3.0 - Criptografia

prof. Fábio Engel

fabioe@utfpr.edu.br

Conteúdo

1. Criptografia
2. Algoritmos de chave simétrica
   - Data Encryption Standard (DES)
   - Triple DES
   - Advanced Encryption Standard (AES)
3. Algoritmos de chave assimétrica
   - RSA
4. Autenticação de mensagem e funções de hash
   - Criptografia de mensagens
   - Código de autenticação de mensagens (MAC)
   - Função de Hash
5. Princípios de Certificados e Gerenciamento de chaves

Criptografia

Criptografia

- Criptografia é a técnica de gerar e transmitir informação de maneira que apenas as partes originalmente envolvidas na comunicação (destinatário e remetente) sejam capazes de conhecer esta informação.
- A pesquisa de métodos de criptografia e estabelecimento de conversações seguras é de importância fundamental na área da computação, principalmente no que se refere às redes.

Criptografia

- Com o crescimento na utilização de sistemas que lidam com informações críticas - serviços bancários e servidores de arquivos (residencial ou empresarial), por exemplo - a garantia de requisitos mínimos de segurança, privacidade e autenticidade é fundamental.
- O fluxo básico de ações de um processo de criptografia pode ser visto a partir da figura a seguir.

Criptografia

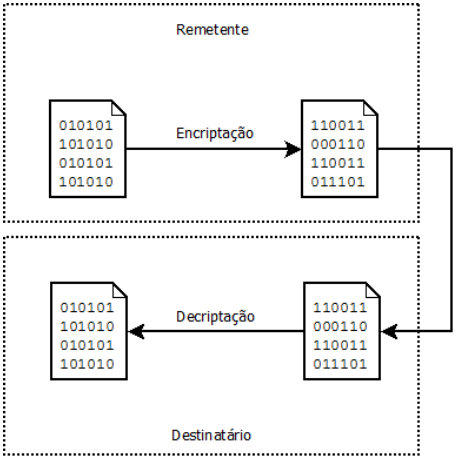


Figura 1: Fluxo do Processo de Criptografia

Criptografia

- Quando se trata de segurança da informação, a criptografia pode satisfazer a estes requisitos:
  - Confidencialidade
  - Integridade
  - Autenticação
  - Não repúdio

- Não repúdio permite que você impeça uma parte de negar uma declaração ou ação prévia. Exemplo:
  - Suponha que um investidor envie a seguinte mensagem a um corretor: "Compre 1000 ações de XYZ a 50". Logo depois que a bolsa executa o pedido, as ações de XYZ caem para 20. O investidor nega a ordem de compra e diz que, na realidade, era uma ordem de venda. É preciso comprovar que determinada parte realmente originou uma mensagem específica em uma hora específica.

Criptografia

- As técnicas de criptografia podem ser divididas em duas categorias principais - as que envolvem a técnica de chave simétrica e as que envolvem a técnica de chaves assimétricas (pública e privada). Estas técnicas serão discutidas a seguir.

Algoritmos de chave simétrica

Algoritmos de chave simétrica

- A criptografia simétrica, também chamada de criptografia convencional ou criptografia de chave única, era o único tipo de criptografia em uso antes do desenvolvimento da criptografia por chave pública na década de 70.
- Este continua sendo o mais usado dos dois tipos de criptografia.
- Antes de começar, definimos alguns termos a partir de [1]:
- Uma mensagem original é conhecida como **texto claro**, enquanto que a mensagem codificada é chamada de **texto cifrado**. O processo de converter texto claro em texto cifrado é conhecido como **cifragem** ou **criptografia**; restaurar o texto claro a partir do texto cifrado é a **decifragem** ou **decriptografia**.

Algoritmos de chave simétrica

- Os muitos esquemas utilizados para a criptografia constituem a área de estudo conhecida como **criptografia**. Esse esquema é conhecido como **sistema criptográfico** ou **cifra**.
- As técnicas empregadas para decifrar uma mensagem sem qualquer conhecimento dos detalhes de criptografia estão na área da **criptoanálise**.
- A criptoanálise é o que os leigos chama de "quebrar o código".
- As áreas da criptografia e criptoanálise juntas são chamadas de **criptologia**.

Algoritmos de chave simétrica

- Um esquema de criptografia simétrica possui cinco ingredientes (representado na figura a seguir):
- **Texto claro**: Mensagem de dados originais, inteligíveis, alimentados no algoritmo como entrada.
- **Algoritmo de criptografia**: Realiza diversas substituições e transformações no texto claro.
- **Chave secreta**: Também é a entrada para o algoritmo de criptografia. A chave é um valor independente do texto claro e do algoritmo. O algoritmo produzirá uma saída diferente, dependendo da chave específica sendo usada no momento.

Algoritmos de chave simétrica

- **Texto cifrado**: Essa é a mensagem embaralhada, produzida como saída. Ela depende do texto claro e da chave secreta. Para determinada mensagem, duas chaves diferentes produzirão dois textos cifrados diferentes. O texto cifrado é um fluxo de dados aparentemente aleatório e, nesse formato, é ininteligível.
- **Algoritmo de decriptografia**: Esse é basicamente o algoritmo de criptografia executado de modo inverso. Ele toma o texto cifrado e a chave secreta e produz o texto claro original.

Algoritmos de chave simétrica

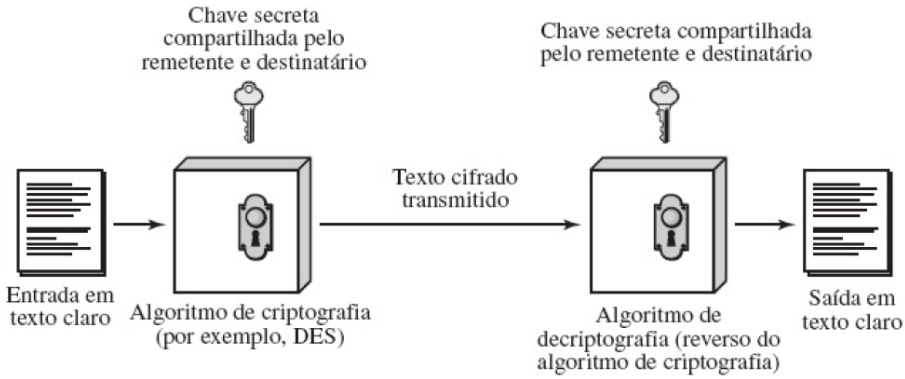


Figura 2: Modelo simplificado da criptografia convencional

Algoritmos de chave simétrica

- Existem dois requisitos para o uso seguro da criptografia convencional:
  1. Precisamos de um algoritmo de criptografia forte. No mínimo, queremos que um oponente que conheça o algoritmo e tenha acesso a um ou mais textos cifrados seja incapaz de decifrar o texto cifrado ou descobrir a chave.
  2. Emissor e receptor precisam ter cópias da chave secreta de uma forma segura e precisam manter a chave protegida. Se alguém puder descobrir a chave e souber o algoritmo, toda a comunicação usando essa chave poderá ser lida.

Algoritmos de chave simétrica

- Consideramos que é impraticável descriptografar uma mensagem com base no texto cifrado mais o conhecimento do algoritmo de criptografia/descriptografia.
- Em outras palavras não precisamos manter o algoritmo secreto: precisamos manter apenas a chave secreta.
- Essa característica da criptografia simétrica é o que a torna viável para uso generalizado.

Algoritmos de chave simétrica - Criptografia

- Os sistemas criptográficos são caracterizados em três dimensões independentes:
  1. **O tipo das operações usadas para transformar texto claro em texto cifrado**. Todos os algoritmos de criptografia são baseados em dois princípios gerais: **substituição**, em que cada elemento no texto claro (bit, letra, grupo de bits ou letras) é mapeado em outro elemento, e **transposição**, em que os elementos no texto claro são reorganizados. O requisito fundamental é que nenhuma informação seja perdida.
  2. **O número de chaves usadas**. Se tanto o emissor quanto o receptor utilizarem a mesma chave, o sistema é considerado como criptografia simétrica, de chave única, de chave secreta, ou convencional. Se emissor e receptor usarem chaves diferentes, o sistema é considerado de criptografia assimétrica, de duas chaves ou de chave pública.
  3. **O modo como o texto claro é processado**. Uma **cifra de bloco** processa a entrada de um bloco de elementos de cada vez, produzindo um bloco de saída para cada bloco de entrada. Uma **cifra em fluxo** processa os elementos da entrada continuamente, produzindo a saída de um elemento de cada vez, enquanto prossegue.

Algoritmos de chave simétrica - Criptoanálise

- Normalmente, o objetivo de atacar um sistema de criptografia é recuperar a chave em uso, em vez de simplesmente recuperar o texto claro de um único texto cifrado. Existem duas técnicas gerais para o ataque a um esquema se criptografia convencional:
  1. **Criptoanálise**: Conta com a natureza do algoritmo e talvez mais algum conhecimento das característica gerais do texto claro, ou ainda alguns pares de amostra de texto claro e texto cifrado. Este tipo explora as características do algoritmo para tentar deduzir um texto claro específico ou deduzir a chave utilizada.
  2. **Ataque por força bruta**: O atacante experimenta cada chave possível em um trecho do texto cifrado, até obter uma tradução inteligível para texto claro.

