrilhões de anos* para ser fatorado. Em 1994 *Lenstra* anunciou a quebra do RSA-129. Para conseguir esta proeza, ele recrutou 600 voluntários pela Internet para ajudá-lo na empreitada. Abaixo apresentamos o número RSA-129 e seus dois fatores:

114381625757888867669235779976146612010218296721242362562561842935706935245733897830597123563958705058989075147599290026879543541

=

349052951084765094914784961990389813341776463849338784399082057 7

×

327691329932667095499619881908344614131776429679929425397982885 33

- Apesar da operação de exponenciação constituir-se em um problema de complexidade polinomial, ainda assim o tempo gasto na execução de exponenciais em torno de 200 casas decimais limita a taxa de bits, atualmente, na ordem de *dezenas de kilobits por segundo*. Em muitos casos este patamar é insuficiente para atender as demandas emergentes dos novos serviços de comunicação.

Sistema Misto RSA / DES

O padrão DES é capaz de cifrar uma seqüência de bits a uma taxa da ordem de dezenas ou mesmo centenas de megabits por segundo, pois realiza operações elementares com bits. No entanto, a gerência das chaves é complexa e insegura, devido ao sistema ser simétrico, requerendo que se defina uma chave de cifragem/decifragem por par de usuários.

Já no sistema RSA as operações de cifragem e decifragem são relativamente lentas, porém a gerência das chaves é simplificada.

Uma solução que tem sido usada é um sistema criptográfico misto:

- O terminal A querendo estabelecer comunicação com o terminal B, requer o envio de uma chave DES.
- Uma chave DES é sorteada por B e enviada para o terminal A, criptografada com auxílio da chave pública RSA de A.
- O terminal A decifra a chave DES através de sua chave secreta RSA, sendo A o único elemento na rede capaz de fazê-lo.
- O terminal A inicia a transferência de dados, criptografando-os com a chave DES recebida. Por sua vez, o terminal B realiza a decifragem através da mesma chave.

Esquema Criptográfico DH

- Sistema criptográfico criado por *Diffie* e *Hellman* em 1976, sendo o primeiro esquema de chave pública proposto.

- Utiliza somente o problema do logaritmo em corpos finitos:

  Utiliza um elemento primitivo g pertencente a um corpo de *Galois* GF(N), onde N é um número primo. Por elemento primitivo se entende o elemento que é capaz de gerar todos os N elementos de GF(N).

  O elemento primitivo g e o primo N são tornados públicos.

- Escolhe-se duas chaves secretas k_A e k_B, e então calculam-se suas *chaves parciais* que são tornadas públicas:

$$y_A = g^{k_A} \bmod N$$
$$y_B = g^{k_B} \bmod N$$

- Os usuários A e B calculam uma chave secreta K em comum:

$$K = (y_B)^{k_A} = \left(g^{k_B}\right)^{k_A} \bmod N$$
$$K = (y_A)^{k_B} = \left(g^{k_A}\right)^{k_B} \bmod N$$

- **Cifragem:** O processo de cifragem é idêntico ao do RSA.

$$C = M^K \bmod N$$

- **Decifragem:** A chave de decifragem k é distinta da chave secreta k_X do usuário receptor X. Ela é calculada de forma a ser complementar à chave de cifragem K, através da congruência:

$$kK \equiv 1 \bmod (N-1)$$

Para decifrar os criptogramas o receptor calcula, usando a chave de decifragem:

$$M = C^k \bmod N$$

- Observa-se que no sistema DH é fácil se obter a chave de cifragem K a partir da decifragem k, e vice-versa. Nem por isso o sistema é simétrico, do ponto de vista da chave secreta k_X, nem a segurança do sistema fica comprometida.

- Uma vantagem deste sistema é que além da cifragem das mensagens, o esquema DH garante a *autenticidade* da origem das mesmas.

Ex.: Considere o sistema DH com o número primo $N = 47$ e um elemento primitivo $g = 23$. Suponha que os usuários A e B escolham suas chaves secretas $k_A = 12$ e $k_B = 33$. Qual será o criptograma enviado de A para B correspondente à mensagem $M = 16$. Como B consegue decifrar o criptograma?

