
Criptografia

Dulce Domingos

acetatos baseados nos livros da bibliografia da disciplina e nos acetatos da disciplina de segurança de anos lectivos anteriores (de P.Veríssimo, A.Bessani, M.Calha, H.Reiser)

1

Sumário

- ❖ Introdução
- ❖ Criptografia e criptanálise
- ❖ Evolução da tecnologia de cifra
- ❖ Tipos de cifra
 - Cifras de transposição
 - Cifras de substituição
 - Monoalfabéticas
 - Polialfabéticas
- ❖ Conceitos teóricos
- ❖ Cifras modernas
 - Cifras simétricas
 - Modos de operação
 - Cifras por blocos
 - Cifras Contínuas
 - Cifras assimétricas
 - Cifras híbridas
 - Funções de síntese
 - Autenticação
 - MACs
 - Assinaturas digitais

2

Introdução

❖ Criptografia

- Kryptós (oculto) + graph (escrever)
- ☹ o uso de criptografia revela-a
 - Fornece indícios de que a informação é sensível
 - Pode ser ilegal

❖ Esteganografia

- Conteúdo sensível é ocultado dentro de outro conteúdo
- Exemplos:
 - Escrita com tinta invisível
 - Ocultar conteúdos dentro de imagens nos bits menos significativos de cada pixel

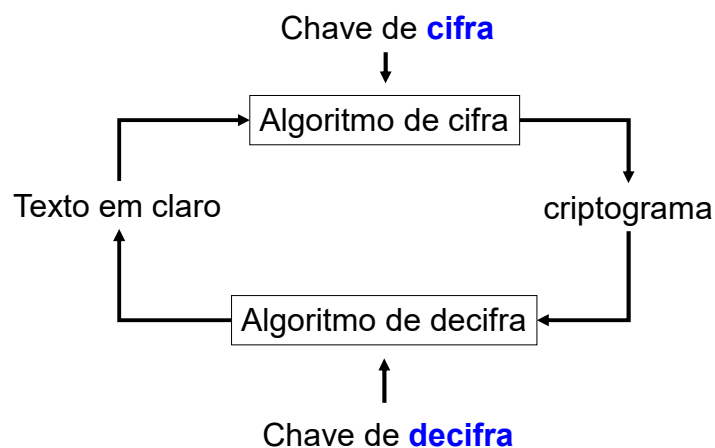
❖ Criptanálise

- Arte ou ciência de violar informação criptografada ou sistemas criptográficos

❖ Criptologia

- Estudo de criptografia e criptanálise

Criptografia



Criptanálise

❖ Objectivos:

- Obter texto original
- Obter chave de cifra
- Obter algoritmo de cifra

❖ Algumas técnicas

- Ataques usando apenas o criptograma (ciphertext-only attacks)
- Ataques com conhecimento de parte do texto original (known-plaintext attacks) + criptograma
- Ataques com texto original escolhido (chosen-plaintext attacks)
 - Ataques com texto original escolhido de forma adaptativa (adaptive chosen-plaintext attacks)
- Ataques com criptogramas escolhidos (chosen-ciphertext attacks)
- Ataques de aniversário (birthday attacks)

Evolução da tecnologia de cifra

❖ Primeiras cifras

➤ Espartanos

- O pergaminho só poderia ser lido se fosse enrolado num bastão com o mesmo diâmetro



➤ Cifra de César

- Substitui uma letra pela k ésima letra seguinte no alfabeto, MOD 26
- Mecanismo:
 - $E_k(m) = (m + k) \bmod 26$ (função de cifra)
 - $D_k(c) = (26 + c - k) \bmod 26$ (função para decifrar)
- Exemplo:
 - $k = 2$
 - Texto em claro: seguranca
 - Texto cifrado: ugixtcpec

Evolução da tecnologia de cifra

❖ Cifras mecânicas

- Máquinas usadas para cifrar e decifrar mensagens
- Antes dos computadores modernos, eram muito usados
 - Na Segunda Guerra Mundial:
 - Enigma (Alemanha),
 - Hagelin (Aliados),
 - Purple (Japão)
- Formas de concretização variam (ex. uso de vários cilindros), mas em geral baseavam-se no uso de várias cifras de substituição extremamente complexas



Sumário

- ❖ Introdução
- ❖ Criptografia e criptanálise
- ❖ Evolução da tecnologia de cifra
- ❖ **Tipos de cifra**
 - **Cifras de transposição**
 - **Cifras de substituição**
 - Monoalfabéticas
 - Polialfabéticas
- ❖ Conceitos teóricos
- ❖ Cifras modernas
 - Cifras simétricas
 - Modos de operação
 - Cifras por blocos
 - Cifras Contínuas
 - Cifras assimétricas
 - Cifras híbridas
 - Funções de síntese
 - Autenticação
 - MACs
 - Assinaturas digitais

Tipos de cifras: cifras de transposição

❖ **Baralham** os caracteres do texto original

❖ Exemplos:

- Permutações fixas em blocos com um número constante de caracteres
 - Permutação 45231
 - Criptograma: raifc
 - Texto original ?
- Blocos verticais de dimensão fixa
 - Blocos verticais de 5 caracteres:
 - eaéo ...
 - loms
 - esqo
 - saun
 - nbeh

Tipos de cifras: cifras de substituição

❖ **Substituem** os caracteres do alfabeto usado no texto original por caracteres de um alfabeto de substituição

❖ **Monoalfabéticas**

- Usam apenas um alfabeto de substituição
- Um carácter do alfabeto original é substituído sempre pelo mesmo carácter
- Exemplo: cifra de César
- Criptanálise
 - Força bruta
 - Ataques com texto original escolhido
 - Padrões estatísticos dos caracteres usados no texto original
- **Polialfabéticas**
 - Aplicação sucessiva e cíclica de várias cifras monoalfabéticas
 - Cifra de Vigenère

Cifra Vigenère

- ❖ Chave: conjunto de caracteres
- ❖ Mecanismo:
 - Repete-se a chave em sequência até que a chave seja do tamanho do texto a ser cifrado
 - Para cada letra, é feita uma substituição:
 - $B + C = D$ ($A=0, B=1, C=2, \dots$)
 - Se a chave tem uma letra apenas, temos uma cifra monoalfabética
- ❖ Exemplo de uso:
 - $k = \text{poema}$

Chave	poemapoemapoe
Texto em claro	elesnaosabemq
Texto cifrado	tzienpcwmbtau
- ❖ Criptanálise:
 - Determinar a dimensão da chave -> criptanálise de N cifras monoalfabéticas
 - Técnicas estatísticas para determinar N: teste de Kasiski e índice de coincidência

© 2024 DI-FCUL Reprodução proibida sem autorização prévia.

11

11

Conceitos teóricos

❖ Cifra perfeita

- Uma cifra diz-se **perfeita** quando, dado um criptograma c , a probabilidade de ele corresponder a um dado texto original m e de ter sido gerado com uma dada chave k é igual à probabilidade de ocorrência do texto m

Para cada texto em claro há sempre uma chave que geraria o criptograma

⇒ O cardinal do espaço de chaves tem de ser igual ou superior ao cardinal do espaço de textos em claro

⇒ Difícil

➢ Cifra de Vernam

- O comprimento da chave é maior ou igual do que o do texto a cifrar

➢ Dificuldades

- Para cada texto tem de ser usada uma chave diferente
- O comprimento das chaves tem de ser igual ou superior ao dos textos
- As chaves não são memorizáveis
- Pré-distribuição de chaves de grande dimensão
- Não faz sentido usar para cifrar dados armazenados

© 2024 DI-FCUL Reprodução proibida sem autorização prévia.

