CRIPTOGRAFIA MODERNA

Simétrica

Professor: Jiyan Yari

Histórico

Na criptografia mecânica é fundamental a ocultação pública da chave e também desejável manter segredo sobre a estrutura da máquina que produz a cifragem.

Com o desenvolvimento e aperfeiçoamento dos computadores e a incrível capacidade de realizar mais de um milhão de operações por segundo e a necessidade de uso da criptografia pelo comércio e bancos, os algoritmos criptográficos passam a ser de conhecimento público e o segredo a residir exclusivamente na chave.

Histórico

Em 1974, a IBM apresenta à agência oficial americana NBS (National Bureau of Standards) uma cifra que alguns pesquisadores vinham desenvolvendo desde 1960.

A NBS, após avaliar o algoritmo com a ajuda da NSA (National Security Agency), introduz algumas modificações, principalmente a redução na dimensão do espaço de chaves, e adota o código como padrão de cifragem de dados para os Estados Unidos.

O código passou a ser conhecido como DES (Data Encryption Standard).

No DES, apesar da exigência de redução imposta pela NBS, a quantidade de chaves distintas que
pode ser definida atinge 256, um número muito elevado.

NSA é o órgão oficial de segurança em comunicações do governo norte-americano. Fundada no início dos anos 50 do século XX pelo presidente Truman, é até hoje responsável oficial pela segurança em termos de criptografia nos Estados Unidos.

Histórico

A experiência acumulada pela NSA coloca-a anos à frente dos esforços públicos em criptografia.

No entanto, é interessante observar o contexto da intervenção da National Security Agency, solicitando a diminuição da dimensão do espaço de chaves do DES.

A NSA forçou a IBM a enfraquecer o sistema de tal forma que o governo americano pudesse eventualmente quebrar mensagens.
Naturalmente, a NSA, ainda hoje, nega o ocorrido.

Histórico

Inicialmente projetado pelos pesquisadores da IBM para atender a demanda dos bancos, o DES foi concebido para implementação em um computador.

O processo de cifragem é realizado em 19 etapas de aplicação de um al-goritmo definido pela chave.

Cada fase necessita de milhões de operações por segundo, portanto, só factíveis em um computador.

Histórico

Durante toda esta etapa, que cobriu a fase mecânica até o princípio da fase digital com o algoritmo DES, um aspecto permaneceu inalterado: a utilização de chaves privadas, caracterizando uma criptografia simétrica.

A chave que produz a mensagem cifrada é a mesma para decifrá-la.

Assim, como você já viu, esta é a principal fragilidade destes códigos.

O problema de distribuição de chaves torna-se difícil para o caso de compras através da internet ou troca de mensagens entre as pessoas que estão ligadas no ciberespaço.

Conversão de textos em código binário

Computadores só podem entender números.

Assim, todo processo de criptografia com o uso de sistemas computacionais, envolve uma pré-codificação em que o texto é transformado em uma seqüência de números.

Estes números são escritos em base 2 (notação binária), o que resulta em umaseqüência de bits.

Conversão de textos em código binário

Com relação à maneira com são aplicados à seqüências de bits da entrada, os sistemas criptográficos dividem-se em:

- Sistemas criptográficos de bloco (block ciphers): partem a mensagem em blocos de tamanho definido (tipicamente 64 bits). O algoritmo é aplicado em cada bloco, resultando em um bloco de bits de igual tamanho.

- Sistemas criptográficos de fluxo (stream ciphers): o algoritmo criptográfico é aplicado bit a bit (ou byte a byte) na seqüência de entrada de bits.

Conversão de textos em código binário

Como a mensagem é transformada em números?

código chamado ASCII (American Standard Code for Information Interchange).

Este código associa números de 0 a 127 a caracteres como '2', 'a', 'A' e '@' e algumas caracteres de controle, como shift e tecla de espaço.

Os caracteres imprimíveis recebem código de 32 a 126. Os outros códigos representam caracteres de controle.

Conversão de textos em código binário

Alguns códigos ASCII são dados na tabela abaixo:

Caracter	NULL	...	A	B	...	Z	...	a	...	DEL
Código ASCII	0		65	66		90		97		127

Conversão de textos em código binário

Com o tempo, o ASCII foi estendido para representar outros símbolos, como vogais acentuadas, algumas letras gregas e sinais especiais.

Para isto utiliza númerosde 0a 255. Naverdade há váriospadrõesem uso.

O padrão ISO-8859-1 é uma das extensões mais utilizadas nos países das Américas, Europa Ocidental e parte da África.

O caracter 'á', por exemplo, corresponde ao código 225 na extensão ISO-8859-1.

Conversão de textos em código binário

Como usamos números de 0 a 255, a notação binária de um código ASCII extendido usa 8 bits (= 1 byte).

Por exemplo, os códigos binários das letras maiúsculas estão representados na tabela a seguir:

A 01000001	N 01001110
B 01000010	O 01001111
C 01000011	P 01010000
D 01000100	Q 01010001
E 01000101	R 01010010
F 01000110	S 01010011
G 01000111	T 01010100
H 01001000	U 01010101
I 01001001	V 01010110
J 01001010	W 01010111
K 01001011	X 01011000
L 01001100	Y 01011001
M 01001101	Z 01011010

Conversão de textos em código binário

Então, por exemplo, a palavra AMOR é transformada no código binário:

A	M	O	R
01000001	01001101	01001111	01010010

Quando escrevemos a representação binária de caracteres com código de 0 a 127 é normalmente usamos apenas 7 bits, omitindo o bit mais significativo (o bit 0 à esquerda do número).