Solução: Os usuários A e B obtêm as suas chaves parciais, que são tornadas públicas:

$$y_A = g^{k_A} = 23^{12} \bmod 47 = 27$$
$$y_B = g^{k_B} = 23^{33} \bmod 47 = 33$$

Em seguida, A e B calculam a chave secreta de cifragem:

$$K = (y_B)^{k_A} = (y_A)^{k_B} \equiv 33^{12} \bmod 47 \equiv 27^{33} \bmod 47 = 25$$

e, também, a chave secreta de decifragem que é a inversa de K:

$$25k \equiv 1 \bmod 46 \quad \Rightarrow \quad k = 35$$

Para criptografar a mensagem $M = 16$, o terminal A utiliza a chave de cifragem:

$$C = M^K \equiv 16^{25} \bmod 47 = 21$$

O terminal B restaura a mensagem recebida através da chave de decifragem:

$$M = C^k \equiv 21^{35} \bmod 47 = 16$$

Exercício:

Considere o sistema DH do exemplo anterior, isto é, $N = 47$ e $g = 23$. O objetivo é enviar e receber uma mensagem secreta de um companheiro. Escolha sua chave secreta $k_X < N$ e calcule a chave parcial y_X. Em seguida divulgue a sua chave parcial ao seu colega e vice-versa. Agora, cada um em separado deve calcular as chaves de cifragem K e decifragem k. Escolha uma mensagem $M < N$ e gere o criptograma com auxílio da chave de cifragem e informe ao seu companheiro. Também, receba o criptograma do colega e decifre-o. Ao final, compare os resultados.

Segurança do Esquema Criptográfico DH

Para descobrir a chave secreta do usuário A, k_A, um criptoanalista dispõe de y_A, g e N. Ele deve obter a chave resolvendo

$$y_A = g^{k_A} \bmod N$$

que é um problema de logaritmo em corpos finitos, que tem a mesma complexidade do problema da fatoração.

Portanto, supondo que cada operação gaste 0,1 μs de processamento, teremos a mesma estimativa para o tempo de quebra do sistema DH em função do número de dígitos de N, como mostrado anteriormente na tabela para o esquema RSA.

Para o mesmo número de dígitos adotado, o sistema DH tem a desvantagem de ser um pouco mais complexo que o RSA, pois é preciso calcular um par de chaves cada vez que se comunica com um usuário diferente.

No entanto, o sistema DH apresenta a vantagem de embutir a autenticação da origem.

Esquema Criptográfico MH

Este sistema criptográfico foi criado por *Merkle* e *Hellman* em 1978, utiliza o problema da mochila com vetores não super-aumentados e, por isso, é difícil de quebrar.

A mensagem de n bits a ser cifrada corresponde ao vetor de dados binário do problema da mochila:

$$M = (m_1, m_2, \ldots, m_n)$$

Cada usuário cria um vetor $W = (w_1, w_2, \ldots, w_n)$ super-aumentado, que cria as condições para que tenhamos um problema da mochila de solução trivial, isto é

$$w_i > \sum_{j}^{i-1} w_j$$

Além disso, escolhe um número primo q maior que o peso máximo da mochila, e também um multiplicador r entre 1 e q. Assim

$$q > \sum_{i=1}^{n} w_i \qquad \text{e} \qquad 1 < r < q$$

A chave pública $K = (k_1, k_2, \ldots k_n)$ será um vetor de comprimento n, cujos elementos são calculados através de

$$k_i = (w_i r) \bmod q$$

O vetor W, o primo q e o multiplicador r são mantidos secretos, enquanto que o vetor K é tornado público.

- **Cifragem:** O processo de cifragem é bastante simples.

$$C = \sum_{i=1}^{n} m_i k_i$$

Note que, como K não é um vetor superaumentado, decifrar o criptograma sem conhecimento adicional, constitui-se num problema da mochila não trivial, portanto, difícil.

- **Decifragem:** O processo de decifragem é realizado pelo receptor a partir da obtenção do peso da mochila:

$$C' = C\, r^{-1} \bmod q = \sum_{i=1}^{n} m_i\, k_i\, r^{-1} \bmod q = \sum_{i=1}^{n} m_i\, w_i \bmod q$$

Como o vetor W foi escolhido superaumentado, do ponto de vista do receptor, o lado direito da expressão é um problema da mochila trivial. O receptor resolvendo-o, então, decifra a mensagem enviada.