12

12

Conceitos teóricos

❖ Cifras seguras **na prática**

- Uma cifra diz-se segura se cumprir o objectivo para que é usada
 - As **vulnerabilidades** da cifra, mesmo quando usada de forma correcta, não permitem a sua criptanálise em **tempo útil** e admitindo um **investimento** tendo em conta a relação **custo-benefício**
 - A cifra é usada de forma correcta, sem aumentar as suas vulnerabilidades intrínsecas

❖ Critérios para avaliar a qualidade das cifras [Claude Shannon, 1949]:

1. Quantidade de secretismo oferecida
 - Tempo mínimo de segurança do criptograma
 - Confusão e difusão (ver acetato seguinte)
2. Dimensão das chaves
 - Cifra de Vernam – gestão de chaves ☹
3. Simplicidade de realização e uso
4. Propagação de erros
5. Dimensão do criptograma
 - O tamanho do texto cifrado não deve ser maior que o do texto em claro – custo de armazenamento ou transmissão

© 2024 DI-FCUL Reprodução proibida sem autorização prévia.

13

13

Conceitos teóricos

❖ Cifras com boa quantidade de secretismo [Claude Shannon, 1949]:

- Confusão
 - Relação entre texto em claro, chave e criptograma deve ser o mais complexa possível
 - É difícil descobrir parte do texto em claro mesmo conhecendo outras
 - É difícil descobrir parte ou toda a chave usada para produzir um criptograma
- Difusão
 - Cada bit de informação do texto original deve influenciar vários bits do criptograma
 - Pequena alteração no texto original implica grandes alterações no criptograma

© 2024 DI-FCUL Reprodução proibida sem autorização prévia.

14

14

Conceitos teóricos

❖ Boas práticas:

- Criptanalista conhece o algoritmo de cifra e as suas fragilidades
 - Segurança baseia-se no desconhecimento da chave
 - Tem de ser baseado em matemática sólida
 - Tem de ter sido analisado por vários especialistas
 - Tem de ter passado no teste do tempo
- Criptanalista pode capturar todos os criptogramas
- Criptanalista conhece partes do texto original

Sumário

- ❖ Introdução
- ❖ Criptografia e criptanálise
- ❖ Evolução da tecnologia de cifra
- ❖ Tipos de cifra
 - Cifras de transposição
 - Cifras de substituição
 - Monoalfabéticas
 - Polialfabéticas
- ❖ Conceitos teóricos
- ❖ **Cifras modernas**
 - **Cifras simétricas**
 - Modos de operação
 - Cifras por blocos
 - Cifras Contínuas
 - Cifras assimétricas
 - Cifras híbridas
 - Funções de síntese
 - Autenticação
 - MACs
 - Assinaturas digitais

Cifras modernas

- ❖ Tipo de chave
 - Cifras simétricas
 - Chave secreta
 - Confidencialidade
 - Eficientes
 - N utilizadores e distribuição segura de chaves ☹
 - Cifras assimétricas
 - Par de chaves
 - Confidencialidade, autenticidade
 - Não eficientes
 - N utilizadores ☺
 - Distribuição de chaves públicas
 - Cifra híbrida
 - Cifra com chave simétrica
 - Distribuição de chave simétrica com cifras assimétricas
 - **Porquê?**
- ❖ Modo de operação
 - Cifras por blocos
 - Cifras contínuas

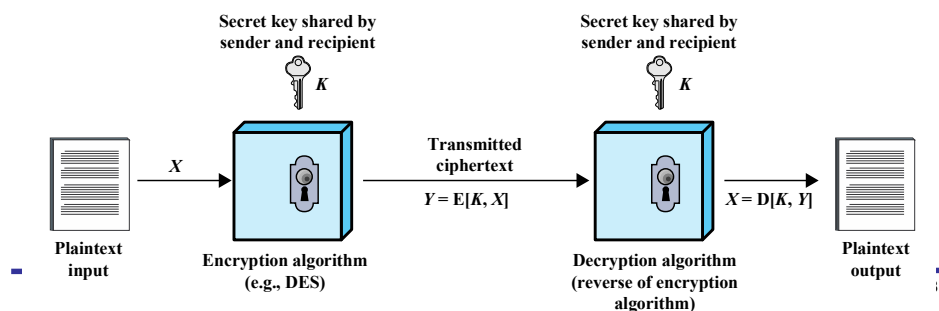
© 2024 DI-FCUL Reprodução proibida sem autorização prévia.

17

17

Criptografia Simétrica

- ❖ Também denominada de chave partilhada ou de chave secreta
 - chave de cifrar e decifrar iguais
 - bastante rápida
- ❖ Propriedade fundamental: $D(K, E(K, m)) = m$
- ❖ Cifras simétricas por blocos
 - *Data Encryption Standard* (DES) (1977);
 - *Triple-DES*;
 - *International Data Encryption Algorithm* (IDEA);
 - *Advanced Encryption Standard* (AES) (2000)
- ❖ Cifras simétricas contínuas



18

Ataques a cifras simétricas

❖ Criptanálise

- Algoritmo
 - Deduzir texto em claro específico
 - Deduzir a chave
- Conhecimento de características do texto em claro
- Exemplos de pares texto em claro/criptograma
- Se a chave for comprometida
 - Compromete mensagens passadas e futuras cifradas com a chave

❖ Ataques de força bruta

- Tenta todas as chaves
- Em média, tem de tentar metade das chaves

Criptografia Simétrica – ponto de situação

- ❖ Em 1998 foi publicado o primeiro projecto de um sistema computacional e do software associado capaz de quebrar qualquer texto cifrado pelo **DES** em poucos dias
- ❖ Muitos desafios foram lançados para quebrar mensagens específicas cifradas com o DES
 - A maioria foi resolvida rapidamente usando **computadores paralelos distribuídos (que são/eram muito caros!.....)**
- ❖ Por outro lado, o **IDEA** tem resistido à passagem do tempo...
- ❖ O NIST seleccionou o algoritmo **Rijndael** como seu *Advanced Encryption Standard* (AES), o sucessor do DES
 - Desenhado para resistir a ataques bem sucedidos ao DES

DES - Data Encryption Standard

- ❖ Criado em **1977** pelo governo americano com o objectivo de ser um algoritmo normalizado para cifrar/decifrar dados sensíveis, mas não classificados.
- ❖ Problemas:
 - Chaves pequenas (**56 bits**)
 - Alguns passos do algoritmo não se sabe o porquê de existirem
 - Muitos algoritmos para “quebrá-lo” já foram desenvolvidos
 - Em **1998** foi apresentado um sistema que quebrava o DES em quatro dias
 - Ficou claro que ele já não era uma solução de segurança viável
- ❖ É um marco histórico

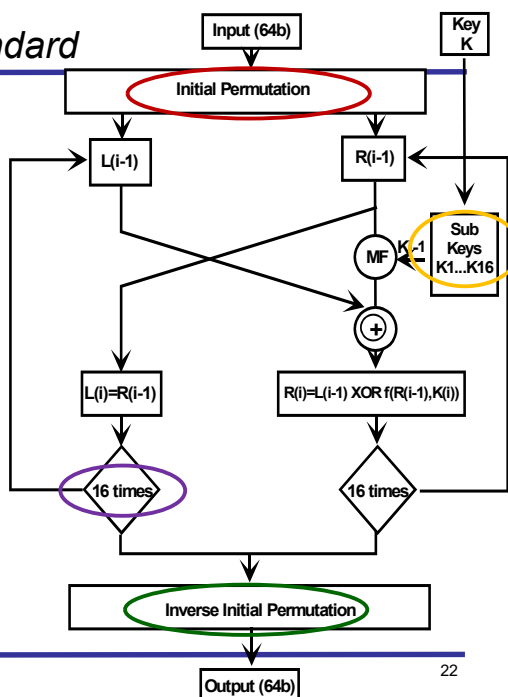
© 2024 DI-FCUL Reprodução proibida sem autorização prévia.