Operação XOR

Uma operação binária muito usada em circuitos eletrônicos e na descrição de sistemas criptográficos é a operação de ou exclusivo, também chamada de XOR.

A definição desta operação é:

$$0 \oplus 0 = 0$$
$$1 \oplus 0 = 1$$
$$0 \oplus 1 = 1$$
$$1 \oplus 1 = 0$$

Se 1 representa verdadeiro e 0 representa falso, e as sentenças S1 e S2 podem ter o valor verdadeiro ou falso, então $S1 \oplus S2$ é verdadeiro quando uma das duas é verdadeira e a outra é falsa (caso $1 \oplus 0$ e caso $0 \oplus 1$).

A operação XOR

Se 1 representa verdadeiro e 0 representa falso, e as sentenças S1 e S2 podem tero valor verdadeiroou falso, então S1$\oplus$S2 é verdadeiro quando uma das duas é verdadeira e a outra é falsa (caso 1$\oplus$0 e caso 0$\oplus$1).

A operação XOR

Se 1 representa verdadeiro e 0 representa falso, e as sentenças S1 e S2 podem tero valor verdadeiroou falso, então S1⊕S2 é verdadeiro quando uma das duas é verdadeira e a outra é falsa (caso 1⊕0 e caso 0⊕1).

Isto é o que se chama de "ou exclusivo", que é o sentido usual do conectivo ou na linguagem cotidiana.

Ou seja quando se diz:

“hoje à noite vou ao teatro ou ao cinema”

é porque pretende-se ir a um ou a outro, mas não aos dois.

A operação XOR

Um método criptográfico de substituição simples é a operação de XOR do texto claro (em forma binária) com uma chave escolhida K (em forma binária).

Por exemplo, se escolhemos a chave LIMA então:

Texto claro: AMOR	P	1000001 1001101 1001111 1010010
Chave: LIMA	K	1001100 1001001 1001101 1000001
Texto cifrado	$P \oplus K$	0001101 0000100 0000010 0010011

A operação XOR

O receptor desta mensagem usaria novamente a chave K=LIMA para decifrar a mensagem.
É fácil ver que se b1 e b2 são bits quaisquer,

$$(b1 \oplus b2) \oplus b2 = b1 \oplus (b2 \oplus b2) = b1$$

Assim, se:

$$C = P \oplus K \text{ então } C \oplus K = (P \oplus K) \oplus K = P \oplus (K \oplus K) = P)$$

Portanto, para decifrar a mensagem em código $C=P \oplus K$, basta nova operação de XOR com a chave K.

Algoritmo DES

Um dos algoritmos de criptografia convencional mais importantes para a história da criptografia é o chamado DES - Data Encryption Standard - Encriptação de Dados Padrão.

Este algoritmo é extremamente importante para compreendermos as técnicas utilizadas internamente pelos algoritmos de criptografia simétrica para garantir seu objetivo, que trata-se de um embaralhamento tão grande do texto claro, que não seja possível decifrá-lo a não ser que se conheça a chave correta ou que se tente todas as chaves possíveis.

Algoritmo DES

Força Bruta

Tentar todas as chaves possíveis, é o chamado ataque de força bruta.

O que garante que um algoritmo não seja vulnerável a tal ataque é o número de chaves que devem ser tentadas.

Se o número de chaves possíveis é muito grande, o esforço computacional envolvido no ataque pode torná-lo impraticável.

Algoritmo DES

Histórico do DES

A computação comercial se tornou uma realidade na década de 50.

Em 1951 haviam empresas que fabricavam computadores sob encomenda; em 1953 a IBM lançou seu primeiro computador.

Em 1957 a IBM introduziu a linguagem de programação FORTRAN.

Em 1959 os circuitos integrados foram inventados, o que iniciou uma nova era na computação em que computadores tornaram-se menores e mais baratos.

Algoritmo DES

Histórico do DES

A partir da década de 60 a criptografia com uso de computadores passou a ser utilizada por um grande número de empresas comerciais.

Uma empresa poderia talvez ter seus próprios protocolos criptográficos, que garantiam a segurança da informação armazenada e transmitida por meios eletrônicos dentro da empresa, mas não podia se comunicar eletronicamente de forma segura com outra empresa, a não ser que ambos usassem os mesmos sistemas criptográficos.

Faltava padronização ...

Algoritmo DES

Histórico do DES

Para resolver este problema, em 1973 a America National Bureau of Standards - ANBS (órgão do governo americano responsável por estabelecer padrões, assim como é o Inmetro no Brasil), fez uma chamada para propostas de sistemas criptográficos para proteger dados armazenados e durante a transmissão.

O objetivo era criar um sistema padrão que poderia ser usado por todas as empresas.

Algoritmo DES

Histórico do DES

Um dos algoritmos criptográficos mais estabelecidos na época, e um dos candidatos fortes a se tornar um padrão em criptografia, era o sistema conhecido como Lucifer da IBM.

Fazia parte da equipe de desenvolvimento um cientista chamado Horst Feistel, alemão que emigrou para os Estados Unidos em 1934.

Algoritmo DES

Histórico do DES

Quando os EUA entraram na Segunda Guerra Mundial, Feistel por ser alemão, foi colocado em prisão domiciliar até 1944.

Durante alguns anos Feistel evitou trabalhar com criptografia, mas acabou fazendo pesquisas com criptografia quando trabalhava para o Cambridge Research Center, da Força Aérea Americana.

Algoritmo DES

Histórico do DES

Isto trouxe-lhe problemas com a NSA (National Security Agency), que é o órgão do Governo Americano responsável por manter a segurança das comunicações militares e de governo dos EUA, por interceptar e decifrar comunicações de outros países.