Ex.: Considere um sistema MH com $n = 6$. Suponha que o terminal B selecionou o vetor W = (171, 197, 459, 1191, 2410, 4517), o primo $q = 9109$ e o multiplicador r = 2251. Determine o vetor chave público de cifragem K_B e o criptograma enviado de A para B correspondente à seqüência de bits de mensagem M = (0, 1, 0, 1, 1, 0). Como o terminal B decifra o criptograma recebido?

Solução:

A chave pública do terminal B é calculada, como a seguir:

$$k_1 = (171 \cdot 2251) \bmod 9109 = 2343$$
$$k_2 = (197 \cdot 2251) \bmod 9109 = 6215$$
$$k_3 = (459 \cdot 2251) \bmod 9109 = 3892$$
$$k_4 = (1191 \cdot 2251) \bmod 9109 = 2895$$
$$k_5 = (2410 \cdot 2251) \bmod 9109 = 5055$$
$$k_6 = (4517 \cdot 2251) \bmod 9109 = 2123$$

Logo,

K_B = (2343, 6215, 3892, 2895, 5055, 2123)

O criptograma enviado por A pode ser, então, facilmente computado:

$$C = \sum_{i=1}^{6} m_i\, k_i = 6215 + 2895 + 5055 = 14165$$

O terminal B ao ter escolhido r = 2251, também terá calculado, previamente, r^{-1}, isto é

$$2251 r^{-1} \equiv 1 \bmod 9109$$

Através do algoritmo de Euclides, obtemos:

$$105 = 9109 - 4 \cdot 2251$$
$$46 = 2251 - 21 \cdot 105 = 85 \cdot 2251 + 21 \cdot 9109$$
$$13 = 105 - 2 \cdot 46 = -174 \cdot 2251 + 43 \cdot 9109$$
$$7 = 46 - 3 \cdot 13 = 607 \cdot 2251 + 150 \cdot 9109$$
$$6 = 13 - 1 \cdot 7 = -781 \cdot 2251 + 193 \cdot 9109$$
$$1 = 7 - 1 \cdot 6 = 1388 \cdot 2251 - 343 \cdot 9109$$

Logo, r^{-1} = 1388.

O peso da mochila é, portanto

$$C' = (14165 \cdot 1388) \bmod 9109 = 3798$$

O terminal B usa, então o vetor secreto W = (171, 197, 459, 1191, 2410, 4517), e recupera a mensagem solucionando o problema da mochila, agora trivial:

M = (0, 1, 0, 1, 1, 0).

Exercício:

Assuma que as partes A e B concordam em usar vetores de comprimento 5. O terminal B escolheu um vetor W dado por:

$$W = (w_1, \cdots, w_5) = (2, 3, 6, 12, 25) \quad \Rightarrow \quad \sum w_i = 48$$

Ainda, B fez a escolha de um número primo q = 53 > 48, e um multiplicador r = 46. Determine o criptograma enviado por A correspondente à mensagem M = (1, 1, 1, 0, 1). Decifre o criptograma recebido por B.

Segurança do Esquema Criptográfico MH

Schroeppel e *Shamir* criaram em 1982 um algoritmo que resolve o problema da mochila com um número de computações da ordem de

$$t(n) = 2^{n/2}$$

A seguir estão listados o número de computações e o tempo gasto para quebrar o esquema de *Merkle* e *Hellman* em função do número de bits do bloco de mensagem, *n*:

n	***t*(*n*)**	**tempo**
50	$3{,}4 \times 10^{7}$	3,3 s
75	$1{,}9 \times 10^{11}$	13,2 min
100	$1{,}1 \times 10^{15}$	3,17 anos
150	$3{,}8 \times 10^{22}$	$1{,}2 \times 10^{8}$ anos
200	$1{,}3 \times 10^{30}$	$4{,}0 \times 10^{15}$ anos

Podemos observar que o esquema MH atinge patamares de confiabilidade maiores que os sistemas RSA e DH quando usamos números com $n \geq 200$.

