Modos de operação de cifras

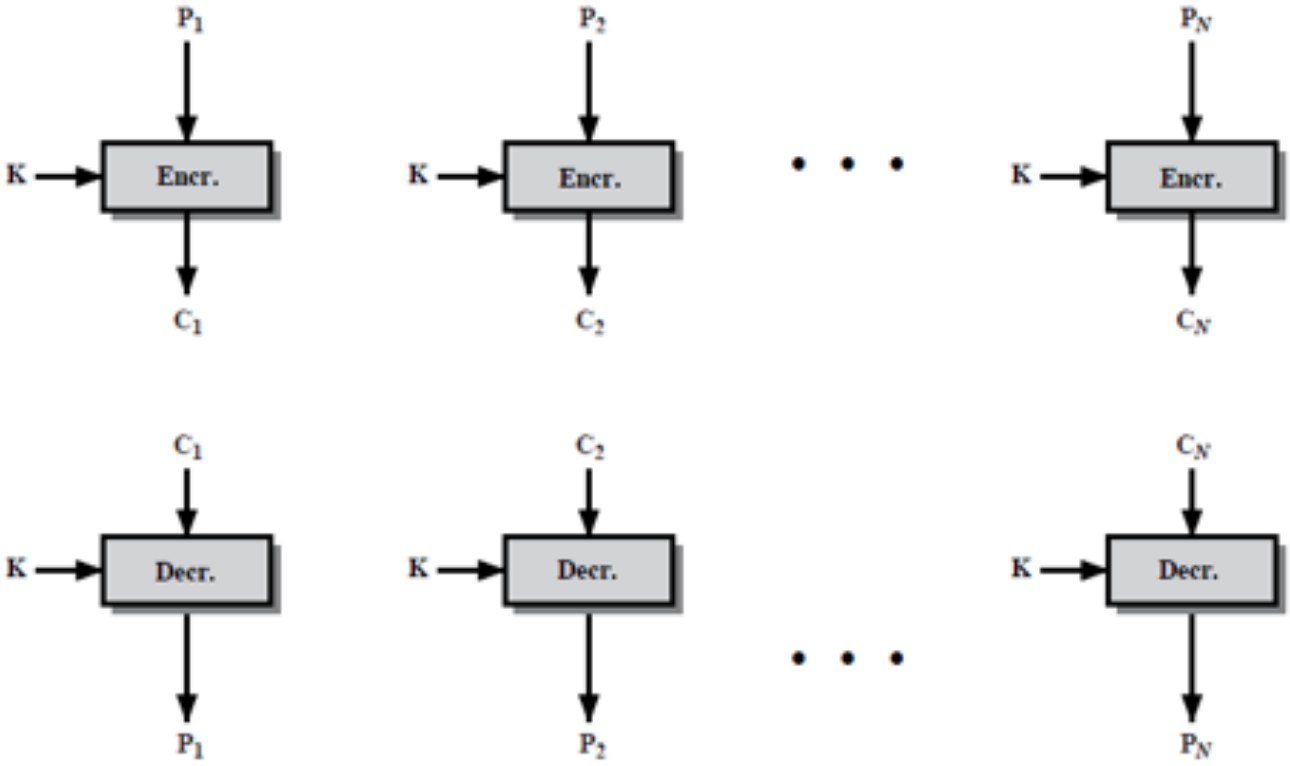
Modos de Operação

- Uma cifra de bloco define um conjunto de transformações indexadas por uma chave
- Cada bloco de n bits é levado em um outro bloco de n bits
- Quando a mensagem excede n bits, diversas abordagens são possíveis
  - Exemplo: quebrar as mensagens em blocos de n bits e encriptá-las individualmente

Modo ECB

- Electronic Codebook (é o exemplo anterior)
- Entrada:
  - Chave k
  - mensagem composta de t blocos de n bits, $m=x_1x_2...x_t$
- Saída: mensagem cifrada c1c2...ct
  - Onde $c_i:=E_k(x_i)$
  - Decriptação: $x_i:=E_k^{-1}(c_i)$
- Blocos idênticos, na mensagem plana, resultam em blocos idênticos, na mensagem cifrada
  - Reordenação dos blocos na mensagem plana provoca simples reordenação na mensagem cifrada