Ainda hoje, a NSA é o líder mundial em interceptação de mensagem e espionagem, pois sabemos disso graças às confirmações de Edward Snowden.

Algoritmo DES

Histórico do DES

A NSA providenciou para que o projeto de Feistel fosse cancelado.

Feistel mudou para outra empresa, mas a NSA o forçou novamente a sair.

Por fim, Feistel foi trabalhar para aIBM,onde desenvolveu, no início da década de70, o sistema Lucifer.

Algoritmo DES

Histórico do DES

Considerado o sistema criptográfico mais forte da época, seria natural que Lucifer fosse adotado como padrão.

No entanto, a NSA interferiu novamente com o trabalho de Feistel.

Aparentemente, a NSA pressionou para que se Lucifer fosse adotado como padrão, então que fosse uma versão limitada dele, forçando para que o algoritmo usasse uma chave de apenas 56 bits.

Algoritmo DES

Histórico do DES

A versão limitada do Lucifer foi chamada DES e amplamente adotada na indústria e no setor bancário como padrão.

O ANSI (American National Standards Institute) adotou, em 1980, o uso do algoritmo DES para o setor bancário.

Algoritmo DES

Histórico do DES

A razão da NSA limitar o Lucifer, acredita-se, seria de que esta versão limitada seria forte o suficiente para o mundo civil, mas ainda dentro da capacidade deles de decifrar uma mensagem, considerando os recursos computacionais de que dispunham.

Tanto é que até pouco tempo atrás (final da década de 90), empresas americanas estavam proibidas por seu governo de exportar sistemas criptográficos “fortes”.

Algoritmo DES

Histórico do DES

FATO:

A perseguição da NSA ao trabalho de Feistel, não estava relacionado a ele ser alemão, mas ao fato de que estava desenvolvendo algo que limitava sua capacidade de interceptar e decifrar qualquer mensagem no mundo.

Algoritmo DES

O algoritmo DES Simplificado

O algoritmo DES simplificado em estudo tem finalidade apenas didática, não é mais seguro, mas mostra as propriedades e estrutura geral do DES.

Este algoritmo é chamado de DES simplificado ou S-DES.

Algoritmo DES

O algoritmo DES Simplificado

O S-DES é um algoritmo de cifra de bloco, isto é, toma blocos de bits como entrada e dá saída a blocos de bits de igual tamanho, que representa o texto cifrado.

Algortimo de Cifra:

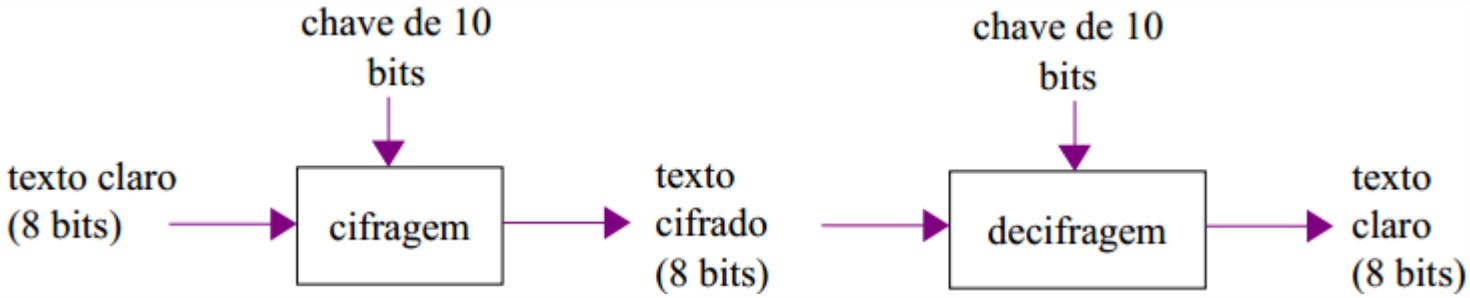
Uma cifra de bloco é um algoritmo criptográfico simétrico que transforma blocos de bits de tamanho fixo (que representa o texto claro) em blocos de bit de mesmo tamanho (representa o texto criptografado).

Algoritmo DES

O algoritmo DES Simplificado

O S-DES usa blocos de8 bits detexto claro, como 10001100, e uma chave de 10 bits, e produz um bloco de 8 bits de texto cifrado na saída.

Para decifrar a mensagem toma-se blocos de 8 bits de texto cifrado e a mesma chave de 10 bits, produzindo blocos de 8 bits de texto claro na saída.



Algoritmo DES

O algoritmo DES Simplificado

O algoritmo para criptografar cada bloco de 8 bits usa 5 estágios.

A saída de cada estágio é a entrada do estágio seguinte.

Os cinco estágios do algoritmo de cifragem são os seguintes:

1. Aplica-se um função de permutação **I P** (**I**nitial **P**ermutation).

2. aplica-se uma função f_{K1} , que depende da chave K e que envolve permutações e repetições;

Algoritmo DES

O algoritmo DES Simplificado

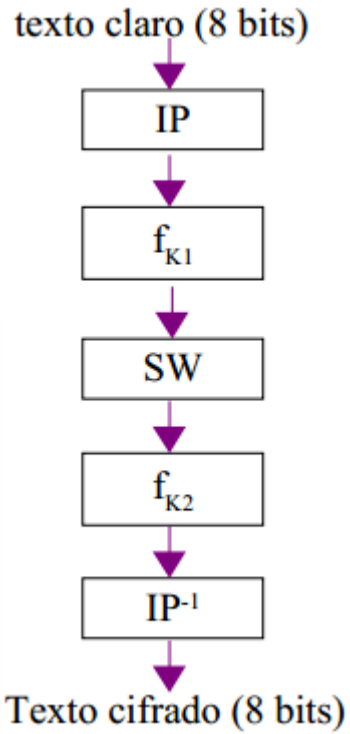
3. Aplica-se uma função SW que permuta as duas metades do bloco de bits;