No entanto, a desvantagem reside no fato de que, se n = 200 o vetor público K será composto, de 200 números com 200 bits, isto é, em torno de 40 Kbits por usuário. Além disso, os *criptogramas* são *maiores que as mensagens*.

Esquemas de Autenticação

- Necessidade de autenticação quando precisamos ter um conhecimento preciso de quem transmitiu certa informação (p. ex., transações comerciais e bancárias).

- Em geral, quando se faz a cifragem de uma mensagem, a autenticação é perdida e vice-versa.

- O que caracteriza que uma informação é autêntica é o fato desta pertencer a um subconjunto de mensagens válidas.

Em um sistema de chave pública, o esquema de Autenticação funciona da seguinte forma:

- O terminal A que deseja transmitir a informação usa a sua chave secreta k para cifrar os dados.

- O terminal B, ao receber o criptograma, utiliza a chave pública de A para decifrá-lo.

- Se a informação decifrada fizer sentido, ou seja, for válida, o terminal B atesta a autenticidade da mensagem, pois, A é o único a possuir a chave secreta k.

- A mensagem cifrada enviada por A não tem caráter sigiloso porque qualquer elemento na rede pode usar a chave pública de A, e recuperá-la.

Esquema de Autenticação de El Gamal

O esquema de autenticação de El Gamal explora a dificuldade de cálculo de logaritmos em corpos finitos.

O terminal A escolhe um número primo p e um elemento primitivo g, que gera um corpo GF(p). Em seguida, ele seleciona um inteiro aleatório $r \in$ GF(p), e calcula:

$$K = g^r \bmod p$$

Os inteiros K, g e p são tornados públicos.

Para autenticar a mensagem M, onde $M \in$ GF(p), o terminal A escolhe outro número aleatório $R \in$ GF(p), tal que mdc(R, p - 1) = 1, e computa

$$X = g^R \bmod p$$

O terminal A resolve, ainda, a congruência para Y:

$$M = (rX + RY) \bmod (p-1)$$

ou seja,

$$Y = \left(R^{-1}M - rX\right) \bmod (p-1)$$

O par (r, R) é mantido secreto, enquanto que a tripla (M, X, Y) é transmitida para o terminal B. *Observe que a mensagem é enviada às claras*.

O terminal B tendo recebido M, X e Y, calcula

$$A' = K^X X^Y \bmod p$$

e, autentica a mensagem se e somente se $A' = A''$

$$A'' = g^M \bmod p$$

Apesar do criptoanalista conhecer o valor de A'', para uma mensagem M desejada, é um problema difícil se obter *autenticadores do processo* X e Y válidos.

A justificativa do porquê o sistema funciona advém de

$$A = g^M = g^{rX} g^{RY}$$

como $K = g^r$ e $X = g^R$, a expressão anterior é igual a

$$A = K^X X^Y$$

Ex.: Considere o esquema de autenticação de El Gamal com $p = 11$, $g = 2$, e que o terminal A escolha aleatoriamente $r = 8$ e $R = 9$ que é primo de 8. Suponha que a mensagem enviada por A, a ser autenticada, seja $M = 5$. Qual serão os autenticadores X e Y enviados por A junto com a mensagem? Como B valida a mensagem recebida?

Solução:

A chave pública de A será, neste caso:

$$K = g^r \bmod p = 2^8 \bmod 11 = 3$$

O autenticador X é, então, calculado:

$$X = g^R = 2^9 \bmod 11 = 6$$

E, em seguida computa-se Y através da congruência, notando que $R^{-1} = 9^{-1} \bmod 10 = 9$:

$$Y = \left(R^{-1}M - rX\right) \bmod (p-1) = 9(5 - 8 \cdot 6) \bmod 10 = 3$$

O terminal B recebe (M, X, Y) = (5, 6, 3) e calcula:

$$A' = K^X X^Y \bmod p = 3^6 6^3 \bmod 11 = 10$$

e

$$A'' = g^M \bmod p = 2^5 \bmod 11 = 10$$

Como $A' = A''$, o terminal B considera a mensagem autêntica.