21

21

DES - Data Encryption Standard

- ❖ PRINCÍPIO:
 - cifrar em blocos de 64 bits de texto em claro, com chave K de 56 bits
 - produz 64 bits de texto cifrado
- ❖ OPERAÇÃO DE CIFRAR:
 - **permutação inicial P dos 64bits;**
 - **Geração de subchaves**
 - **16 iterações de transformação;**
 - **permutação final, inversa de P**
 - **Princípio de difusão e confusão**
- ❖ TRANSFORMAÇÃO:
 - Isto é uma Estrutura de Feistel
 - $L_i = R_{i-1};$
 - $R_i = L_i \text{ XOR } f(R_{i-1}, K_i)$
 - Sendo:
 - f , função que faz substituição e permutação;
 - K_i , chave temporária de 48bits usada na iteração i (produzida a partir de K e i)
- ❖ OPERAÇÃO DE DECIFRAR:
 - mesmas operações, com as chaves em ordem inversa



© 2024 DI-FCUL Reprodução proibida sem autorização prévia.

22

22

DES – alguns detalhes – Permutação Inicial

❖ 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

❖ O bit 58 do texto em claro é colocado na posição 1

❖ O bit 50 do texto em claro é colocado na posição 2

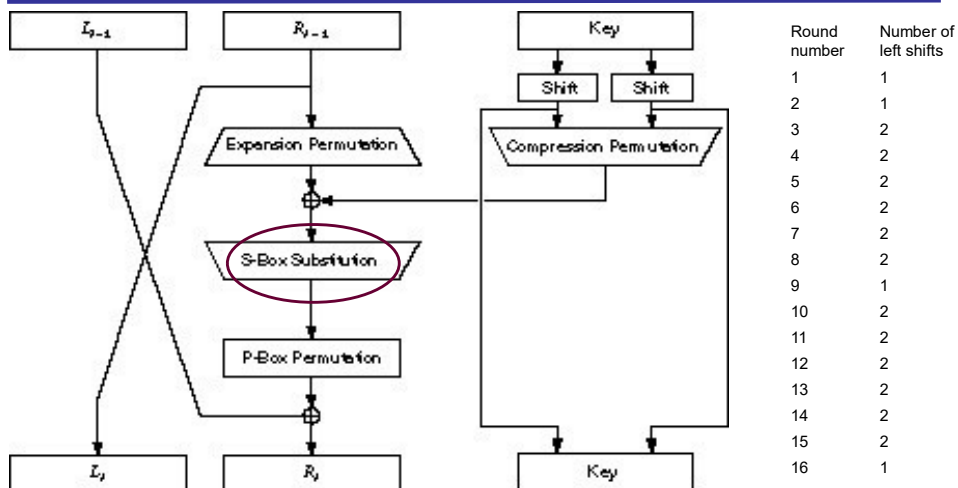
❖ ...

© 2024 DI-FCU

http://en.wikipedia.org/wiki/DES_supplementary_material

23

DES – alguns detalhes – geração de subchaves

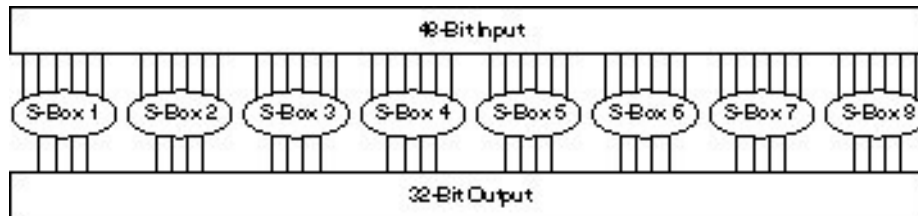


© 2024 DI-FCUL Reprodução proibida sem autorização prévia.

24

24

DES – alguns detalhes – SBOX substitution



S-boxes

S_1

	x0000x	x0001x	x0010x	x0011x	x0100x	x0101x	x0110x	x0111x	x1000x	x1001x	x1010x	x1011x	x1100x	x1101x	x1110x	x1111x
0yyyyy0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0yyyyy1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
1yyyyy0	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
1yyyyy1	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Exemplo: $110011_2 \rightarrow 11_{10} = 1011_2$

© 2024 DI-FCUL Reprodução proibida sem autorização prévia.

25

25

DES - *Data Encryption Standard*

❖ Problemas

- Chaves pequenas (56 bits)
- Não se conhece a justificação para alguns passos do algoritmo
- 1998: é possível comprometer chaves DES em 4 dias
 - Força bruta
- Chaves fracas, semi-fracas e potencialmente fracas
 - Pouco ou nada alteram o texto original

❖ Solução: Variantes do DES

- Utilização de várias cifras
 - Dual DES
 - Triple DES

© 2024 DI-FCUL Reprodução proibida sem autorização prévia.

26

26

Dual DES

❖ Dual DES

- usa 2 chaves: K_1 e K_2 :
 - “chaves” com $2n$ bits
 - Cifra: $C = E(K_2, E(K_1, P))$
 - Decifra: $P = D(K_2, D(K_1, C))$
 - Problema: pode ser atacado com 2^{n+1} tentativas (vs as previstas: 2^{2n})
 - Como?
 1. força bruta calcula $E(K, P)$ para todas as $K \rightarrow 2^n$
 2. força bruta calcula $D(K, C)$e compara com os valores calculados em 1
- **meet-in-the-middle-attack**

© 2024 DI-FCUL Reprodução proibida sem autorização prévia.

27

27

Triple DES

❖ Triple DES

- Usa DES 3 vezes com 2 ou 3 chaves
 - Normalizado em 1985: ANSI standard X9.17
 - 2 chaves K_1 e K_2 :
 - » Cifra: $C = E(K_1, D(K_2, E(K_1, P)))$
 - » Decifra: $P = D(K_1, E(K_2, D(K_1, C)))$
 - 3 chaves:
 - » Cifra: $C = E(K_3, D(K_2, E(K_1, P)))$
 - » Decifra: $P = D(K_1, E(K_2, D(K_3, C)))$
 - » Requer $O(2^{2n})$ cifras e $O(2^n)$ de memória com chaves de 56 bits
- Ataque muito difícil

© 2024 DI-FCUL Reprodução proibida sem autorização prévia.