Algoritmos de chave simétrica - Criptoanálise

- Um ataque por força bruta envolve a tentativa de cada chave possível até que seja obtida uma tradução inteligível de texto cifrado para texto claro. A tabela a seguir mostra o tempo envolvido para diversos espaços de chaves.
- Os resultados são mostrados para quatro tamanhos de chave binária. O tamanho de chave de 56 bits é usado com o algoritmo DES, e o tamanho de chave de 168 bits é usado para o triple DES.

Tamanho da chave (bits)	Número de chaves alternativas	Tempo necessário para 1 decriptografia/μs
32	$2^{32} = 4{,}3 \times 10^9$	2^{31} μs = 35,8 minutos
56	$2^{56} = 7{,}2 \times 10^{16}$	2^{55} μs = 1142 anos
128	$2^{128} = 3{,}4 \times 10^{38}$	2^{127} μs = $5{,}4 \times 10^{24}$ anos
168	$2^{168} = 3{,}7 \times 10^{50}$	2^{167} μs = $5{,}9 \times 10^{36}$ anos

Figura 3: Tempo médio exigido para busca completa da chave

Algoritmos de chave simétrica - Técnicas de Substituição

- O estudo das técnicas de criptografia clássica nos permite ilustrar as técnicas básicas da criptografia simétrica usadas hoje e os tipos de ataques criptoanalíticos que precisam ser antecipados.
- Os dois componentes básicos de todas as técnicas de criptografia são **substituição** e **transposição**.
- Uma técnica de **substituição** é aquela em que as letras de texto claro são substituídas por outras letras ou por números ou símbolos:

Algoritmos de chave simétrica - Técnicas de Substituição

- O uso mais antigo que conhecemos de uma cifra de substituição, e o mais simples, foi feito por Júlio César. A cifra de César consiste em substituir cada letra do alfabeto pela letra que fica três posições adiante no alfabeto. Por exemplo:
  - claro: hello world
  - cifra: khoor zruog
- Podemos definir a transformação listando todas as possibilidades, da seguinte forma:

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

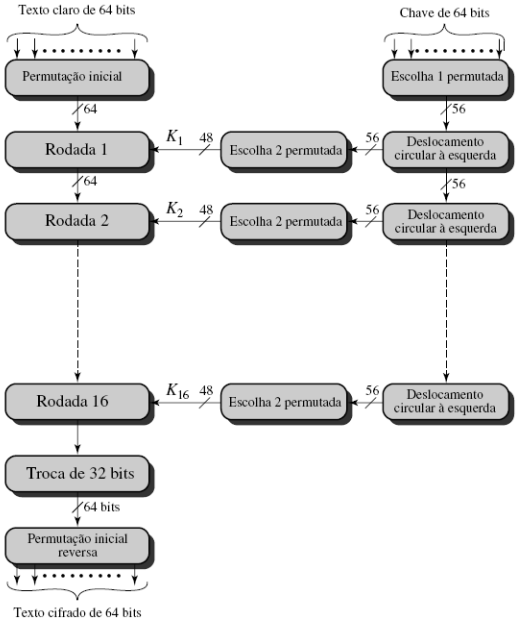
Data Encryption Standard (DES)

- O esquema de criptografia mais usado é baseado no *Data Encryption Standard* (DES), adotado em 1977 pelo *National Bureau of Standards*.
- Para DES, os dados são codificados em blocos de 64 bits usando uma chave de 56 bits.
- O algoritmo transforma a entrada de 64 bits em uma série de etapas em uma saída de 64 bits.
- As mesmas etapas, com a mesma chave, são usadas para reverter a criptografia.

DES - Funcionamento

- O esquema geral para a criptografia DES está ilustrado na Figura a seguir.
- Assim como em qualquer esquema de criptografia, existem duas entradas na função de criptografia: o texto claro a ser codificado e a chave.
- Neste caso, o texto claro precisa ter 64 bits de extensão e a chave tem 56 [1] bits de extensão.

[1] Na realidade, a função espera uma chave de 64 bits como entrada. Porém, somente 56 desses bits são usados; os outros 8 bits podem ser usados como bits de paridade ou simplesmente definidos arbitrariamente


Texto claro de 64 bits
Chave de 64 bits
Permutação inicial
Escolha 1 permutada
64
56
Rodada 1
K_1
48
Escolha 2 permutada
56
Deslocamento circular à esquerda
64
56
Rodada 2
K_2
48
Escolha 2 permutada
56
Deslocamento circular à esquerda
Rodada 16
K_{16}
48
Escolha 2 permutada
56
Deslocamento circular à esquerda
Troca de 32 bits
64 bits
Permutação inicial reversa
Texto cifrado de 64 bits

DES - Funcionamento

- Examinando o lado esquerdo da figura, podemos ver que o processamento do texto claro prossegue em três fases.
- Primeiro, o texto claro de 64 bits passa por uma permutação inicial (IP), que reorganiza os bits para produzir a entrada permutada.
- Isto é seguido por uma fase consistindo em 16 rodadas da mesma função, que envolve funções de permutação e substituição.
- A saída da última (décima sexta) rodada consiste em 64 bits que são uma função do texto claro de entrada e da chave.
- As metades esquerda e direita da saída são trocadas para produzir a pré-saída.
- Finalmente, a pré-saída é passada por uma permutação (IP^{-1}) que é o inverso da função permutação inicial, para produzir o texto cifrado de 64 bits.

DES - Funcionamento

- A parte direita da figura anterior, mostra o modo como a chave de 56 bits é usada.
- Inicialmente, a chave é passada por uma função de permutação.
- Depois, para cada uma das 16 rodadas, uma subchave (k_i) é produzida pela combinação de um deslocamento circular à esquerda e uma permutação;
- A função de permutação é a mesma para casa rodada, mas uma subchave diferente é produzida, devido aos deslocamentos repetidos dos bits da chave.

DES - Funcionamento

- **A permutação inicial** e seu inverso são definidos por tabelas, como mostram as Tabelas a seguir.
- As tabelas devem ser interpretadas da seguinte maneira. A entrada de uma tabela consiste em 64 bits numerados de 1 a 64. As 64 entradas na tabela de permutação contém uma permutação dos números de 1 a 64. Cada entrada na tabela de permutação indica a posição de um bit de entrada numerado na saída, que também consistem em 64 bits.

Tabelas de permutação para DES

(a) Permutação inicial (IP)

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

(b) Permutação inicial inversa (IP^{-1})

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

Figura 5: Tabelas de permutação para DES

(c) Permutação de expansão (E)

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

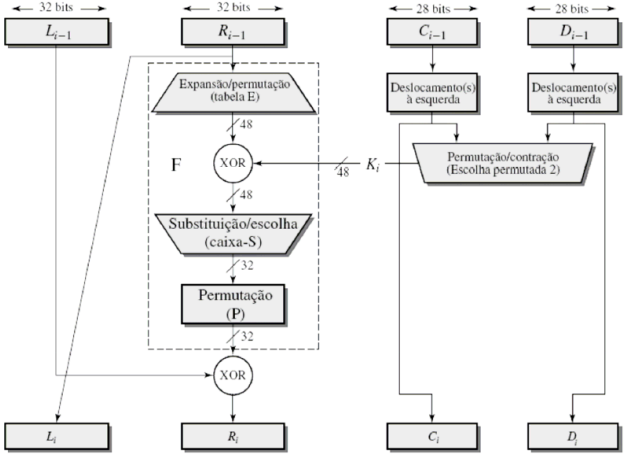
(d) Função de permutação (P)

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

Figura 6: Tabelas de permutação para DES

DES - Funcionamento

- **Detalhes de uma rodada individual**



DES - Funcionamento

- **Detalhes de uma rodada individual**
  - Novamente, comece focando o lado esquerdo do diagrama.
  - As metades esquerda e direita de cada valor intermediário de 64 bits são tratadas como quantidades de 32 bits separadas, rotuladas com L e R.
  - A chave da rodada K_I é de 48 bits. A entrada R é de 32 bits. Essa entrada R é expandida inicialmente para 48 bits usando uma tabela que define uma permutação mais uma expansão que envolve duplicação de 16 dos bits de R. É realizado XOR dos 48 bits resultantes com os bits de K_i. Esse resultado de 48 bits passa por uma função de substituição que produz uma saída de 32 bits, que é permutada conforme definido pela tabela da figura 6.d.