Exercício:

Considere o sistema de autenticação de El Gamal com $p = 11$ e $g = 2$, e que a chave pública do terminal A é $K = 8$. O terminal B recebe duas mensagens do terminal A juntamente com os seus autenticadores:

$$(M_1, X_1, Y_1) = (3, 7, 6)$$
$$(M_2, X_2, Y_2) = (4, 2, 1)$$

As mensagens M_1 e M_2 são válidas?

Esquema de Autenticação RSA

O esquema de autenticação RSA é semelhante ao sistema de cifragem RSA, com a diferença que são utilizadas as chaves do terminal transmissor ao invés das do receptor.

Desta forma, o terminal A obtém N através do produto de dois primos p e q, tal que $N = pq$. Em seguida A calcula

$$\lambda(N) = \text{mmc}(p-1, q-1)$$

Através da função gama, o transmissor está apto a calcular o par de chaves por meio da congruência

$$kK \equiv 1 \bmod \lambda(N)$$

O par (K, N) é divulgado publicamente.

Na transmissão o terminal A cifra a mensagem usando a sua chave secreta k

$$C = M^k \bmod N$$

O terminal B conhecendo a tripla (C, K, N) está apto a recuperar a mensagem

$$M = C^K \bmod N$$

Uma maneira do criptoanalista atacar este tipo de sistema, é gerando um criptograma aleatório C_1, tal que

$$C_1^K = M_1 \bmod N$$

e enviando-o ao usuário B, na tentativa de enganá-lo.

Entretanto se N é grande, apenas uma pequena parte dos números entre 0 e $N - 1$ constituirão mensagens válidas e, o criptoanalista estará gerando uma grande quantidade de criptogramas sem sentido.

Ex.: Um sistema bancário utiliza o esquema de autenticação RSA com $N = 77$. O correntista A tem a chave pública $K = 17$. Para autenticar uma transação, o computador central pede ao correntista o número da sua conta bancária. Como o sistema atesta a autenticidade do usuário, se a sua conta é $M_1, M_2, M_3, M_4 = 2, 2, 2, 2$?

Solução: Neste caso, a chave secreta de A é $k = 23$, logo

$$C_1, C_2, C_3, C_4 = 2^{23} \bmod 77 = 74$$

Os criptogramas $C_1, C_2, C_3, C_4 = 74, 74, 74, 74$ são enviados para B, que os decifra com auxílio da chave pública de A, obtendo

$$M_1, M_2, M_3, M_4 = 74^{17} \bmod 77 = 2$$

Como o número da conta bancária é a esperada, a operação é considerada autêntica.

Esquema de Cifragem e Autenticação RSA

O único sistema criptográfico conhecido que pode ser adaptado tanto para Autenticação como para Cifragem é o RSA.

Assim, é possível estruturar um sistema RSA que realize a cifragem e autenticação dos dados concatenadamente.

Suponha que o terminal A deseja enviar uma mensagem M *cifrada* e *autenticada* para o terminal B. Neste caso, A realiza autentica a mensagem através de sua chave secreta k_A e, em seguida, a criptografa usando a chave pública de B:

$$C_A = M^{k_A} \bmod N$$
$$C_C = (C_A)^{K_B} \bmod N = M^{k_A K_B} \bmod N$$

O terminal B ao receber o criptograma, procede na ordem inversa de A, e obtém:

$$(C_C)^{k_B} = M^{k_A K_B k_B} \bmod N \equiv M^{k_A} \bmod N = C_A$$
$$(C_A)^{K_A} = M^{k_A K_A} \bmod N = M$$

Assim, o único elemento na rede que é capaz de decifrar a mensagem é o terminal B, pois é o único a ter a chave secreta de decifragem k_B. Além disso, a autenticidade da mensagem não pode ser questionada, porque A é o único detentor da chave secreta de autenticação k_A.

Exercício:

Considere um sistema de cifragem e autenticação RSA com $N = 5 \cdot 7 = 35$. Sabendo-se que as chaves públicas de A e B são $K_A = 11$ e $K_B = 29$, determine o criptograma enviado pelo terminal A correspondente à mensagem $M = 3$. Calcule como o terminal B decifra e autentica a mensagem recebida.

Assinatura Digital

A assinatura digital tem por objetivo confirmar a autenticidade de um bloco de informações trocado entre dois usuários numa rede de comunicação.