4. Aplica-se uma função f_{K2}.
Note que é a mesma função, mas alimentada por uma chave diferente: K_2.

5. Aplicamos a permutação inversa de **IP** , isto é, aplicamos IP^{-1};

Algoritmo DES

O algoritmo DES Simplificado



Algoritmo DES

O algoritmo DES Simplificado

Se P é um bloco de 8 bits de texto claro, então o texto cifrado C é:

$$C = IP^{-1}(f_{K_2}(SW(f_{K_1}(IP(P)))))$$

A função $f_{K'}$ depende de uma chave K‘ de 8 bits.
Ela aparece 2 vezes no algoritmo de cifragem, mas com chaves diferentes.
Uma possibilidade seria usar duas chaves de 8 bits, totalizando 16 bits.
Outra possibilidade é usar uma chave de mais de 8 bits e, a partir dessa, gerar chaves de 8 bits.
Assim, temos um esquema de chaves e sub-chaves.
O S-DES usa uma chave K de 10 bits e dela extraímos duas chaves de 8 bits.

Algoritmo DES

O algoritmo DES Simplificado

A chave inicial sofre uma permutação P10 (Permutação dos 10 bits), em seguida selecionamos um sub-bloco de 8 bits. Este sofre uma permutação de 8 bits P8, o que resulta na primeira chave de 8 bits K1.

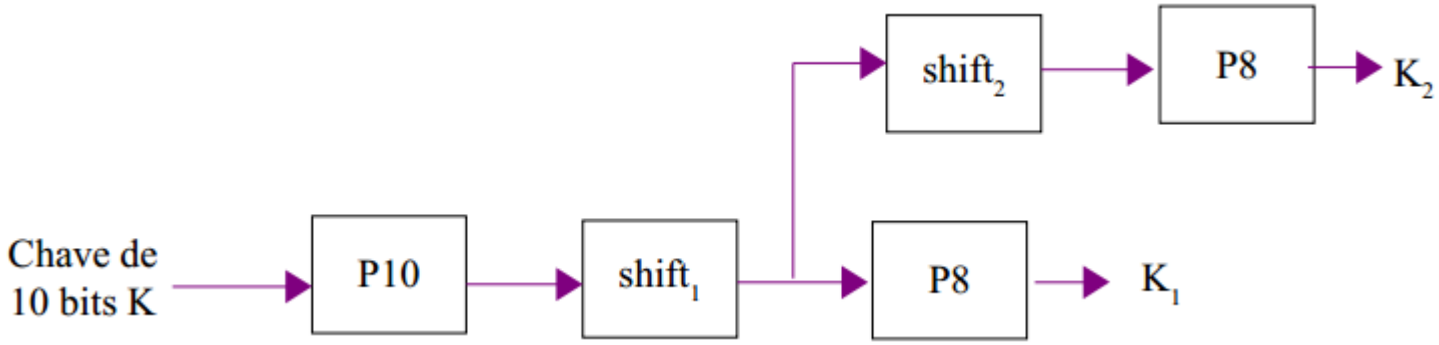
Selecionamos outro bloco de 8 bits e outra permutação, resultando na chave de 8 bits K2.

Os sub-blocos são obtidos através de operações de deslocamento de bits (Shift).

Algoritmo DES

O algoritmo DES Simplificado

O mecanismo de geração de chaves do DES é dado por:



$$K_1 = P8(shift1(P10(K)))$$

$$K_2 = P8(shift2(shift1(P10(K))))$$

Algoritmo DES

O algoritmo DES Simplificado

Representando a chave de 10 bits por (k1 k2 ... K10), a permutação P10 é dada por:
(k3,k5,k2,k7,k4,k10, k1,k9,k8,k6)

Quer dizer, o 3º bit (k3) de K ocupa a 1ª posição do bloco P10(K), o 5º bit (k5) de k ocupa a 2ª posição do bloco P10(K) e assim por diante.

Outra maneira de representar esta permutação é pela lista:

P10
3 5 2 7 4 10 1 9 8 6

Algoritmo DES

O algoritmo DES Simplificado

Esta lista deve ser lida da esquerda para a direita.

Por exemplo o 1º bit de P10(K) é o 3º bit de K , o 2º bit de P10(K) é o 5º bit de K , o 3º bit de P10(K) é o 2º bit de K , ... , o 10º bit de P10(K) é o 6º bit de K.

Por exemplo, se K = 1 0 1 1 0 0 1 0 1 0 , então:

P10(K) = 1 0 0 1 1 0 1 1 0 0

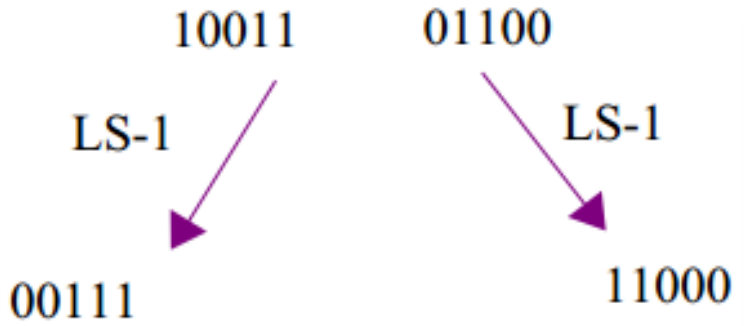
Tentar fazer este exemplo por conta própria.