DES - Funcionamento

- **Detalhes de uma rodada individual**
  - O papel das caixas-S na função F está ilustrado na Figura 7.
  - A substituição consiste em um conjunto de oito caixas-S, cada uma aceitando 6 bits como entrada e produzindo 4 bits como saída. Essas transformações são definidas na Tabela presentes nas Figuras 8 e 9
    - O primeiro e o último bits da entrada para a caixa S_i formam um número binário de 2 bits para selecionar uma das quatro substituições definidas pelas quatro linhas na tabela para S_i.
    - Os quatro bits do meio selecionam uma das dezesseis colunas.
    - O valor decimal na célula selecionada pela linha e coluna é então convertido em sua representação de 4 bits para produzir a saída.

DES - Funcionamento

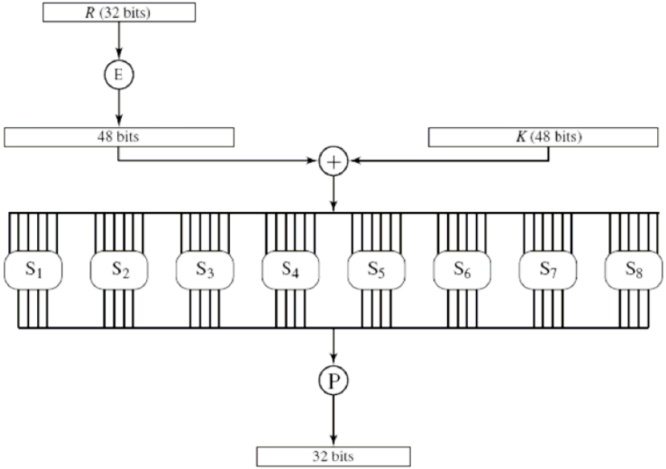


Figura 7: Cálculo de F(R,K)

Função de substituição

S_1

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

S_2

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

S_3

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

S_4

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

Figura 8: Caixas S

DES - Funcionamento

Função de substituição

S_5

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

S_7

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

S_6

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

S_8

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

Figura 9: Caixas S

DES - Funcionamento

- **Geração de chave**
  - Uma chave de 64 bits é usada como entrada para o algoritmo.
  - Os bits da chave são numerados de 1 até 64; cada oitavo bit é ignorado.
  - A chave é primeiro submetida a uma permutação controlada por uma tabela rotulada como Escolha Permutada Um (Figura 10).
  - A chave de 56 bits resultante é então tratada como duas quantidades de 28 bits, rotuladas como C_0 e D_0. Em cada rodada, D_{i-1} e C_{i-1} estão sujeitos separadamente a um deslocamento esquerdo circular, ou rotação, de 1 ou 2 bits, conforme determinado pela tabela contida na Figura 11.d.
  - Esses valores deslocados servem como entrada para a rodada seguinte. Eles também servem como entrada para a Escolha Permutada Dois (Figura 11.c), que produz uma saída de 48 bits.

(b) Escolha Permutada Um (PC-1)

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

Figura 10: Escalonamento de chaves do DES

DES - Funcionamento

(c) Escolha Permutada Dois (PC-2)

14	17	11	24	1	5	3	28
15	6	21	10	23	19	12	4
26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

(d) Escalonamento de Deslocamentos à Esquerda

Número da rodada	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Bits rotacionados	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Figura 11: Escalonamento de chaves do DES

DES - O efeito avalanche

- Uma propriedade desejável de qualquer algoritmo de criptografia é que uma pequena mudança no texto claro ou na chave produza uma mudança significativa no texto cifrado. Em particular, uma mudança em um bit do texto claro ou um bit da chave deverá produzir uma mudança em muitos bits do texto cifrado.
- Se a mudança fosse pequena, esta poderia oferecer um modo de reduzir o tamanho do texto claro ou o espaço de chave a ser pesquisado.

DES - O efeito avalanche

- O DES exibe um forte efeito avalanche. A tabela a seguir mostra alguns resultados. Na tabela fora utilizados dois textos claros que diferem por um bit:
  - 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
  - 10000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
- Com a chave
  - 0000001 1001011 0100100 1100010 0011100 0011000 0011100 0110010

(a) Mudança no texto claro		(b) Mudança na chave	
Rodada	Número de bits que diferem	Rodada	Número de bits que diferem
0	1	0	0
1	6	1	2
2	21	2	14
3	35	3	28
4	39	4	32
5	34	5	30
6	32	6	32
7	31	7	35
8	29	8	34
9	42	9	40
10	44	10	38
11	32	11	31
12	30	12	33
13	30	13	28
14	26	14	26
15	29	15	34
16	34	16	35

Figura 12: Efeito Avalanche no DES

A força do DES

- Com um tamanho de chave de 56 bits, existem 2^{56} chaves possíveis, o que é aproximadamente 7.2×10^{16} chaves. Assim, um ataque de força bruta parece ser impraticável.
- Supondo que, na média, metade do espaço de chave tenha de ser pesquisado, uma única máquina realizando uma criptografia DES por microssegundo levaria mais de mil anos para quebras a cifra.
- Porém, a suposição de uma criptografia por microssegundo é bastante conservadora. Desde 1977, Diffie e Hellman postularam que existia tecnologia para montar uma máquina paralela com 1 milhão de dispositivos de criptografia, cada um podendo realizar uma criptografia por microssegundo. Isso reduziria o tempo médio de busca para cerca de 10 horas.
- Os autores estimaram que o custo seria de aproximadamente 20 milhões de dólares em 1977.

A força do DES

- O DES, finalmente e definitivamente, provou ser inseguro em julho de 1998, quando a Electronic Frontier Foundation (EFF) anunciou que tinha quebrado uma criptografia DES usando uma máquina "decifradora de DES" de uso especial, montada por menos de 250 mil dólares. O ataque levou menos de três dias.
- A EFF publicou uma descrição detalhada da máquina, permitindo que outros montassem seu próprio decifrador. E, naturalmente, os preços dos hardwares continuarão a cair enquanto as velocidades aumentam, tornando o DES praticamente inútil.
- Felizmente, existem várias alternativas ao DES, sendo que as mais importantes são triple DES e AES, explicados a seguir.

Triple DES

- No início de 1979, a IBM percebeu que o tamanho da mensagem DES era muito pequeno e criou uma forma de aumentá-lo usando a criptografia tripla [2].
- O método escolhido, que desde então foi incorporado ao padrão internacional 8732, está ilustrado na figura a seguir.

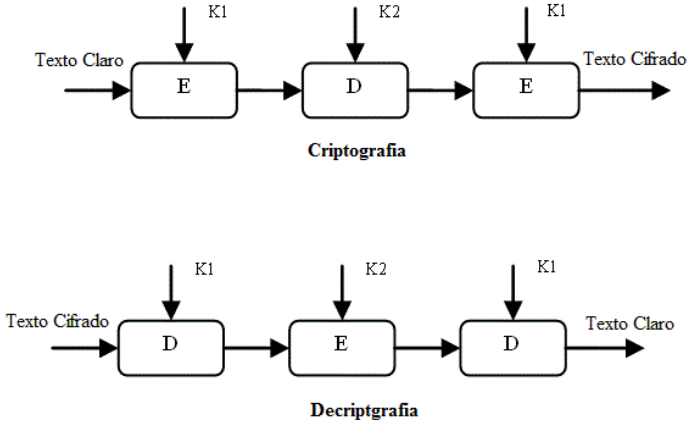


Figura 13: Criptografia e Decriptografia tripla usando o DES

Triple DES

- Na figura, K_1 e K_2 representam as chaves e E e D representam as ações de Criptografar e Decriptografar respectivamente.
- No primeiro estágio, o texto simples é criptografado com K_1 da maneira usual do DES.
- No segundo estágio, o DES é executado no modo de descriptografia, com o uso de K_2 como chave.
- Por fim, outra criptografia é feita com K_1

Triple DES

- Este projeto levanta duas questões: Primeiro, por que são utilizadas apenas duas chaves em vez de três? Segundo, por que foi usado **EDE (Encrypt Decrypt Encrypt)**, em vez de **EEE (Encrypt Encrypt Encrypt)**?
- São utilizadas duas chaves porque até mesmo os criptógrados mais paranóicos concordam que 112 bits serão suficientes para aplicações comerciais durante um bom tempo [2]. O uso de 168 bits só criaria overhead desnecessário de gerenciar e transportar outra chave, com pouco ganho real.
- O motivo para criptografar, descriptografar e criptografar mais uma vez é a compatibilidade retroativa com os sistemas DES de chave única. Do ponto de vista da criptografia, os dois mapeamentos são igualmente fortes. No entanto, ao user EDE em vez de EEE, um computador que utiliza a criptografia tipla pode comunicar com outro que utiliza a criptografia simples apenas definindo $K_1 = K_2$

Triple DES

- O 3DES possui dois atrativos que garantem seu uso generalizado pelos próximo anos.
- Primeiro, com seu tamanho de chave de 168 bits, ele contorna a vulnerabilidade do ataque por força bruta ao DES.
- Segundo, o algoritmo de criptográfia básico no 3DES é igual ao do DES, esse algoritmo foi submetido a mais análises detalhadas que qualquer outro algoritmo de criptografia por um longo período, e nenhum ataque criptoanalítico eficaz contra o algoritmo, além da força bruta, teve sucesso.