As assinaturas digitais devem ter as características semelhantes às assinaturas humanas:

- As assinaturas devem ser únicas. A assinatura digital deve ser gerada apenas pelo seu usuário.
- As assinaturas digitais devem ser difíceis de falsificar.
- As assinaturas digitais devem ser fáceis de autenticar.
- Uma assinatura digital deve ser difícil de ser negada pelo seu originador.

As assinaturas digitais apresentam algumas diferenças em relação às assinaturas humanas:

- As assinaturas humanas são invariantes com o tempo, enquanto que as digitais precisam ser variantes, para evitar que um criptoanalista ataque o sistema, simplesmente gravando uma assinatura digital e retransmitindo-a posteriormente.

  Portanto, é interessante que as assinaturas dependam do conteúdo da mensagem e, também, do assinante.

- Além disso, é recomendável que as assinaturas digitais dependam do instante de tempo em que foram transmitidas, evitando que um criptoanalista possa reproduzí-las posteriormente, para um mesmo conteúdo de mensagem.

Existem, basicamente, dois métodos de verificação de autenticidade de uma assinatura digital:

1. *Método de Autenticação Direta*: O processo de autenticação é executado pelo próprio receptor da assinatura.

2. *Método de Autenticação Indireta*: O processo de autenticação é realizado por um árbitro na rede, que resolve as disputas entre os possíveis transmissores de um pacote de informação.

Métodos de Compressão

O método mais simples de geração de assinaturas é através de compressão de mensagens.

Do ponto de vista da eficiência de transmissão, é interessante que as assinaturas sejam menores que as mensagens. Com a utilização de técnicas de compressão de dados, é possível se obter este intento.

Em geral, qualquer método de compressão transforma uma mensagem M, de N bits, em um *resumo da mensagem* RS(M) de n bits, com $n < N$.

Os métodos de compressão devem garantir que a geração de dois resumos iguais para duas mensagens diferentes seja bastante improvável.

Esquema de Assinatura RSA

Assuma que o transmissor A queira assinar uma mensagem *M*. Inicialmente, a mensagem é passada por um compressor de dados, obtendo-se o resumo RS(*M*). A seguir, através da sua chave secreta k_A, ele cifra o resumo obtido, gerando, assim, a assinatura:

$$\mathrm{SN}(M) = k_A\{\mathrm{RS}(M)\}$$

Tanto a mensagem *M* quanto a assinatura SN(*M*) são transmitidas através da rede.

O receptor B tem condições de calcular o resumo da mensagem e compará-lo com o resumo decifrado através da chave pública de A, K_A:

$$\mathrm{RS}(M) = K_A\{\mathrm{SN}(M)\}$$

Se os dois resumos são idênticos, então, tanto a mensagem quanto a sua assinatura são válidos.

Bibliografia

[1] Almeida, Celso - Apostila de Criptografia - Unicamp, 1995

[2] Denning, D. E. R. - Criptography and Data Security - Addison Wesley, 1982.

[3] Diffie, W., Hellman, M. E. - Privacy and Authentication: An Introduction to Cryptography - Proc. IEEE, vol. 67, no. 3, Mar. 1979, pp. 397 -427.

[4] Merkle, R. C., Hellman, M. E. - Hiding Information and Signatures in Trap-Door Knapsacks - IEEE Trans. Inf. Theory, vol. IT24, Sep. 1978, pp. 525-530.

[5] National Bureau of Standards - Data Encryption Standard - FIPS, Publication no. 46, Jan. 1977.

[6] Rivest, R. L., Shamir, A., Adleman, L. - On Digital Signatures and Public Key Cryptosystems - Commun. ACM, vol 21, Feb. 1978, pp. 120-126.

[7] Seberry J., Pieprzyk J. - Cryptography: An Introduction to Computer Security - Prentice Hall, 1988.

[8] Shamir, A. - A Polynomial Time Algorithm for Breaking the Basic Merkle-Hellman Cryptosystem - IEEE 23rd Ann. Symp. Found. Comput. Sci., 1982, pp. 145-153.

[9] Sklar, Bernard - Digital Communications: Fundamentals and Applications - Prentice Hall, 1988.

[10] Van Tilborg, Henk C. A. - An Introduction to Criptology - Kluwer Academic Publishers, 1988.