28

28

IDEA - International Data Encryption Algorithm

- ❖ Proposto em **1991** por investigadores da ETH Zurich
- ❖ Mais robusto que o DES
- ❖ usa chaves de **128 bits**:
 - ataque força bruta requer 10^{38} cifrações
 - com chip de 10^9 cifr/seg, requer 10^{13} anos
- ❖ Tão rápido quanto o DES:
 - Concretização em software: 386@33MHz faz 880Kbps
 - Concretização em hardware (chip da ETH Zurich) faz 177Mbps @25MHz
 - (agora escalem isso proporcionalmente aos processadores de hoje)
- ❖ Usado no PGP (*Pretty Good Privacy*)
- ❖ Nunca foi normalizado

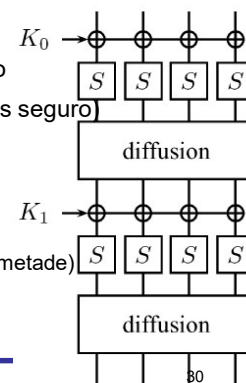
© 2024 DI-FCUL Reprodução proibida sem autorização prévia.

29

29

AES - Advanced Encryption Standard

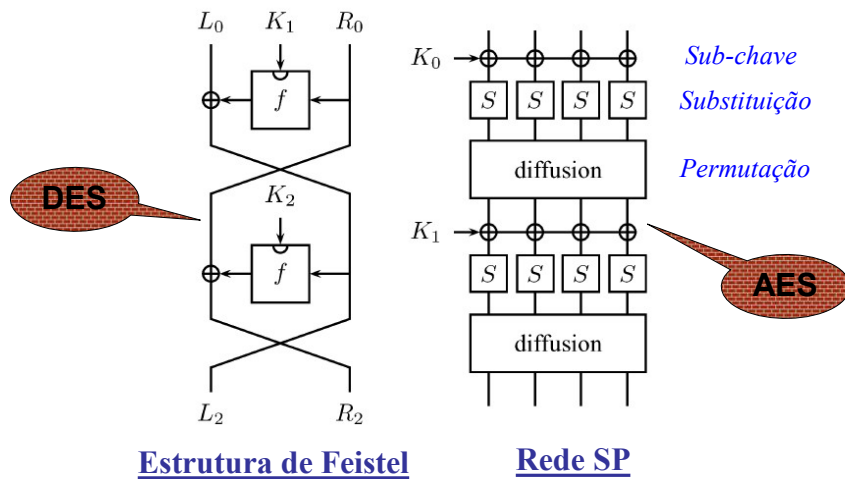
- ❖ Histórico:
 - Novo padrão do NIST para substituir o DES
 - Processo de seleção público (iniciado em 1997) onde se escolheu um algoritmo de entre vários candidatos
 - O escolhido foi o Rijndael – Novembro de 2001
- ❖ Princípios:
 - Recebe como entrada **blocos de 128 bits** de texto em claro
 - As chaves podem ter **128, 192, 256 bits** (quanto maior, mais seguro)
 - Produz blocos de 128 bits de texto cifrado
 - Funciona iterativamente
 - Cada bloco é dividido em 4 grupos de 4 bytes
 - Um bloco inteiro é modificado em cada iteração (no DES é só metade)
 - Rápida e eficiente em CPUs pequenos e grandes



© 2024 DI-FCUL Reprodução proibida sem autorização prévia.

30

AES - Advanced Encryption Standard



© 2024 DI-FCUL Reprodução proibida sem autorização prévia.

31

31

Cifras simétricas

Key size (bits)	Cipher	Number of Alternative Keys	Time Required at 10^9 decryptions/s	Time Required at 10^{13} decryptions/s
56	DES	$2^{56} \approx 7.2 \times 10^{16}$	2^{55} ns = 1.125 years	1 hour
128	AES	$2^{128} \approx 3.4 \times 10^{38}$	2^{127} ns = 5.3×10^{21} years	5.3×10^{17} years
168	Triple DES	$2^{168} \approx 3.7 \times 10^{50}$	2^{167} ns = 5.8×10^{33} years	5.8×10^{29} years
192	AES	$2^{192} \approx 6.3 \times 10^{57}$	2^{191} ns = 9.8×10^{40} years	9.8×10^{36} years
256	AES	$2^{256} \approx 1.2 \times 10^{77}$	2^{255} ns = 1.8×10^{60} years	1.8×10^{56} years

© 2024 DI-FCUL Reprodução proibida sem autorização prévia.

32

32

Sumário

- ❖ Introdução
- ❖ Criptografia e criptanálise
- ❖ Evolução da tecnologia de cifra
- ❖ Tipos de cifra
 - Cifras de transposição
 - Cifras de substituição
 - Monoalfabéticas
 - Polialfabéticas
- ❖ Conceitos teóricos
- ❖ Cifras modernas
 - Cifras simétricas
 - **Modos de operação**
 - Cifras por blocos
 - Cifras Contínuas
 - Cifras assimétricas
 - Cifras híbridas
 - Funções de síntese
 - Autenticação
 - MACs
 - Assinaturas digitais

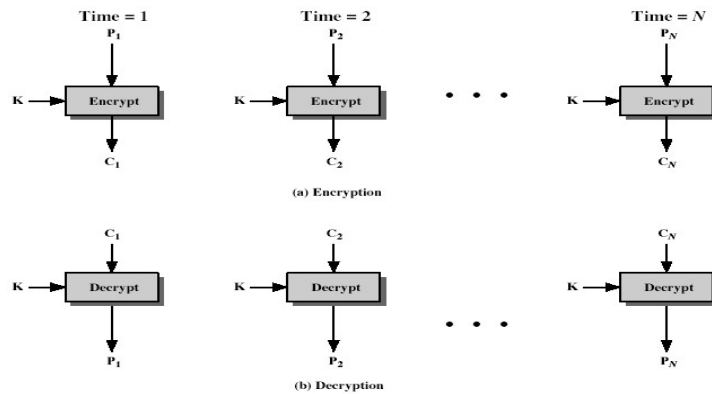
Modos de operação

- ❖ **Cifra por blocos**
 - Processa o texto em claro por blocos, um bloco em cada iteração
 - Produz um bloco por cada bloco de entrada
 - Pode reusar chaves
- ❖ Cifras contínuas
 - Processa o texto em claro de forma contínua
 - Normalmente mais rápidas

Modos de Cifra por Blocos

❖ ECB - *Electronic CodeBook*

- cifra por blocos independentes
- Fraquezas
 - Reprodução de padrões de texto original – dois blocos iguais produzem o mesmo criptograma
 - Vulnerável a ataques de reordenação ou *replay*