Triple DES

- A principal desvantagem do 3DES é que o algoritmo é relativamente lento em software.
- O DES original foi projetado para implementação em hardware em meados da década de 1970, e não produz um código de software eficiente.
- O 3DES, que tem três vezes mais rodadas que o DES, é correspondentemente mais lento.
- Uma desvantagem secundária é que tanto o DES quanto o 3DES utilizam um tamanho de bloco de 64 bits. Por motivo de eficiência e de segurança, um tamanho de bloco maior é desejável.

Triple DES

- Por causa dessas desvantagens,o 3DES não é um candidato razoável para uso em longo prazo.
- Como alternativa, o NIST (National Institute of Standards and Technology) em 1997 pediu propostas para um novo Advanced Encryption Standard (AES).

Advanced Encryption Standard (AES)

- Segundo a proposta, **Advanced Encryption Standard (AES)** deveria ter um grau de segurança igual ou superior ao 3DES e uma eficiência bem melhorada.
- Além desses requisitos gerais, o NIST especificou que o AES deveria ser uma cifra de bloco simétrica com uma tamanho de bloco de 128 bits e suporte para tamanhos de chave de 128, 192 e 256 bits.
- Em uma primeira rodada de avaliações, 15 algoritmos propostos foram aceitos. Uma segunda rodada estreitou a competição para 5 algoritmos, e por fim,e em novembro de 2001, o NIST selecionou o Rijndael como o algoritmo AES.

Advanced Encryption Standard (AES)

- A proposta do Rijndael para o AES definiu uma cifra em que o tamanho do bloco e o tamanho da chave podem ser especificados independentemente como 128, 192 ou 256 bits. A especificação do AES usa as mesmas três alternativas de tamanho de chave, mas limita o tamanho do bloco a 128 bits.
- Como o DES, o Rijndael utiliza substituição e permutações, e também emprega várias rodadas. O número de rodadas depende do tamanho da chave e do tamanho do bloco, sendo 10 para chaves de 128 bits com blocos de 128 bits, passando para 14 no caso da maior chave ou do maior bloco.
- No entanto, diferente do DES, todas as operações envolvem bytes inteiros, a fim de permitir implementações eficientes, tanto em hardware quanto em software

Advanced Encryption Standard (AES)

- O algoritmo foi projetado não só por segurança, mas também para aumentar a velocidade.
- Uma boa implementação de software em uma máquina de 2 GHz deve ser capaz de alcançar uma taxa de criptografia de 700 Mbps, que é rápida o suficiente para codificar mais de 100 vídeos MPEG-2 em tempo real.
- As implementações de hardware são ainda mais rápidas.

Outros algoritmos criptográficos

- A tabela abaixo mostra algumas dentre as cifras de chave simétrica mais comuns.

Cifra	Autor	Comp. da chave	Comentários
Blowfish	Bruce Schneir	1 a 448 bits	Velho e lento
DES	IBM	56 bits	Muito fraco para usar agora
IDEA	Massey e Xuejia	128 bits	Bom, mas patenteado
RC4	Ronald Rivest	1 a 2048 bits	Algumas chaves são fracas
RC5	Ronal Rivest	128 a 256 bits	Bom, mas patenteado
AES	Daemen e Rijmen	128 a 256 bits	Melhor escolha
Serpent	Anderson, Bihan, Knudsen	128 a 256 bits	Muito forte
3DES	IBM	168 bits	Segunda melhor escolha
Twofish	Bruce Schneier	128 a 256	Muito forte, amplamente utilizado

Algoritmos de chave assimétrica

Algoritmos de chave assimétrica

- Historicamente, o problema da distribuição de chaves sempre foi o elo mais fraco da maioria dos sistemas de criptografia.
- Independente de quanto um sistema de criptografia fosse sólido, se um intruso conseguisse roubar a chave, o sistema acabava sendo inútil.
- Como todos os criptólogos sempre presumem que a chave de criptografia e a chave de decriptografia são iguais (ou facilmente derivadas uma da outra) e que a chave é distribuída a todos os usuários do sistema, tinha-se a impressão de que havia um problema inerente ao sistema: as chaves tinha de ser protegidas contra roubo, mas também tinham de ser distribuídas; portanto, elas não podiam ser simplesmente trancadas no caixa-forte de um banco.

Algoritmos de chave assimétrica

- Em 1976, dois pesquisadores da University of Stanford, Diffie e Hellman (1976), propuseram um sistema de criptografia radicalmente novo, no qual as chaves de criptografia e de descriptografia eram diferentes, e a chave de descritografia não podia ser derivada da chave de criptografia.
- Em sua proposta, o algoritmo de criptografia E e o algoritmo de descriptografia D tinham de atender a três requisitos:
  1. $D(E(P)) = P$
  2. É extremamente difícil deduzir a chave K_1 a partir da chave K_2.
  3. E não pode ser decifrado por um ataque de texto simples escolhido.

Algoritmos de chave assimétrica

- O primeiro requisito diz que, se aplicarmos D a uma mensagem criptografada, $E(P)$, obteremos outra vez a mensagem de texto simples original P. Sem essa propriedade, o destinatário legitimo não poderia decodificar o texto cifrado.
- O segundo é auto-explicativo.
- O terceiro é necessário porque, como veremos a seguir, os intrusos podem experimentar o algoritmo até se cansarem. Sob essas condições, não há razão para a chave criptográfica não se tornar pública

Algoritmos de chave assimétrica

- O método funciona da seguinte forma: uma pessoa, digamos Alice, desejando receber mensagens secretas, primeiro cria dois algoritmos que atendem aos requisitos anteriores. O algoritmo de criptografia e a chave de Alice se tornam públicos, daí o nome criptografia de chave pública.
- Por exemplo, Alice poderia colocar sua chave pública na *home page* que ela tem na Web. Usaremos a notação E_A para indicar o algoritmo de criptografia parametrizado pela chave pública de Alice. De modo semelhante, o algoritmo de descriptografia parametrizado pela chave privada de Alice é D_A. Bob faz o mesmo, publicando E_B, mas mantendo secreta a chave D_B.

Algoritmos de chave assimétrica

- Agora vamos ver se podemos resolver o problema de estabelecer um canal seguro entre Alice e Bob, que nunca haviam tido um contato anterior. Supondo que tanto a chave de criptografia de Alice, E_A, quanto a chave de criptografia de Bob, E_B, estejam em arquivos de leitura pública. Agora, Alice pega sua primeira mensagem P, calcula $E_B(P)$ e a envia para Bob.
- Em seguida, Bob a descriptografa aplicando sua chave secreta D_B, ou seja, ele calcula $D_B(E_B(P)) = P$
- Ninguém mais pode ler a mensagem criptografada $E_B(P)$, porque o sistema de criptografia é considerado sólido e porque é muito difícil derivar D_B da chave E_B publicamente conhecida.
- Para enviar uma resposta R, Bob transmite $E_A(R)$. Agora, Alice e Bob podem se comunicar com segurança.

Algoritmos de chave assimétrica

- Talvez seja interessante fazer uma observação sobre a terminologia. A criptografia de chave pública exige que cada usuário tenha duas chaves: uma chave pública, usada pelo mundo inteiro para criptografar as mensagens a serem enviadas para esse usuário, e uma chave privada, que o usuário utiliza para descriptografar mensagens.
- Faremos referência a essas chaves como chave pública e chave privada, respectivamente, e vamos distingui-las das chaves secretas (também chamadas chaves simétricas) usadas na criptografia de chave simétrica convencional.

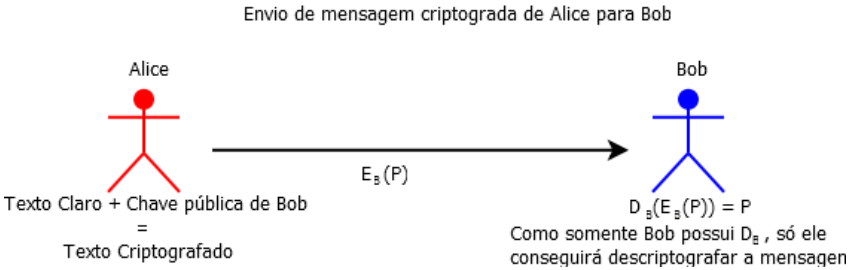


Figura 14: Sistema de Criptografia Diffie Helman

RSA

- O único problema é que temos de encontrar algoritmos que realmente satisfaçam a todos os três requisitos. Devido às vantagens potenciais da criptografia de chave pública, muitos pesquisadores estão se dedicando integralmente a seu estudo, e alguns algoritmos já foram publicados.
- Um método muito interessante foi descoberto por um grupo de pesquisadores do MIT e é conhecido pelas iniciais dos três estudiosos que o criaram (Rivest, Shamir, Adleman): **RSA**.
- Ele sobreviveu a todas as tentativas de rompimento por mais de um quarto de século e é considerado um algoritmo muito forte. Grande parte da segurança prática se baseia nele.
- Sua principal desvantagem é exigir chaves de pelo menos 1024 bits para manter um bom nível de segurança (em comparação com 128 bits para os algoritmos de chave simétrica), e isso o torna bastante lento.