Algoritmo DES

O algoritmo DES Simplificado

A operação de $shift_1$ é uma rotação circular esquerda (LS-1), operado separadamente nos 5 primeiro bits e nos 5 últimos bits.

Por exemplo:

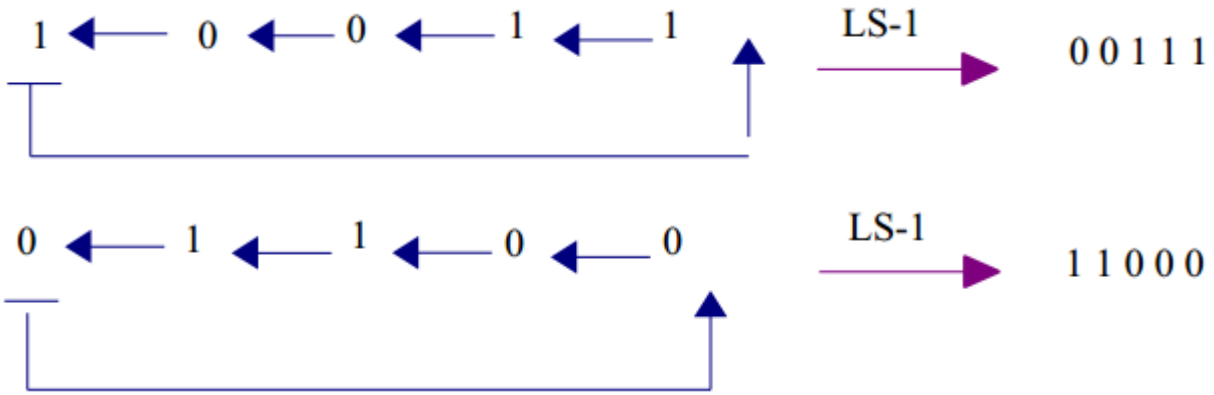


Assim $shift_1(P10(K))$ = 0 0 1 1 1 1 1 0 0 0

Algoritmo DES

O algoritmo DES Simplificado

LS-1 significa Left Shift1, isto é rotação circular esquerda de uma casa.



Assim:

$$shift_1(1\ 0\ 0\ 1\ 1\ 0\ 1\ \ 1\ 0\ 0) = (0\ 0\ 1\ 1\ 1\ \ 1\ 1\ 0\ 0\ 0)$$

Algoritmo DES

O algoritmo DES Simplificado

O próximo passo é a aplicação da permutação P8. Ela é usada para permutar e selecionar 8 bits de um bloco de 10 bits.

A permutação de 8 bits é:

P8

6 3 7 4 8 5 10 9

Então:

P8(0 0 1 1 1 1 1 0 0 0) = (1 1 1 1 0 1 0 0)

Neste exemplo, K_1 = 1 1 1 1 0 1 0 0

Algoritmo DES

O algoritmo DES Simplificado

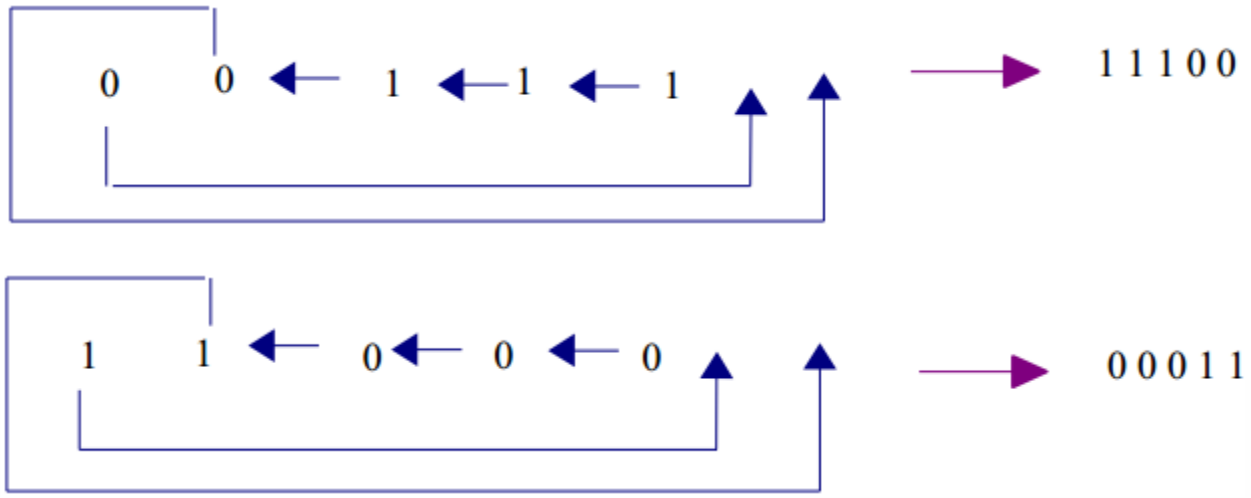
Voltando ao resultado da operação de shift1, aplicamos uma operação shift2, que é uma permutação circular à esquerda de 2 posições (LS-2), nos 5 primeiros bits e nos 5 últimos bits.