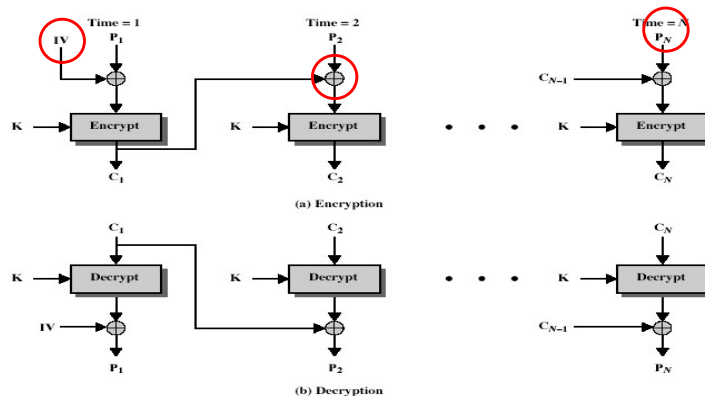
© 2024 DI-FCUL Reprodução

35

Modos de Cifra por Blocos

❖ CBC - *Cypher Block Chaining*

- O texto em claro é "XOR" com o texto cifrado do bloco anterior antes de ser cifrado
- Reduz risco de replicação de padrões
- Initialization Vector (IV): usado no 1º bloco (necessário para decifrar -> ver API do Java)
- *Padding*: bits para compor blocos inteiros do tamanho requerido pelo algoritmo



© 2024 DI-FCUL Reprodução prc

36

Modos de cifra por bloco - padding

❖ Modos de cifra: ECB e CBC

- Iso10126
 - Exemplo: Blocos de 8 bytes e texto em claro 0x616263
 - Texto com padding 0x616263???????05
- PKCS7
 - Exemplo: Blocos de 8 bytes e texto em claro 0x616263
 - Texto com padding 0x6162630505050505
- Bit padding
 - 1011 1001 1101 0100 0010 0111 1 0000 0000

Modos de operação

❖ Cifra por blocos

- Processa o texto em claro por blocos, um bloco em cada iteração
- Produz um bloco por cada bloco de entrada
- Pode reusar chaves

❖ Cifras contínuas

- Processa o texto em claro de forma contínua
- Normalmente mais rápidas

Cifras contínuas

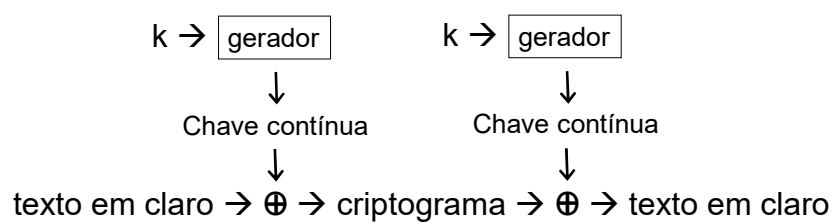
- ❖ XOR (Exclusive OR)
 - $(A \text{ XOR } B) \text{ XOR } A = B$
- ❖ Operação:
 - Geração da chave em tempo real por uma black box
 - **Chaves usadas apenas uma vez**
 - Atenção- ver WEP, fraquezas de RC4 (usado em SSL e WAP)
 - Distribuição?
 - A chave contínua é gerada em simultâneo em ambos os pontos – sincronização
 - Black boxes são parametrizadas por uma master key
 - Cifra/decifra
 - Processa um bit/byte de cada vez, continuamente

© 2024 DI-FCUL Reprodução proibida sem autorização prévia.

39

39

Cifras contínuas



- ❖ Usam chaves contínuas
- ❖ Usam princípio da confusão (não da difusão)
- ❖ As chaves não devem ser **reusadas**
 - Porquê ?
- ❖ O período de uma chave contínua deve ser o mais longo possível
 - Porquê ?
- ❖ A sequência de bits de uma chave contínua deve aparentar ser aleatória

© 2024 DI-FCUL Reprodução proibida sem autorização prévia.

40

40

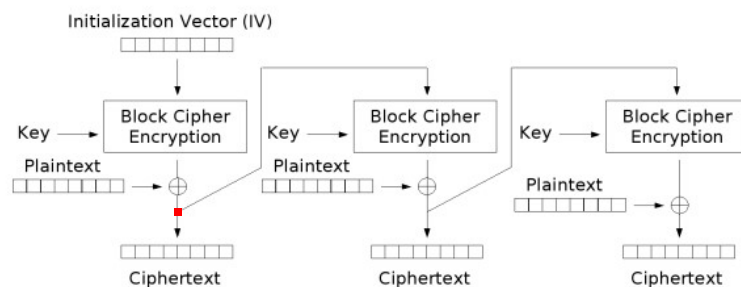
Cifras contínuas

❖ CFB - Cypher Feedback

➤ Transforma **cifra por blocos em cifra contínua**

➤ Vantagens sobre o CBC:

- a cifra de bloco só é utilizada na direcção de cifrar (independentemente de a operação ser cifrar ou decifrar) o que simplifica a sua implementação, e
- a mensagem não necessita de ser "padded" para um múltiplo do tamanho do bloco porque o algoritmo trabalha com qualquer quantidade de bytes



© 2024 DI-FCUL Reproduç

Cipher Feedback (CFB) mode encryption

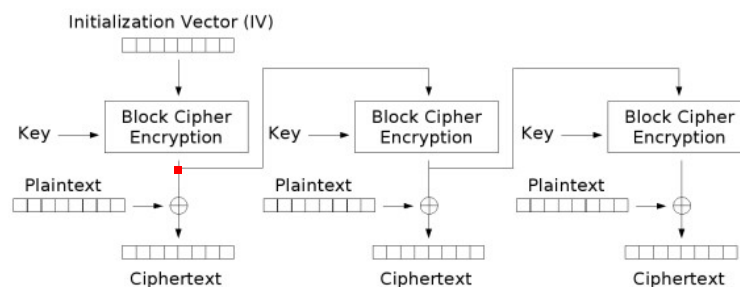
41

Cifra contínuas

❖ OFB – Output Feedback

➤ Mantém as vantagens do CFB e ainda acrescenta outra

- a mensagem não é utilizada para iniciar o bloco seguinte, o que implica que as operações de cifra de bloco podem ser feitas **antecipadamente** permitindo que o último passo seja realizado em paralelo assim que o texto (mensagem ou mensagem cifrada) estiver disponível



© 2024 DI-FCUL Reproduç

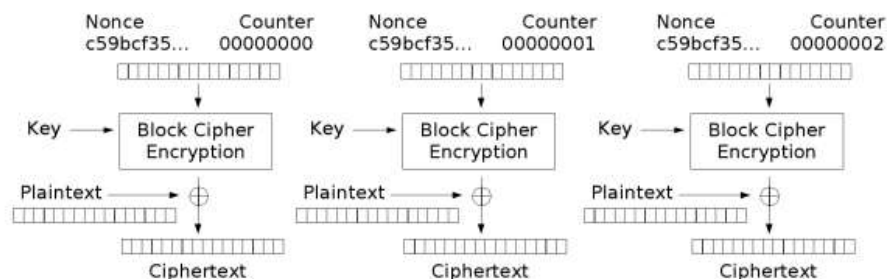
Output Feedback (OFB) mode encryption

42

Cifra contínuas

❖ CTR (counter)

- Apresentado por Diffie e Hellman em 1979
- Modo de cifra padrão para o AES
- Nonce+contador devem ser diferentes em cada operação de cifra
 - Caso seja usada a mesma chave e o mesmo nonce+contador para cifrar dois conteúdos diferentes, eles serão cifrados com duas chaves contínuas iguais ☹



© 202

Counter (CTR) mode encryption

43

Cifras simétricas contínuas

❖ Requisitos de robustez:

- Secretismo, aleatoriedade e uso único da **chave de fluxo**
- A chave de fluxo precisa ser distribuída nas duas pontas do canal

❖ Uso em sistemas reais:

- Em comunicação, a *chave de fluxo* é uma **sequência pseudo-aleatória** produzida em tempo-real à velocidade do fluxo de texto, por uma caixa-preta;
- A chave de fluxo é gerada nos dois extremos em simultâneo (as duas caixas pretas são sincronizadas)
- A chave de fluxo é parametrizada por uma **chave mestra**
- É susceptível a erros de bits, que podem dessincronizar o fluxo
- Exemplo de algoritmo: RC4 (usado no SSL e no WAP)

© 2024 DI-FCUL Reprodução proibida sem autorização prévia.