RSA

- O método RSA se baseia em alguns princípios da teoria dos números, o método pode ser resumido nos seguintes passos:
  1. Escolha dois números primos extensos, p e q (geralmente, de 1024 bits).
  2. Calcule $n = p \times q$ e $z = (p - 1) \times (q - 1)$.
  3. Escolha um número d tal que z e d sejam co-primos (tenham apenas o número 1 como divisor comum).
  4. Encontre e de forma que $e \times d = 1 \bmod z$

RSA

- Com esse parâmetros calculados, divida o texto claro (considerado um string de bits) em blocos, de modo que cada mensagem de texto claro P fique no intervalo $0 \leq P < n$
- Isso pode ser feito agrupando-se o texto simples em blocos de k bits, onde k é o maior inteiro para o qual a desigualdade $2^k < n$ é verdadeira.
- Para criptografar a mensagem P, calcule $C = P^e (\text{mod } n)$
- Para descriptografar C, calcule $P = C^d (\text{mod} n)$

- É possível provar que, para todo P na faixa especificada, as funções de criptografia e descriptografia são inversas entre si. Para realizar a criptografia, você precisa de e e n, enquanto para a descriptografia, são necessários d e n. Portanto, a **chave pública** consiste no par (e, n) e a **chave privada** consiste em (d, n).

- A segurança do método se baseia na dificuldade de fatorar números extensos. Se pudesse fatorar o valor n (publicamente conhecido), o criptoanalista poderia então encontrar p e q e, a partir desses, encontrar z. Com o conhecimento de z e e, é possível encontrar d utilizando-se o algoritmo de Euclides. Felizmente, os matemáticos têm tentado fatorar números extensos por pelo menos 300 anos, e o conhecimento acumulado sugere que o problema é extremamente difícil.

- De acordo com Rivest e seus colegas, a fatoração de um número de 500 dígitos requer 1025 anos, usando-se a força bruta. Nesse caso, eles pressupõem o melhor algoritmo conhecido e um computador com um tempo por instrução de 1 s. Mesmo que os computadores continuem a se tornar cada vez mais rápidos na proporção de uma ordem de magnitude por década, ainda se passarão séculos até que a fatoração de um número de 500 dígitos se torne viável e, nesse tempo, nossos descendentes poderão simplesmente escolher p e q ainda maiores.

Autenticação de mensagem e funções de hash

Autenticação de mensagem e funções de hash

- A autenticação de mensagens é um procedimento para verificar se as mensagens recebidas provêm da origem afirmada.
- A autenticação da mensagem pode ainda verificar a sequência e o instante correto de envio.
- Qualquer mecanismo de autenticação de mensagens possui dois níveis de funcionalidade. No nível mais baixo, é preciso haver algum tipo de função que produza um autenticador: um valor a ser usado para autenticar a mensagem. Essa função é então usada como uma primitiva em um protocolo de autenticação de nível mais alto, que permite a um receptor verificar a autenticidade de uma mensagem.

Autenticação de mensagem e funções de hash

- As funções usada para produzir um autenticador podem ser agrupadas em 3 classe:
- Criptografia de mensagens;
- Código de autenticação de mensagens (MAC - Message Authentication Code);
- Função de Hash;

Criptografia de mensagens

- A **criptografia de mensagens** por si só pode oferecer uma medida de autenticação. No entanto, a análise difere para esquemas de criptografia de chave simétrica e pública.
- Considere o uso da criptografia simétrica. Uma mensagem M transmitida da origem A para o destino B é criptografada usando uma chave secreta K compartilhada por A e B. Se nenhuma outra parte conhecer a chave, então a confidencialidade é obtida: nenhuma outra parte pode recuperar o texto claro da mensagem.
- Além disso, podemos dizer que B tem garantias de que a mensagem foi gerada por A, pois A é a única outra parte que possui K e, portanto, a única outra parte com a informação necessária para construir o texto cifrado que pode ser decriptografado por K.

Criptografia de mensagens

- Assim, podemos dizer que a criptografia simétrica oferece autenticação, bem como confidencialidade. Porém essa afirmação precisa ser qualificada.
- Dada uma função de decriptografia D e uma chave secreta K, o destino aceitará qualquer entrada X e produzirá saída $Y = D(K, X)$.
- Se X for o texto cifrado de uma mensagem legítima M produzida pela função de criptografia correspondente, então Y será alguma mensagem de texto claro M. Caso contrário, Y provavelmente será uma sequência de bits sem significado.
- Podem ser necessário alguns meios automatizados para determinar em B se Y é texto claro legítimo e, por consequência, proveniente de A.

Criptografia de mensagens

- Considerando agora a criptografia de chave pública, seu uso oferece confidencialidade, mas não autenticação.
- A origem (A) utiliza a chave pública PU_b do destino (B) para criptografar M. Como somente B tem a chave privada correspondente PR_b, somente B pode decriptografar a mensagem.
- Este esquema não oferece autenticação, pois qualquer oponente também pode usar a chave pública de B para criptografar uma mensagem, afirmando ser A.

Criptografia de mensagens

- Para oferecer autenticação, A utiliza sua chave privada para criptografar a mensagem, e B usa a chave pública de A para decriptografá-la. Isso oferece autenticação usando o mesmo tipo de raciocínio do caso da criptografia simétrica.
- Novamente, o mesmo raciocínio de antes se aplica: é preciso haver alguma estrutura interna no texto claro para que o receptor possa distinguir entre o texto claro bem formado e os bit aleatórios.
- Se tal estrutura existe, então este esquema realmente oferece autenticação, ele também fornece o que é conhecido como assinatura digital.
- Note que esse esquema não oferece confidencialidade, uma vez que qualquer um da possa da chave pública de A pode decriptografar o texto cifrado.

Criptografia de mensagens

- Para oferecer confidencialidade e autenticação, A pode criptografar M primeiro usando sua chave privada, o que resulta na assinatura digital e, depois usando a chave pública de B, o que fornece a confidencialidade.
- A desvantagem dessa abordagem é que o algoritmo de chave pública, que é complexo, precisa ser usado quatro vezes, em vez de duas, em cada comunicação.

Criptografia de mensagens

A → B: E(K, M)

- Oferece confidencialidade
  — somente A e B compartilham K
- Oferece um grau de autenticação
  — Só poderia vir de A
  — Não foi alterada em trânsito
  — Requer alguma formatação/redundância
- Não oferece assinatura
  — Receptor poderia forjar mensagem
  — Emissor poderia negar mensagem

(a) Criptografia simétrica

A → B: E(PU_b, M)

- Oferece confidencialidade
  — Somente B tem PR_b para decriptografar
- Não oferece autenticação
  — Qualquer parte poderia usar PU_b para criptografar a mensagem e afirmar ser A

(b) Criptografia de chave pública (simétrica): confidencialidade

$A \rightarrow B: E(PR_a, M)$

- Oferece autenticação e assinatura
- — Somente A tem PR_a para criptografar
- — Não foi alterada em trânsito
- — Requer alguma formatação/redundância
- — Qualquer parte pode usar PU_a para verificar a assinatura

(c) Criptografia de chave pública: autenticação e assinatura

$A \rightarrow B: E(PU_b, E(PR_a, M))$

- Oferece confidencialidade por causa de PU_b
- Oferece autenticação e assinatura por causa de PR_a

(d) Criptografia de chave pública: confidencialidade, autenticação e assinatura

Figura 16: Consequência de confidencialidade e autenticação da criptografia da mensagem

Código de autenticação de mensagens (MAC)

- Uma técnica de autenticação alternativa envolve o uso de uma chave secreta para gerar um pequeno bloco de dados de tamanho fixo, conhecido como soma de verificação criptográfica ou MAC, que é anexado a mensagem.
- Essa técnica assume que duas partes na comunicação, digamos A e B, compartilham a chave secreta K. Quando A tem uma mensagem para enviar a B, ele calcula o MAC como uma função da mensagem e da chave: $\text{MAC} = C(K, M)$ onde:
  - M = mensagem de entrada;
  - C = função MAC;
  - K = chave secreta compartilhada;
  - MAC = código de autenticação de mensagem (Message Authentication Code);

Código de autenticação de mensagens (MAC)

- A mensagem mais o MAC são transmitidos ao destinatário desejado. O destinatário realiza o mesmo cálculo sobre a mensagem recebida, usando a mesma chave secreta, para gerar um novo MAC.
- O MAC recebido é comparado com o MAC calculado. Se considerarmos que somente o receptor e o emissor conhecem a chave secreta, e se o MAC recebido for igual ao MAC calculado, então:
  1. O receptor tem garantias de que a mensagem não foi alterada. Se um atacante alterar a mensagem, mas não alterar o MAC, então o cálculo do MAC pelo receptor será diferente do MAC recebido;
  2. O receptor tem garantias de que a mensagem é do emissor declarado. Como ninguém mais sabe a chave secreta, ninguém mais poderia preparar uma mensagem com um MAC apropriado.
  3. Se a mensagem inclui um número de sequência, então o receptor pode ter certeza da sequência apropriada, pois um atacante não poderá alterar o número da sequência.