Como $shift_1(P10(K)) = 0011111000$, temos:

$$shift_2(0011111000) = (1110000011)$$

Algoritmo DES

O algoritmo DES Simplificado



Aplicando novamente P8 a este resultado, obtemos:

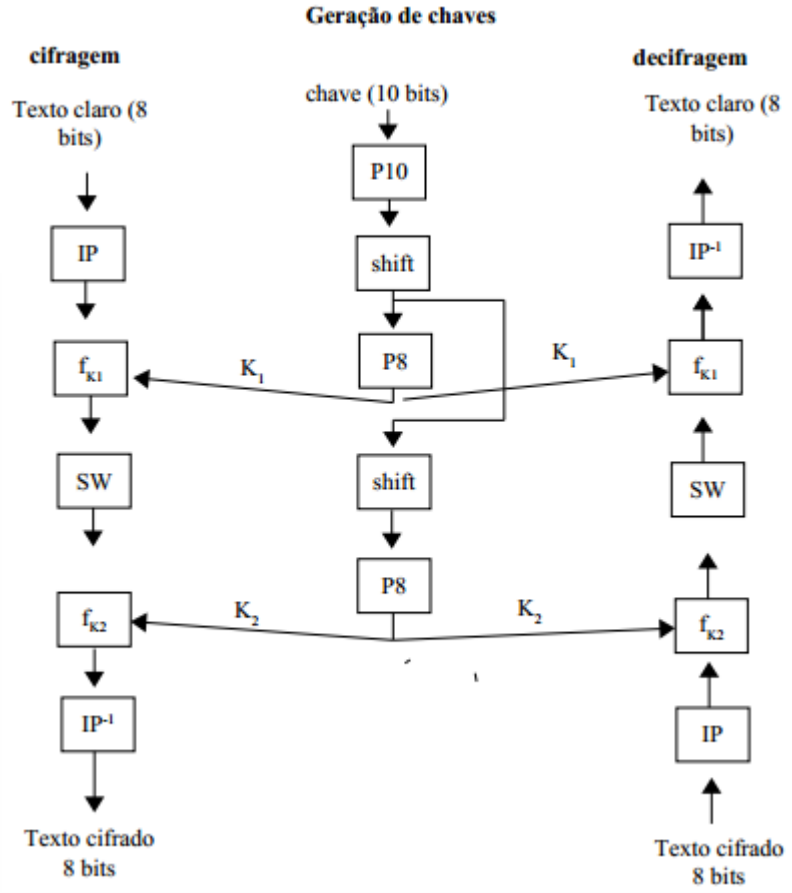
P8(1110000011) = 0100011

Assim, $K_2 = 0100011$

Algoritmo DES

O algoritmo DES Simplificado

Pode-se, portanto, descrever o S-DES no quadro:



Algoritmo DES

O algoritmo DES Simplificado

A função f_K

Agora resta estudar a função f_K, que é a única parte do algoritmo que depende da chave K. É a parte mais complicada e o coração do algoritmo.

A função f_K consiste de uma combinação de funções de permutações e substituições.

Algoritmo DES

O algoritmo DES Simplificado

A função f_K

Sejam L e R as funções que retornam os 4 bits mais à esquerda e mais à direita, respectivamente, de um bloco de 8 bits, teríamos:

$$L(10110101) = 1011$$

$$R(10110101) = 0101$$

Algoritmo DES

O algoritmo DES Simplificado

A função f_K

Na definição de f_K aparece outra função F que tem como entrada um bloco de 4 bits e como saída um bloco de 4 bits e usa uma subchave S_K (isto é, uma chave menor derivada da chave K).

Define-se f_K como:

$$fK(L,R)=(L \oplus F(R,SK), R)$$

onde $\oplus$ é o operador XOR (operador OR exclusivo bit a bit).

Algoritmo DES

O algoritmo DES Simplificado

A função f_K

Suponha que F(1101, SK) = (1110), onde SK é alguma subchave. Repare que ainda não se sabe a definição de F.

$$
\begin{aligned}
& f_K(10111101) \\
&= f_K(1011,1101) \\
&= (1011 \oplus F(1011,SK),1101) \\
&= (1011 \oplus 1110,1101) \\
&= (0101,1101) \\
&= 01011101
\end{aligned}
$$

Algoritmo DES

O algoritmo DES Simplificado

A função f_K

Agora vamos definir a função F, que é a parte mais complicada do algoritmo.

A função começa com uma permutação que expande o bloco de 4 bits da entrada para um bloco de 8 bits.

Esta permutação é chamada E/P (Expansion/Permutation):

E/P

4 1 2 3 2 3 4 1

Algoritmo DES

O algoritmo DES Simplificado

A função f_K

Vamos representar o resultado desta permutação pelo bloco:

(n4 n1 n2 n3 n2 n3 n4 n1)

Se a subchave é K1=(k11, k12, k13, k14, k15, k16, k17, k18), então adicionamos os dois blocos usando XOR, o que resulta em:

$$(n_4 \oplus k_{11}, n_1 \oplus k_{12}, n_2 \oplus k_{13}, n_3 \oplus k_{14}, n_2 \oplus k_{15}, n_3 \oplus k_{16}, n_4 \oplus k_{17}, n_1 \oplus k_{18}) = (p_{00}, p_{01}, p_{02}, p_{03}, p_{10}, p_{11}, p_{12}, p_{13})$$

Algoritmo DES

O algoritmo DES Simplificado

A função f_K

A partir deste bloco de 8 bits, o algoritmo avança para os S-Boxes.

Os primeiros 4 bits alimentam um processo chamado S-BOX S1 e os 4 últimos bits alimentam o S-BOX S2.

Algoritmo DES

O algoritmo DES Simplificado

A função f_K

O que são estes S-boxes?
São matrizes de 4x4 especificadas e um processo de substituição, que, com os bits de entrada, de uma maneira especificada, lê os bits de saída entre as entradas destas matrizes.
O uso dos S-boxes é um elemento fundamental do DES (que usa S-boxes maiores).
Muitos algoritmos modernos de criptografia usam S-boxes mais ou menos da mesma maneira que o DES e o S-DES que estamos estudando agora.