44

44

Modos de cifra - comparação

- ❖ Reforço de segurança
 - Esconder padrões de texto original
 - exemplo: ECB ☹
 - Confusão na entrada da cifra
 - exemplo : CBC com realimentação de bits do criptograma
 - Possibilidade de reutilização de uma chave de cifra
 - Cifras contínuas – cuidado !
 - Alteração determinística do texto em claro através da manipulação do criptograma
 - Cifra contínua – fácil
- ❖ Optimização
 - Efectuar pré-processamento
 - Exemplos: OFB e CTR
 - Paralelização do modo de cifra
 - Exemplos: ECB, CTR
- ❖ Tolerância a faltas
 - Propagação de erros
 - Exemplo: ECB – erro num bit apenas afecta o respectivo criptograma
 - Recuperação de sincronismo de perda de bits

© 2024 DI-FCUL Reprodução proibida sem autorização prévia.

45

45

Criptografia Simétrica – vantagens/desvantagens

- ❖ Chave secreta
 - se perdida ou revelada em qualquer ponta, o canal é comprometido
- ❖ Distribuição de chaves
 - O problema do ovo e da galinha: “*como distribuir a chave para ter canais seguros sem ter canais seguros?*”
 - E se as chaves precisarem de ser mudadas frequentemente
- ❖ Gestão de chaves
 - Grande escala ☹
 - Comunicação arbitrária entre 10 participantes requer 45 chaves
 - 100 participantes -> quase 5000
 - $(n(n-1)/2)$ chaves são requeridas para n participantes
- ❖ Eficientes

© 2024 DI-FCUL Reprodução proibida sem autorização prévia.

46

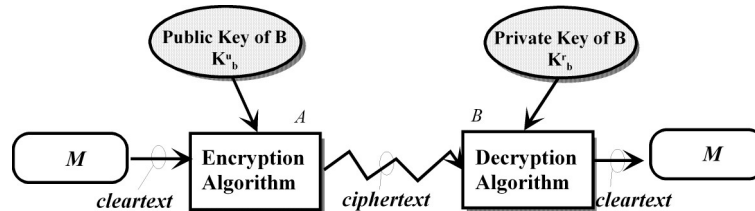
46

Sumário

- ❖ Introdução
- ❖ Criptografia e criptanálise
- ❖ Evolução da tecnologia de cifra
- ❖ Tipos de cifra
 - Cifras de transposição
 - Cifras de substituição
 - Monoalfabéticas
 - Polialfabéticas
- ❖ Conceitos teóricos
- ❖ Cifras modernas
 - Cifras simétricas
 - Modos de operação
 - Cifras por blocos
 - Cifras Contínuas
 - **Cifras assimétricas**
 - **Cifras híbridas**
 - Funções de síntese
 - Autenticação
 - MACs
 - Assinaturas digitais

47

Criptografia Assimétrica



- ❖ Também chamada de cifra de chave pública
 - Cifra com chave pública K_u e decifra com chave privada K_r
 - Em geral é muito mais lenta que a criptografia simétrica
- ❖ PRINCÍPIO:
 - Usam problemas matemáticos, para os quais não existe **solução em tempo polinomial**, aplicados a **grande números** – factorização, cálculo de logaritmos discretos e knapsacks
- ❖ PROPRIEDADES:
 - $D(K_r, E(K_u, m)) = m$ e $E(K_u, D(K_r, m)) = m$
- ❖ Exemplos de Algoritmos:
 - Diffie-Hellman, para calcular um número secreto partilhado (1976); Rivest-Shamir-Adleman (RSA) (1978), ElGamal

48

Criptografia Assimétrica - Diffie-Hellman

❖ PRINCÍPIO:

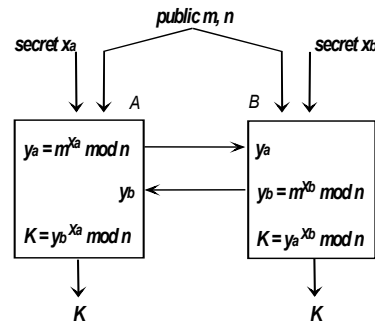
- Lançou as bases para criptografia assimétrica em 1976
- É baseado em uma *one-way function* (função irreversível) e na dificuldade em factorizar números resultantes do produto de números primos grandes

❖ OBJECTIVO:

- Obter um número secreto K , partilhado entre A e B, sem o comunicar em claro

❖ OPERAÇÃO:

- escolher dois números primos m e n públicos (n grande)
- A gera um número aleatório x_a
- A calcula $y_a = m^{x_a} \text{ MOD } n$
- B gera um número aleatório x_b
- B calcula $y_b = m^{x_b} \text{ MOD } n$
- y_a e y_b são tomados públicos
- Cada um calcula K localmente
- $K = y_b^{x_a} \text{ MOD } n = y_a^{x_b} \text{ MOD } n = m^{x_a x_b} \text{ MOD } n$



© 2024 DI-FCUL Reprodução proibida sem autorização prévia.

49

49

Segurança do Diffie-Hellman

❖ A segurança do Diffie-Hellman é baseada na dificuldade em se resolver o seguinte problema:

- Dado um elemento m e os valores m^x e m^y , qual o valor de m^{xy} ?

❖ Isto é equivalente ao cálculo de logaritmos discretos:

- $x = \log_m m^x$
- $y = \log_m m^y$

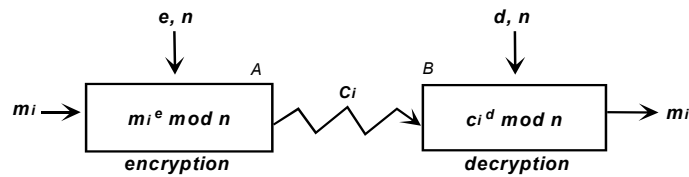
(seria relativamente simples se não estivessemos a falar de aritmética modular – lembrem-se dos “*mod n*” nas fórmulas)

© 2024 DI-FCUL Reprodução proibida sem autorização prévia.