Código de autenticação de mensagens (MAC)

- Uma função MAC é semelhante à criptografia. Uma diferença é que o algoritmo MAC não precisa ser reversível, como precisaria ser para a decriptografia.
- Em geral, a função MAC é uma função muitos-para-um.
- O domínio da função consiste de mensagens de comprimento arbitrário, enquanto a imagem consiste de todos os MACs possíveis e todas as chaves possíveis.
- Se um MAC de n bits for usado, então existem 2^n MACs possíveis, enquanto existem N mensagens possíveis com $N \gg 2^n$. Além disso, com uma chave de k bits, existem 2^k chaves possíveis.

Código de autenticação de mensagens (MAC)

- Por exemplo, suponha que estejamos usando mensagens de 100 bits e um MAC de 10 bits. Então, existe um total de 2^{100} mensagens diferentes, mas somente 2^{10} MACs diferentes. Assim, em média, cada valor de MAC é gerado por um total de $2^{100}/2^{10} = 2^{90}$ mensagens diferentes.
- Se uma chave de 5 bits for usada, então existem $2^5 = 32$ mapeamentos diferentes do conjunto de mensagens para o conjunto de valores MAC.
- Acontece que, devido às propriedades matemáticas da função de autenticação, ela é menos vulnerável a ser quebrada do que a criptografia.

Código de autenticação de mensagens (MAC)

- O processo representado na figura a seguir (a) oferece autenticação, mas não confidencialidade, pois a mensagem como um todo é transmitida às claras. A confidencialidade pode ser obtida realizando-se a criptografia da mensagem depois (b) ou antes (c) do algoritmo MAC.
- Nos dois casos, duas chaves distintas são necessárias, cada qual compartilhada pelo emissor e o receptor.

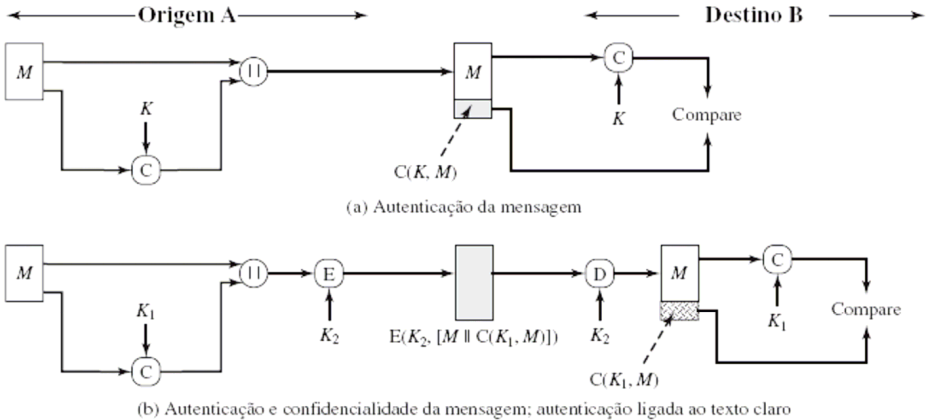
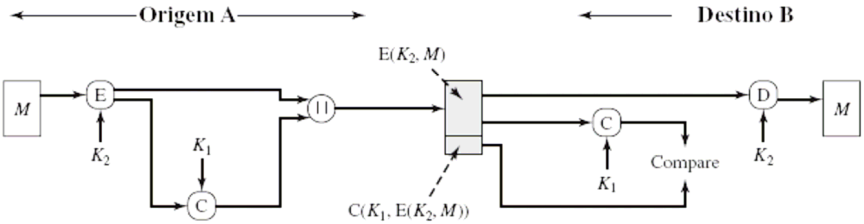


Figura 17: Uso básicos do código de autenticação de mensagens (MAC)



(c) Autenticação e confidencialidade da mensagem; autenticação ligada ao texto cifrado

Figura 18: Uso básicos do código de autenticação de mensagens (MAC)

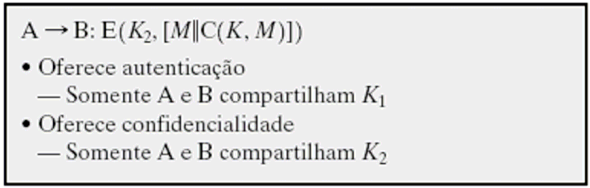
Código de autenticação de mensagens (MAC)

- No primeiro caso, o MAC é calculado tendo a mensagem como entrada, e depois é concatenado à mensagem. O bloco resultante é, então, criptografado.
- No segundo caso, a mensagem é criptografada antes. O MAC é então calculado a partir do texto cifrado resultante e é concatenado ao texto cifrado para formar o bloco transmitido. Normalmente, é preferível ligar a autenticação diretamente ao texto claro, de modo que o método da figura b é utilizado.

Código de autenticação de mensagens (MAC)

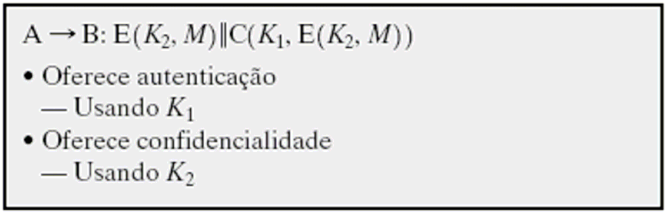
- Finalmente, observe que o MAC não oferece uma assinatura digital, pois emissor e receptor compartilham a mesma chave.
- A tabela a seguir resume as implicações de confidencialidade e autenticação das técnicas ilustradas na figura anterior.

(a) Autenticação da mensagem



(b) Autenticação e confidencialidade da mensagem: autenticação ligada ao texto claro

Figura 19: Usos básico do código de autenticação de mensagens



(c) Autenticação e confidencialidade da mensagem: autenticação ligada ao texto cifrado

Figura 20: Usos básico do código de autenticação de mensagens

Código de autenticação de mensagens (MAC)

- Em criptografia, **HMAC (Hash-based Message Authentication Code)** é uma construção específica para calcular o código de autenticação de mensagem (MAC) envolvendo uma função hash criptográfica em combinação com uma chave secreta.
- Da mesma forma que em qualquer MAC, este pode ser usado para simultaneamente verificar tanto a integridade como a autenticidade de uma mensagem. Qualquer função hash criptográfica, tal como MD5 ou SHA-1, pode ser usada no cálculo do HMAC; o algoritmo MAC resultante é denominado HMAC-MD5 ou HMAC-SHA1 em conformidade.
- A força criptográfica do HMAC depende da criptográfica da função hash subjacente, do tamanho do hash produzido como saída em bits, e do tamanho e da qualidade da chave criptográfica.

Função de Hash

- Uma variação no código de autenticação de mensagens é a função de hash unidirecional.
- Assim como o código de autenticação de mensagens, uma função de hash aceita uma mensagem de comprimento variável M como entrada e produz uma saída de comprimento fixo, conhecida como código de hash $H(M)$.

Função de Hash

- Diferentemente de um MAC, um código de hash não usa uma chave, sendo uma função apenas da mensagem de entrada.
- O código de hash também é conhecido como síntese de mensagem ou valor de hash.
- O código de hash é uma função de todos os bits da mensagem e oferece uma capacidade de detecção de erros: uma mudança em qualquer bit ou bits na mensagem resulta em uma mudança no código de hash.

Função de Hash

- As funções hash possuem 4 importantes propriedades:
  1. Se M for fornecido, o cálculo de $H(M)$ será muito fácil.
  2. Se $H(M)$ for fornecido, será efetivamente impossível encontrar M.
  3. Dado M, ninguém pode encontrar M' tal que $H(M') = H(P)$
  4. Uma mudança na entrada de até mesmo 1 bit produz uma saída muito diferente.

Função de Hash

- Para atender ao critério 3, a função de hash deve ter pelo menos 128 bits, de preferência mais.
- Para atender ao critério 4, o hash deve desfigurar completamente os bits, o que não é diferente dos algoritmos de criptografia de chave simétrica que vimos.
- Calcular o hash de um trecho de texto simples é muito mais rápido que criptografar esse texto simples com um algoritmo de chave pública; portanto, as funções hash podem ser usados para agilizar algoritmos de assinatura digital.
- As funções hash mais amplamente utilizadas são o MD5 e o SHA-1.