Algoritmo DES

O algoritmo DES Simplificado

A função f_K

As duas matrizes 4 x 4 são as seguintes:

$$S_0 = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 1 & 0 & 3 & 2 \\ 3 & 2 & 1 & 0 \\ 0 & 2 & 1 & 3 \\ 3 & 1 & 3 & 2 \end{bmatrix} \end{array} \qquad S_1 = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 0 & 1 & 2 & 3 \\ 2 & 0 & 1 & 3 \\ 3 & 0 & 1 & 0 \\ 2 & 1 & 0 & 3 \end{bmatrix} \end{array}$$

Algoritmo DES

O algoritmo DES Simplificado

A função f_K

Dos primeiros 4 bits (p00 p01 p02 p03), tomamos o primeiro e o quarto (p00 p03), consideramos este bloco de 2 bits um número na base 2 e verificamos o valor do número na base 10.

Fazemos o mesmo para o segundo e terceiro bits.

Algoritmo DES

O algoritmo DES Simplificado

A função f_K

Com isto temos 2 números, identificamos a entrada em S0 tendo estes dois números como número de linha e coluna.

Verificamos a entrada correspondente em S0 e passamos esta entrada para a base 2, resultando em um bloco de 2 bits.

Algoritmo DES

O algoritmo DES Simplificado

A função f_K

Por exemplo, se (p00 p01 p02 p03) = (0 1 0 0),
então
(p00 p03) = 0 (na base10)

e (p01 p02) = 2 (na base 10)

Na matriz S0, a entrada na linha 0 e coluna 2 é o número 3, que se escreve (11) na base 2.

Logo, temos:

$$(0\ 1\ 0\ 0) \xrightarrow{S_0\text{-BOX}} (1\ 1)$$

Algoritmo DES

O algoritmo DES Simplificado

A função f_K

O mesmo processo aplicado no BOX S1 resulta em mais 2 bits, o que compõe um bloco de 4 bits.

A este resultado aplicamos a permutação:

P4
2 4 3 1

O resultado de P4 é a saída da função F.

Algoritmo DES

O algoritmo DES Simplificado

A função f_K

Usando, por exemplo a chave K1=(1 0 1 0 0 1 0 0) a entrada (1 1 0 1) resulta no seguinte:

Aplicando E/P=[4 1 2 3 2 3 4 1] , obtemos:

N=[1 1 1 0 1 0 1 1]

Algoritmo DES

O algoritmo DES Simplificado

A função f_K

Somando a chave K1, obtemos:
$N \oplus K1 = (0\ 1\ 0\ 0\ 1\ 1\ 1\ 1)$

Os grupos de 4 bits são:
(0 1 0 0) - que entra no S_0-BOX e
(1 1 1 1) - que entra no S_1-BOX.

Algoritmo DES

O algoritmo DES Simplificado

A função f_K

Para o primeiro grupo de 4 bits, temos:

(0 0) (base 2) = 0 (base 10) → linha 0 da matriz S_0

(1 0) (base 2) = 2 (base 10) → coluna 2 da matriz S_0

Como $S_0[0,2] = 3$ e 3=(11) na base 2, então o Box S_0 resulta no bloco de 2 bits (1 1) para a entrada (0 1 0 0)

Algoritmo DES

O algoritmo DES Simplificado

A função f_K

Para o segundo grupo de 4 bits, temos:

(1 1) (base 2) = 3 (base 10) → linha 3 da matriz S_1

(1 1) (base 2) = 3 (base 10) → coluna 3 da matriz S_1

Como $S_1[3,3] = 3$ e 3 = (1 1) na base 2, então o Box S_1 resulta no bloco de 2 bits (1 1) para a entrada (1 1 1 1)

Algoritmo DES

O algoritmo DES Simplificado

A função f_K

Reunindo os dois resultados obtemos o bloco de 4 bits (1 1 1 1)

$$F(1\ 1\ 0\ 1, K1) = (1\ 1\ 1\ 1)$$

Observe que, como $f_K(L , R) = (L \oplus F(R , K), R)$

f_K opera somente nos 4 bits da direita.

Algoritmo DES

O algoritmo DES Simplificado

A função f_K

Por esta razão, aplicamos f_K duas vezes no processo de cifragem:
Aplicamos f_{K1}, em seguida a função de troca SW, que podemos definir como:

$$SW(L, R) = (R, L)$$

e depois f_{K2}, garantindo que todos os bits da entrada de 8 bits são devidamente embaralhados.

Algoritmo DES

O algoritmo DES Simplificado

Análise do S-DES

O grande problema do algoritmo S-DES proposto é o tamanho da chave.

As chaves são de 10 bits, logo, há somente $2^{10}=1024$ chaves possíveis.

Um ataque de força bruta é certamente possível: pode-se aplicar em um texto cifrado todas as chaves possíveis, e para cada chave, verifica-se se o resultado faz sentido.

Algoritmos de Criptografia Simétrica

Alguns dos principais algritmos de criptografia simétrica são:
- IDEA (128)
- BLOWFISH (32 a 448)
- RC2 (8 a 1024)
- RC4 (40 a 256)
- RC5 (32 a 2040)
- CAST (128)
- AES (128)
- TWOFISH (128)

Algoritmos de Criptografia Simétrica

International Data Encryption Algorithm (IDEA).