50

50

Criptografia Assimétrica - RSA



- ❖ Foi publicado em 1977 por três investigadores do MIT: Rivest, Shamir e Adleman (RSA)
- ❖ Pode ser usado tanto para cifrar quanto para assinar

51

Criptografia Assimétrica - RSA

É mais lento à medida
que d e e crescem:
(d é geralmente grande
enquanto e é pequeno)

- ❖ Gerar Chaves:
 - Escolhe dois números primos grandes p, q
 - Considere $n = pq$ e $z = \phi(n) = (p-1)(q-1)$
 - Escolha $e < n$ tal que e é primo relativo (não tem factores em comum) de $z = \phi(n)$
 - Calcula d tal que $ed \text{ MOD } z = 1$
 - **chave pública:** $K_u = (e, n)$; **chave privada:** $K_r = (d, n)$
- ❖ Cifrar:
 - $E(K_u, m) = m^e \text{ MOD } n = c$
- ❖ Decifrar:
 - $D(K_r, c) = c^d \text{ MOD } n = m$

52

Criptografia Assimétrica - RSA

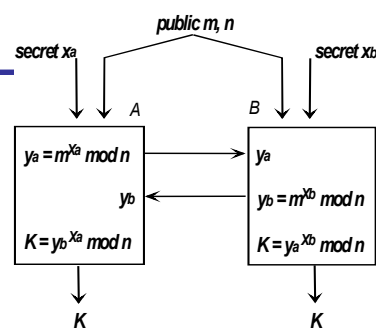
❖ Criptanálise:

1. Procura de chaves “à bruta”
 - Inexequível se usarmos chaves grandes (≥ 1024 bits)
2. Ataques matemáticos
 - d é fácil de calcular a partir de e se forem conhecidos p e q -> factorização de n 's grandes
 - Determinar m a partir de c , e e n -> Função inversa da exponenciação modular: logaritmo modular
 - **ainda seguro com chaves ≥ 1024 bits**
3. Ataques temporais (*timing attacks*) na execução da operação de decifração
 - Consegue estimar d pelo tempo que demora uma decifração

Criptografia Assimétrica

❖ El Gamal

- Baseado em Diffie-Hellman
- Atenção: Bob deve escolher o k aleatório diferente para cada mensagem que envia para Alice.
- Chave pública de Alice: m, n, y_a
- Bob calcula x_b , cifra a mensagem com K e envia y_b para Alice



❖ Elliptic curve cryptography (ECC)

- Tão segura quanto RSA, mas com chaves menores
- Elliptic curve Diffie–Hellman (ECDH)
 - Chave de sessão em canal inseguro
 - <https://www.youtube.com/watch?v=F3zzNa42-tQ>

Criptografia Assimétrica - vantagens/desvantagens

- ❖ Eficiência ☹️
- ❖ Escala 😊
- ❖ Distribuição da chaves públicas – cuidado !
 - autenticidade das chave públicas
 - Como saber se afirmações do tipo “12DH457B6A9 é chave pública do Pedrinho” são verdadeiras?
 - Um chave pública a ser enviada pode ser interceptada e substituída...
 - Ou, se uma base de dados de chaves públicas (PKI ou CA) é comprometida, qualquer chave armazenada pode ser substituída por uma chave falsa criada pelo atacante
 - A definição de **autoridades de certificação** (alguém que certifique a autenticidade das chaves) é necessária

© 2024 DI-FCUL Reprodução proibida sem autorização prévia.

55

55

Cifras híbridas

- ❖ Cifras simétricas
 - eficientes 😊
 - escala ☹️
- ❖ Cifras assimétricas
 - eficientes ☹️
 - escala 😊
- ❖ Cifras híbridas
 - Troca de chaves simétricas com cifras assimétricas

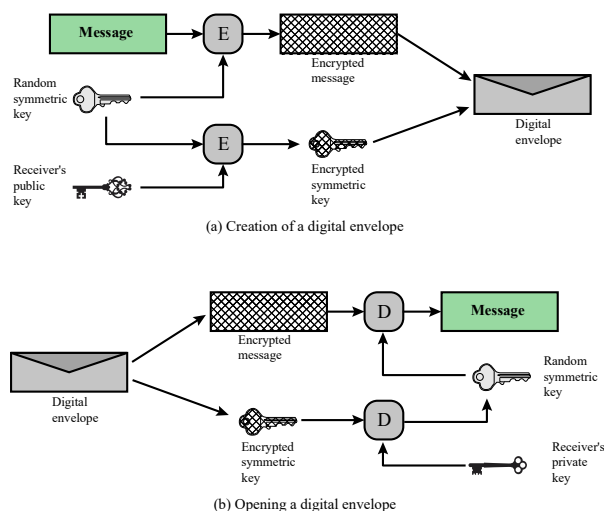


Figure 2.8 Digital Envelopes

© 2024 DI-FCUL Reprodução proibida sem a

56

Sumário

- ❖ Introdução
- ❖ Criptografia e criptanálise
- ❖ Evolução da tecnologia de cifra
- ❖ Tipos de cifra
 - Cifras de transposição
 - Cifras de substituição
 - Monoalfabéticas
 - Polialfabéticas
- ❖ Conceitos teóricos
- ❖ Cifras modernas
 - Cifras simétricas
 - Modos de operação
 - Cifras por blocos
 - Cifras Contínuas
 - Cifras assimétricas
 - Cifras híbridas
 - **Funções de síntese**
 - **Autenticação**
 - **MACs**
 - **Assinaturas digitais**

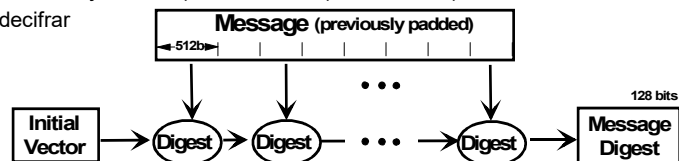
© 2024 DI-FCUL Reprodução proibida sem autorização prévia.

57

57

Síntese Segura ou *Digest* de Mensagens

- ❖ Objectivo
 - Produzem valores de dimensão constante a partir de entradas (mensagens, ficheiros, ...) de dimensão variável
 - A **função de compressão** é aplicada de forma iterativa
 - 2 argumentos de entrada da função de compressão: síntese prévia, bloco a processar
 - Não servem para cifrar/decifrar



- ❖ Propriedades
 - Resistência à descoberta do texto original
 - Dada a síntese H , é muito difícil descobrir um texto M , tal que $H = h(M)$
 - Resistência à descoberta de um segundo texto original
 - Dado um Texto M , é muito difícil descobrir M' ($M' \neq M$) tal que $h(M) = h(M')$
 - Resistência à colisão
 - É difícil descobrir dois textos quaisquer, M e M' , $M' \neq M$, tais que $h(M) = h(M')$
- ❖ Exemplos de Funções: MD5, SHA-1, SHA-256

© 2024 DI-FCUL Reprodução proibida sem autorização prévia.

58

58

Funções de síntese: Resistência à colisão

❖ Dimensão das sínteses

- Com sínteses de 128 bits e mensagens 1000 bits: $\sim 2^{872}$ mensagens dão origem à mesma síntese
- Síntese ≥ 128 bits

❖ Ataque do Aniversário (*Birthday attack*):

- Baseia-se no “paradoxo do aniversário” e é usado para encontrar um par de mensagens com a mesma síntese (colisão)
- Para sínteses de n bits, o atacante deve tentar aproximadamente $2^{n/2}$ mensagens
- In a group of at least 23 randomly chosen people, there is more than 50% probability that some pair of them will have the same birthday. Such a result is counter-intuitive to many.