MD5

- O MD5 (Message-Digest algorithm 5) é um algoritmo de hash de 128 bits unidirecional desenvolvido pela RSA Data Security, Inc., descrito na RFC 1321, usado por softwares com protocolo ponto-a-ponto (P2P), verificação de integridade e logins. Foi desenvolvido para suceder ao MD4 que tinha alguns problemas de segurança.
- Por ser um algoritmo unidirecional, um hash MD5 não pode ser transformado novamente na password (ou texto) que lhe deu origem. O método de verificação é, então, feito pela comparação das duas hash (uma da base de dados, e a outra da tentativa de login).

MD5

- O MD5 é de domínio público para uso em geral. A partir de uma mensagem de um tamanho qualquer, ele gera um valor hash de 128 bits; com este algoritmo, é computacionalmente impraticável descobrir duas mensagens que gerem o mesmo valor, bem como reproduzir uma mensagem a partir do seu hash.
- O algoritmo MD5 é utilizado como mecanismo de integridade em vários protocolos de padrão Internet.
- Em 2008, Ronald Rivest e outros, publicaram uma nova versão do algoritmo o MD6 com hash de tamanhos 224, 256, 384 ou 512 bytes. O algoritmo MD6 iria participar do concurso para ser o novo algoritmo SHA-3, porém logo depois removeu-o do concurso por considera-lo muito lento, anunciando que os computadores de hoje são muito lentos para usar o MD6.

MD5

- Exemplo de hash MD5

MD5("The quick brown fox jumps over the lazy dog") =
9e107d9d372bb6826bd81d3542a419d6

- Mesmo uma pequena alteração na mensagem vai criar um hash completamente diferente, ex. ao mudar d para c:

MD5("The quick brown fox jumps over the lazy cog") = 1055d3e698d289f2af8663725127bd4b

MD5

- O MD5 já é usado há mais de uma década, e muitos pessoas o têm atacado. Foram encontradas algumas vulnerabilidades, mas certas etapas internas impedem que ele seja rompido. Porém, se as barreiras restantes dentro do MD5 caírem, ele poderá falhar eventualmente. Apesar disso, no momento em que escrevemos, ele ainda resiste [2].

SHA-1

- A outra função importante do sumário de mensagens é o SHA-1 (Secure Hash Algorithm 1), desenvolvido pela NSA e aprovado pelo NIST.
- O SHA-1 gera um valor hash de 160 bits, a partir de um tamanho arbitrário de mensagem.
- O funcionamento interno do SHA-1 é muito parecido com o observado no MD4, indicando que os estudiosos da NSA basearam-se no MD4 e fizeram melhorias em sua segurança.

SHA-1

- Atualmente, não há nenhum ataque de criptoanálise conhecido contra o SHA-1. Mesmo o ataque da força bruta torna-se impraticável, devido ao seu valor hash de 160 bits. Porém, não há provas de que, no futuro, alguém não possa descobrir como quebrar o SHA-1.
- Novas versões de SHA-1 estão em desenvolvimento para hashes de 256, 384 e 512 bits.

SHA-1

- Exemplo de hash SHA-1

SHA1("The quick brown fox jumps over the lazy dog") = 2fd4e1c6 7a2d28fc ed849ee1 bb76e739 1b93eb12

- Mesmo uma pequena mudança na mensagem resultará em um resumo completamente diferente, que concede à função um bom efeito avalanche. Por exemplo, ao alterar d para c:

SHA1("The quick brown fox jumps over the lazy cog") = de9f2c7f d25e1b3a fad3e85a 0bd17d9b 100db4b3

Certificação Digital

- Como visto anteriormente, a criptografia de mensagens com chave assimétricas pode garantir a autenticação e a confidencialidade na troca de mensagens.
- Para que isto aconteça é necessário que:
  - O remetente de uma mensagem criptografada precisa conhecer a chave pública do destinatário.
  - O destinatário de uma mensagem autenticada precisa conhecer a chave pública do remetente.

Certificação Digital

- É necessário que o usuário tenha certeza de que a chave pública que está utilizando é autêntica.
- No caso de um pequeno grupo, a troca das chaves públicas, guardadas de forma segura, pode ser aceitável.
- No entanto, para um grande grupo (Internet, por exemplo) a troca de chave manual é impraticável.
- A solução é o uso de certificados digitais.

Certificação Digital

- **Certificados digitais** são arquivos que contém as informações necessárias a identificação de um indivíduo ou programa, componente, produto, etc, incluindo sua chave pública.
- A principal função de um certificado é vincular uma chave pública ao nome de um protagonista (indivíduo, empresa, etc).
- Os certificados em si não são secretos ou protegidos. Usualmente estão disponíveis em uma base de acesso livre na Internet (diretório X.500).

Serviço de autenticação X.509

- A recomendação X.509 do ITU-T faz parte da série de recomendações X.500 que definem um serviço de diretório.
- O diretório é um servidor ou conjunto de servidores distribuídos que mantém um banco de dados de informações sobre usuários, como endereço de rede, e outras informações sobre o usuário.
- O diretório pode servir com um repositório de certificados de chave pública do tipo visto no módulo anterior.

Serviço de autenticação X.509

- Cada certificado contém a chave pública de um usuário e é assinado com a chave privada de uma autoridade de certificação confiável.
- A estrutura de certificado e os protocolos de autenticação definidos no X.509 são utilizado no SSL/TLS.
- O X.509 é baseado no uso da criptografia de chave pública e assinaturas digitais.
- O padrão não dita o uso de um algoritmo específico, mas recomenda o RSA.

Autoridades Certificadoras

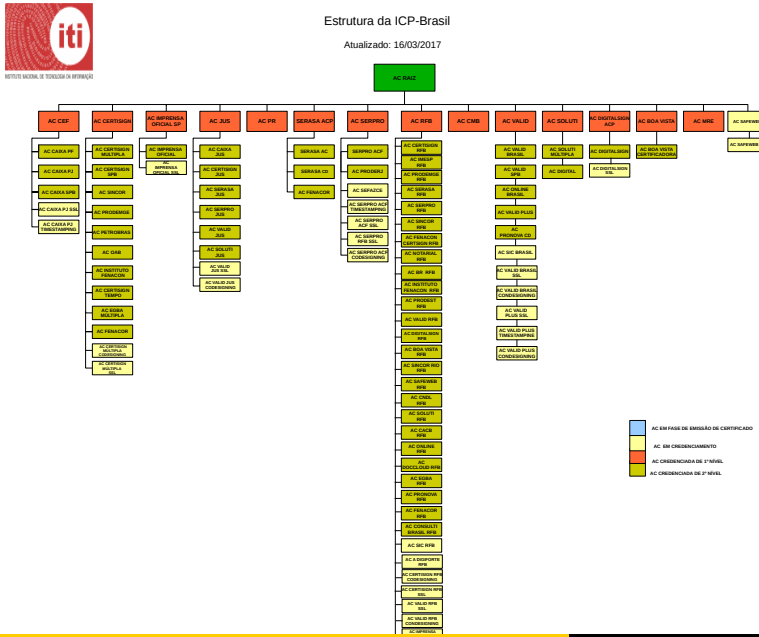
- O núcleo do esquema X.509 é o certificado de chave pública associado a cada usuário.
- Esses certificados de usuário são considerados como sendo criados por alguma autoridade certificadora (CA) confiável e colocados no diretório pela CA ou pelo usuário.
- CA (Certification Authority) são entidades que emitem certificados para possuidores de chaves públicas e privadas (pessoa, dispositivo, servidor).
- O próprio servidor de diretório não é o responsável pela criação das chaves públicas ou pela função de certificação, ele simplesmente oferece um local de fácil acesso para os usuários obterem certificados.

Autoridades Certificadoras

- Funções de uma CA:
  - Gerar, entregar e armazenar a chave privada de forma segura;
  - Distribuir a chave pública;
  - Atualizar o par de chaves;
  - Assinar a chave pública para gerar o certificado. Assinar certificados digitais garantindo sua validade
  - Manter e divulgar uma lista com os certificados revogados (Certificate Revocation List - CRL);
  - CAs podem estar encadeadas em hierarquias de certificação, em que a CA de um nível inferior valida sua assinatura com a assinatura de uma CA mais alta na hierarquia.

Autoridades Certificadoras

- É conveniente que cada nação conte com uma Infra-estrutura de Chaves Públicas (ICP) ou, em inglês, Public Key Infrastructure (PKI), isto é, um conjunto de políticas, técnicas e procedimentos para que a certificação digital tenha amparo legal e forneça benefícios reais à sua população.
- O Brasil conta com a ICP-Brasil para essa finalidade.
- A ICP-Brasil trabalha com uma hierarquia onde a CA-Raiz, isto é, a instituição que gera as chaves das CAs e que regulamenta as atividades de cada uma, é o Instituto Nacional de Tecnologia da Informação (ITI).