O IDEA é um cifrador simétrico de bloco, desenvolvido por Xuezia Lay e James Massey parao Instituto Federalde Tecnologiada Suiça (ETH – Zürich) e primeiramente descrito em 1991.

O cifrador é patenteado em diversos países mas é livre para uso não-comercial. O nome “IDEA” também é um nome registrado.

O IDEA é usado no PGP.

Algoritmos de Criptografia Simétrica

BLOWFISH

O Blowfish é um algoritmo de cifra de bloco desenvolvido por Bruce Schneier.
Foi publicado pela primeira vez em 1993.
Suas principais características são:
1. Rapidez – Ainda hoje, o Blowfish é um dos algoritmos de cifra de bloco em uso mais rápidos;
2. Pequeno tamanho – Blowfish pode rodar em menos de 5K de memória;
3. Simplificidade – A estrutura do Blowfish é simples, o que facilita sua implementação e a análise de sua segurança;

Algoritmos de Criptografia Simétrica

BLOWFISH

4. Segurança variável – o comprimento da chave pode ser escolhido, indo de 32 a 448 bits. Isto permite que o usuário escolha entre rapidez (com o uso de chaves menores) e segurança (uso de chaves maiores), dependendo das necessidades da aplicação.

Obs:
Blowfish não é objeto de patente e é distribuído livremente, o que contribui muito para sua grande popularidade no mundo do software criptográfico.

Algoritmos de Criptografia Simétrica

RC2

É um algoritmo de cifra de bloco desenvolvido por Ron Rivest, em 1987.
O termo RC vem de “Rivest Cipher” (O cifrador de Rivest). Este Ron Rivest é o mesmo Rivest do RSA (Rivest-Shamir-Adlerman).
O desenvolvimento do RC2 foi financiado pela Lotus, que buscava um algoritmo criptográfico que pudesse ser incorporado ao seu pacote Lotus Notes.
Uma das características é que o algoritmo deveria ter um tamanho de chave pequeno e ser, por isso, vulnerável.
A Lotus vendia o Lotus Notes mundo afora e é bom lembrarmos que até 1996, as leis americanas proibiam a exportação de software criptográfico que usasse chaves de mais de 40 bits.

Algoritmos de Criptografia Simétrica

RC4

O RC4 é um algoritmo de cifra de fluxo amplamente utilizado.

Ele está embutido, por exemplo, nos protocolos SSL (Secure Socket Layer) − protocolo utilizado para proteção de tráfego na Internet − e no protocolo WEP − utilizado em redes sem fio.

O RC4 usa chaves de tamanho variável, tipicamente entre 40 e 256 bits.

Algoritmos de Criptografia Simétrica

RC5

Criado por Ron Rivest é o RC5. Trata-se de um algoritmo de cifra de bloco, com tamanho de bloco variável (32, 64 ou 128 bits), tamanho de chave variável (ate 2040 bits) e número de iterações do processo variável (de 0 a 255 vezes).

A escolha sugerida é de blocos de 64 bits, chave de 128 bits e 12 rounds (iterações da etapa principal).

É um algoritmo de bloco de Feistel notável por sua simplicidade.

Toda a rotina de cifragem e decifragem pode ser descrita em poucas linhas de código.

Algoritmos de Criptografia Simétrica

CAST-128

É um algoritmo de cifra de bloco usado em diversos produtos e protocolos.
É o algoritmo padrão em algumas versões do PGP. Foi criado em 1996 por Carlisle Adams e Stafford Tavares.
O CAST-128 é um algoritmo de Feistel, com 12 a 16 iterações da etapa principal, tamanho de bloco de 64 bits e chave de tamanho variável (40 a 128 bits, com acréscimos de 8 bits).
Os 16 rounds são usados quando a chave tem comprimento maior que 80 bits.
O algoritmo é patenteado, mas pode ser usado livremente, sem pagamento de licença.

Algoritmos de Criptografia Simétrica

AES

O algoritmo Rijndael, rebatizado AES, de advanced Encryption Standard, foi adotado pelo governo americano como protocolo padrão de criptografia e adotado pelo NIST (National Institue os Standards and Technology) como o substituto oficial do DES.
O algoritmo foi publicado em 1998, desenvolvido por dois criptógrafos belgas, Joan Daemen e Vincent Rijmen.
É um algoritmo rápido, tanto em software quanto em hardware, relativamente fácil de ser implementado e requer pouca memória.
Desde sua escolha como padrão, sua adoção em produtos que envolvem criptografia tem crescido muito.

Algoritmos de Criptografia Simétrica

AES

O algoritmo Rijndael, rebatizado AES, de advanced Encryption Standard, foi adotado pelo governo americano como protocolo padrão de criptografia e adotado pelo NIST (National Institue os Standards and Technology) como o substituto oficial do DES.
O algoritmo foi publicado em 1998, desenvolvido por dois criptógrafos belgas, Joan Daemen e Vincent Rijmen.
É um algoritmo rápido, tanto em software quanto em hardware, relativamente fácil de ser implementado e requer pouca memória.
Desde sua escolha como padrão, sua adoção em produtos que envolvem criptografia tem crescido muito.

Algoritmos de Criptografia Simétrica

TWOFISH

É uma das poucas cifras incluídas no OpenPGP.
O Twofish é uma chave simétrica que emprega a cifra de bloco de 128 bits, utilizando chaves de tamanhos variáveis, podendo ser de 128, 192 ou 256 bits.
Ele realiza 16 interações durante a criptografia, sendo um algoritmo bastante veloz.
A cifra Twofish não foi patenteada estando acessível no domínio público, como resultado, o algoritmo Twofish é de uso livre para qualquer um utilizar sem restrição.