Síntese Segura - MD5

- ❖ Proposto por Ron Rivest (investigador do MIT)
- ❖ O último de uma série de funções MD2, MD4, ...
- ❖ Produz um valor de síntese de 128 bits
- ❖ Até recentemente era a função de síntese segura mais usada
 - Recentemente foram encontradas falhas tanto através de ataques de força bruta quanto por criptanálise
- ❖ Não é **recomendada a sua utilização**
- ❖ Especificado num padrão IETF (RFC1321)

Síntese Segura- Secure Hash Algorithm (SHA-1)

- ❖ Proposto pelo NIST e pela NSA
- ❖ SHA-1
 - Produz sínteses de 160 bits
 - Não seguro
- ❖ SHA-2
 - SHA-256
 - SHA-512
 - SHA-224 e SHA-384
- ❖ SHA-3
 - Dimensões idênticas a SHA-2
 - 2012
- ❖ Existem diversas versões do SHA. Estas são mais seguras na medida em que o tamanho da síntese produzida aumenta

Síntese Segura - versões do SHA

	SHA-1	SHA-256	SHA-384	SHA-512
Message digest size	160	256	384	512
Message size	$< 2^{64}$	$< 2^{64}$	$< 2^{128}$	$< 2^{128}$
Block size	512	512	1024	1024
Word size	32	32	64	64
Number of steps	80	80	80	80
Security	80	128	192	256

- Notes: 1. All sizes are measured in bits.
2. Security refers to the fact that a birthday attack on a message digest of size n produces a collision with a workfactor of approximately $2^{n/2}$.

Sínteses seguras

- ❖ Password
 - Armazenamento de passwords
- ❖ Detecção de intrusões
 - Guardar a hash dos ficheiros do SO
- ❖ O que obtemos e como ?
- ❖ Outras aplicações
 - Slides seguintes

MAC - Message Authentication Code

- ❖ Funções de síntese – integridade
- ❖ Message Autentication Code – MAC
 - Usa **chave secreta** partilhada
 - O que garante ?
 - Como produzir MACs
 - Cifrar mensagem e síntese da mensagem
 - Cifrar síntese da mensagem ☹
 - Funções chaveadas
 - Funções de cifra por blocos: DES-MAC (criptograma gerado em modo CBC)
 - Funções de cifra contínua
 - Funções de síntese - HMAC

HMAC - Hash Message Authentication Code

- ❖ Definido no RFC2104 do IETF
- ❖ Utiliza funções de síntese na mensagem:
- ❖ $H_k(m) = h(k' \oplus opad \parallel h(k' \oplus ipad \parallel m))$
 - \oplus ou exclusivo
 - \parallel concatenação
 - K' é a chave K preenchida (*padded*) para ter o tamanho de um bloco
 - *opad* e *ipad* são constantes específicas de preenchimento
 - Overhead é de apenas mais 3 execuções da função de síntese em relação à geração do *digest* da mensagem
- ❖ Qualquer função de síntese segura pode ser usada no HMAC.
- ❖ A segurança e a eficiência do algoritmo dependem da função de síntese segura usada.
- ❖ A função de síntese pode ser substituída com o fim de melhorar segurança e/ou eficiência.

© 2024 DI-FCUL Reprodução proibida sem autorização prévia.

65

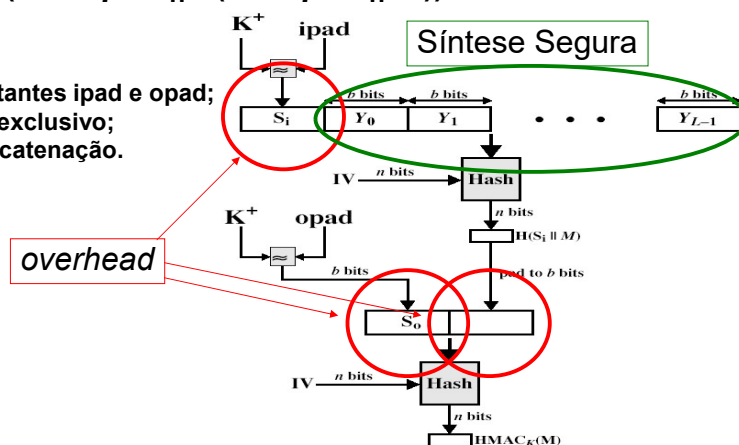
65

HMAC - Hash Message Authentication Code

$$H_k(m) = h(k' \oplus opad \parallel h(k' \oplus ipad \parallel m))$$

onde:

- Constantes *ipad* e *opad*;
- \oplus ou exclusivo;
- \parallel concatenação.



© 2024 DI-FCUL Reprodução proibida sem autorização prévia.

66

66

MAC - Message Authentication Code

❖ MAC:

- Alice e Beto compartilham uma chave k
- Alice envia $m \parallel H_k(m)$ para Beto
- Beto verifica que recebeu m de Alice

❖ E perante terceiros?

❖ Beto pode provar que foi Alice que enviou ?

ERRADO!

❖ Isto não é uma assinatura digital!

- Uma terceira parte não pode determinar se foi Alice ou Beto quem gerou a mensagem
- i.e., não satisfaz a propriedade de não-repudição

Assinaturas Digitais

❖ Autenticidade

- quem assinou é identificável univocamente pela assinatura

❖ Integridade

- uma assinatura correcta num documento garante que este não é alterável sem detecção
- Não-reutilização
 - a assinatura ou parte do documento não é reutilizável em outro documento

❖ Não-repudição

- o assinante não pode negar a sua assinatura
- Não-forjamento
 - quem assinou é o próprio e fê-lo deliberadamente

Assinaturas Digitais

❖ Modo de operação

- Cifrar com chave privada a síntese do texto original
- Algoritmos de assinatura mais usados
 - RSA
 - DSA

❖ DSS/DSA

- Proposta de assinaturas digitais do NIST (1991)
- Digital Signature Standard (DSS)
- Digital Signature Algorithm (DSA)
 - derivado do algoritmo de assinatura ElGamal

Assinaturas às cegas

❖ Assinaturas às cegas

- David Chaum, 1982
- Garante o anonimato de quem solicita a assinatura e do que é efectivamente assinado
- Utilidade
 - Dinheiro electrónico
 - Pagamentos anónimos
 - Votação electrónica

➤ Exemplo com RSA

- K aleatório

1. Obscurecer: $m' = k^e \cdot m \bmod n$
2. Assinar: $A(m') = (m')^d \bmod n$
3. Anular 1: $A = k^{-1} \cdot A(m') \bmod n$

$$\begin{aligned} k^{-1} \cdot A(m') &\equiv k^{-1} \cdot (m')^d \\ &\equiv k^{-1} \cdot (k^e \cdot m)^d \\ &\equiv k^{-1} \cdot k^{ed} \cdot m^d \\ &\equiv k^{-1} \cdot k \cdot m^d \\ &\equiv m^d \pmod{n} \end{aligned}$$

Cifras homomórficas

❖ Decifra(Operação(criptograma))=Operação'(texto em claro)

❖ $f(a \oplus b) = f(a) \otimes f(b)$

❖ Exemplo

➤ RSA multiplicação

$$E(a).E(b)=a^e b^e \bmod n = (ab)^e \bmod n = E(ab)$$

❖ Casos de uso

- Verificação de vírus sem decifra ficheiros
- Pesquisa em ficheiros cifrados (na nuvem)

Números aleatórios

❖ Usos

➤ Chaves,

❖ Aleatório

❖ Imprevisível

❖ Pseudo-aleatório

Bibliografia

- ❖ Segurança em Redes Informáticas
 - Cap 2
- ❖ Computer Security Principles and Practice
 - Cap 2, 20, 21