Estrutura da ICP-Brasil
Atualizado: 16/03/2017
AC RAIZ
AC CEF
AC CAIXA PF
AC CAIXA PJ
AC CAIXA SPB
AC CAIXA PJ SSL
AC CAIXA PJ TIMESTAMPING
AC CERTISIGN
AC CERTISIGN MULTIPLA
AC CERTISIGN SPB
AC SINCOR
AC PRODEMGE
AC PETROBRAS
AC GIAB
AC INSTITUTO FENACON
AC CERTISIGN TEMPO
AC IEMA MULTIPLA
AC FENACON
AC CERTISIGN MULTIPLA SSL
AC CERTISIGN MULTIPLA SSL
AC IMPRENSA OFICIAL SP
AC IMPRENSA OFICIAL
AC IMPRENSA OFICIAL SSL
AC JUS
AC CAIXA JUS
AC CERTISIGN JUS
AC SERASA JUS
AC SERPRO JUS
AC VALID JUS
AC SOLUTI JUS
AC VALID JUS SSL
AC VALID JUS CODESIGNING
AC PR
SERASA ACP
SERASA AC
SERASA CD
AC FENACON
AC SERPRO
SERPRO ACP
AC PRODERJ
AC SEFAZCS
AC SERPRO ACP TIMESTAMPING
AC SERPRO ACP SSL
AC SERPRO RFB SSL
AC SERPRO ACP CODESIGNING
AC RFB
AC CERTISIGN RFB
AC IRIDP RFB
AC PRODEMGE RFB
AC SERASA RFB
AC SERPRO RFB
AC SINCOR RFB
AC FENACON CERTISIGN RFB
AC SOTEMAR RFB
AC BR RFB
AC INSTITUTO FENACON RFB
AC PRODEST RFB
AC VALID RFB
AC SIGNTHEDOCS RFB
AC BOA VISTA RFB
AC SINCOR RIO RFB
AC SAFEWEB RFB
AC CNDL RFB
AC SOLUTI RFB
AC CACB RFB
AC ONLINE RFB
AC SOCOLOUD RFB
AC IEMA RFB
AC PRONOEMA RFB
AC FENACON RFB
AC CONSULTI BRASIL RFB
AC SIC RFB
AC A DOCPORTE RFB
AC CERTISIGN RFB CODESIGNING
AC CERTISIGN RFB SSL
AC VALID RFB SSL
AC VALID RFB CODESIGNING
AC IMPRENSA
AC CMB
AC VALID
AC VALID BRASIL
AC VALID SPB
AC ONLINE BRASIL
AC VALID PLUS
AC PRONOEMA CD
AC SIC BRASIL
AC VALID BRASIL SSL
AC VALID BRASIL CODESIGNING
AC VALID PLUS SSL
AC VALID PLUS TIMESTAMPING
AC VALID PLUS CODESIGNING
AC SOLUTI
AC SOLUTI MULTIPLA
AC DIGITAL
AC DIGITALSIGN ACP
AC DIGITALSIGN
AC DIGITALSIGN SSL
AC BOA VISTA
AC BOA VISTA CERTIFICADORA
AC MRE
AC SAFEWEB
AC SAFEWEB CD
AC EM FASE DE EMISSÃO DE CERTIFICADO
AC EM CREDENCIAMENTO
AC CREDENCIADA DE 1º NÍVEL
AC CREDENCIADA DE 2º NÍVEL

Autoridades Certificadoras

- A ICP-Brasil oferece duas categorias de certificados digitais: A e S, sendo que cada uma se divide em quatro tipos: A1, A2, A3 e A4; S1, S2, S3 e S4. A categoria A é direcionada para fins de identificação e autenticação, enquanto que o tipo S é direcionado a atividades sigilosas. Vejas as características que tornam as versões de ambas as categorias diferentes entre si:
  - **A1 e S1**: geração das chaves é feita por software; chaves de tamanho mínimo de 1024 bits; armazenamento em dispositivo de armazenamento (como um HD); validade máxima de um ano;
  - **A2 e S2**: geração das chaves é feita por software; chaves de tamanho mínimo de 1024 bits; armazenamento em cartão inteligente (com chip) ou token (dispositivo semelhante a um pendrive); validade máxima de dois anos;

Autoridades Certificadoras

- - **A3 e S3**: geração das chaves é feita por hardware; chaves de tamanho mínimo de 1024 bits; armazenamento em cartão inteligente ou token; validade máxima de três anos;
  - **A4 e S4**: geração das chaves é feita por hardware; chaves de tamanho mínimo de 2048 bits; armazenamento em cartão inteligente ou token; validade máxima de três anos.
- Os certificados A1 e A3 são os mais utilizados, sendo que o primeiro é geralmente armazenado no computador do solicitante, enquanto que o segundo é guardado em cartões inteligentes (smartcards) ou tokens protegidos por senha.

Certificados

- O formato geral de um certificado inclui os seguintes elementos:
- **Versão**: Indica a versão do formato do certificado.
- **Número de série**: Valor inteiro, exclusivo dentro da CA emitente, que é associado sem ambiguidades a esse certificado.
- **Identificador do algoritmo de assinatura**: O algoritmo usado para assinar o certificado.

Certificados

- **Nome do emissor**: O nome X.500 da CA que criou e assinou o certificado.
- **Período de validade**: Consiste em duas datas, a primeira e a última em que o certificado é válido.
- **Nome do titular**: O nome do usuário a quem o certificado se refere, ou seja, o certificado certifica a chave pública do titular que mantém a chave privada correspondente.

Certificados

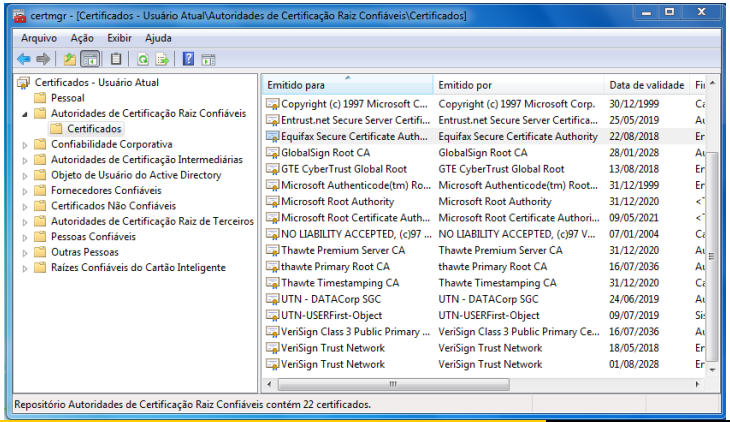
- **Informação de chave pública do titular**: A chave pública do titular, mais um identificador do algoritmo para o qual a chave deve ser usada, juntamente com quaisquer parâmetros associados.
- **Identificador exclusivo do emissor**: Um campo de sequência de bits opcional usado para identificar exclusivamente a CA emissora caso o nome X.500 tenha sido reutilizado para entidades diferentes.
- **Identificador exclusivo do titular**: Um campo de sequência de bits opcional, usado para identificar exclusivamente o titular caso o nome X.500 tenha sido reutilizado para diferentes entidades.

Certificados

- **Extensões**: Um conjunto de um ou mais campos de extensão.
- **Assinatura**: Abrange todos os outros campos do certificado; contém o hash dos outros campos, criptografados com a chave privada da CA. Este campo inclui o identificador do algoritmo de assinatura.

Certificados

- No Windows 7 você pode ver os certificados do seu computador usando o Gerenciador de Certificados.
- Para isto, clique no botão iniciar e digite **certmgr.msc** na caixa de pesquisa e pressione Enter.



Certificados

- Tipos mais comuns de certificados e suas funções:
- **Sistema de Arquivos com Criptografia**: Criptografar e descriptografar documentos.
- **Autenticação do servidor**: Verificar a identidade de um servidor para os computadores conectados a ele.
- **Autenticação de cliente**: Verificar a identidade de um computador para um servidor ao qual esteja conectado.

Certificados

- **Email seguro**: Criptografar e assinar digitalmente o email.
- **Assinatura do código**: Verificar o editor de um programa. Por exemplo, se você baixa um programa do ActiveX, a sua assinatura digital verificará se ele foi publicado pela organização listada como editora.
- **Recuperação de arquivos**: Recuperar arquivos criptografados se o certificado EFS for acidentalmente excluído ou danificado.

Certificados

- Obtenção de um Certificado:
  - Cliente gera um par de chaves pública e privada (por exemplo, usando RSA);
  - Envia-se um pedido de certificado para a Autoridade de Registro;
  - AR (Autoridade de Registro Regional) faz a prova de existência do requisitante e retransmite o pedido para a CA;
  - CA assina e envia o certificado;
  - Usuário instala seu certificado;
  - Usuário divulga o certificado;

Certificados

- Política de Certificação:
  - A autoridade de registro (AR), tendo a delegação de uma CA para tal, faz uma investigação no solicitante e determina:
  - Se o pedido deve ser atendido;
  - Quais as características que deve ter;

Certificados

Figura 21: Processo de Registro de Certificados

Referências

[1] W. Stallings, *Criptografia e segurança de redes*. Editora Pearson Prentice Hall, 2008.

[2] A. S. TANENBAUM, *Redes de computadores*. Editora Campus, 2003.