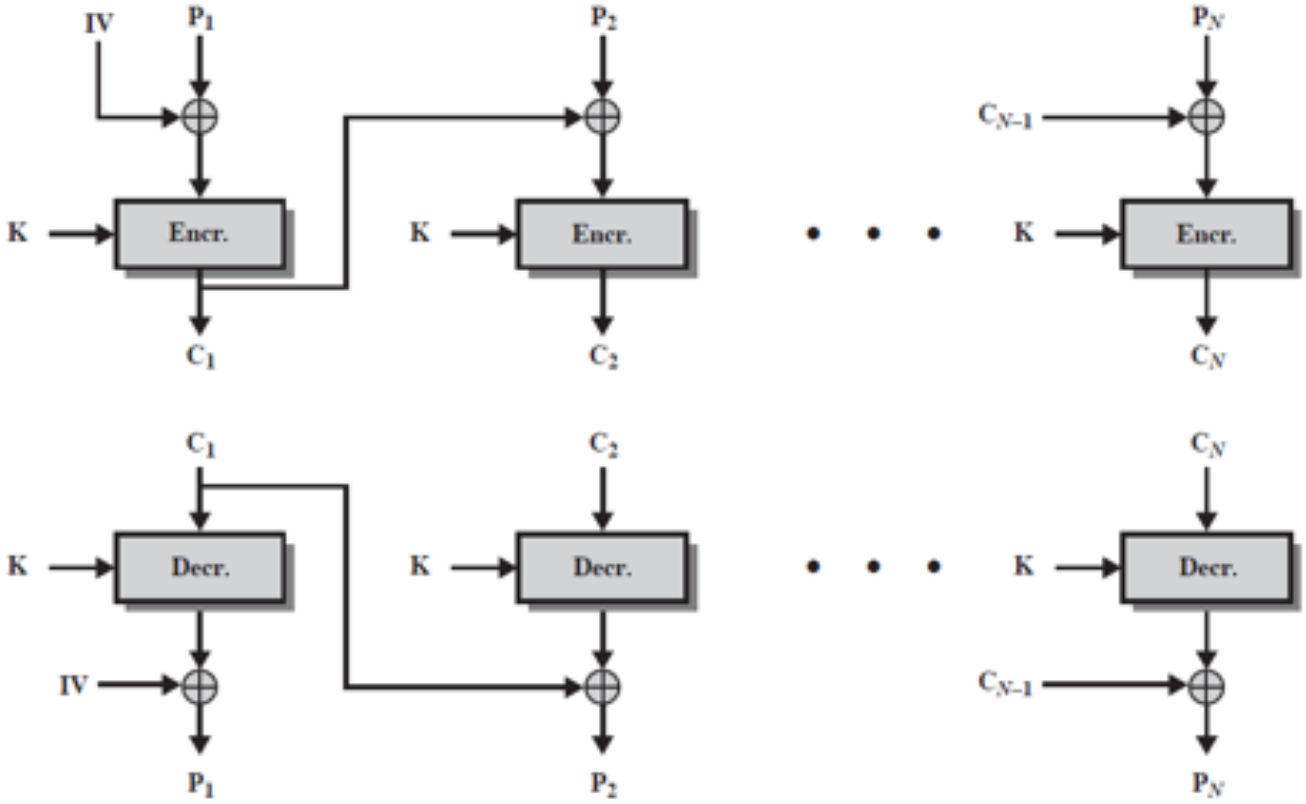
Modo ECB



Modo CBC

- Cipher Block Chaining
- Bloco cifrado depende do bloco cifrado anterior
- Entrada:
  - Chave *k*
  - mensagem composta de t blocos de n bits, $M=x_1x_2...x_t$
  - Inicialization vector IV de *n* bits
- Saída: mensagem cifrada $c_1c_2...c_t$
  - Onde $c_i:=E_k(x_i \oplus c_{i-1})$; $c_0=IV$
  - Decriptação: $x_i:=c_{i-1} \oplus E_k\text{-}1(c_i)$
- Rearranjo de blocos na mensagem plana determina conjunto diferente de blocos na mensagem cifrada

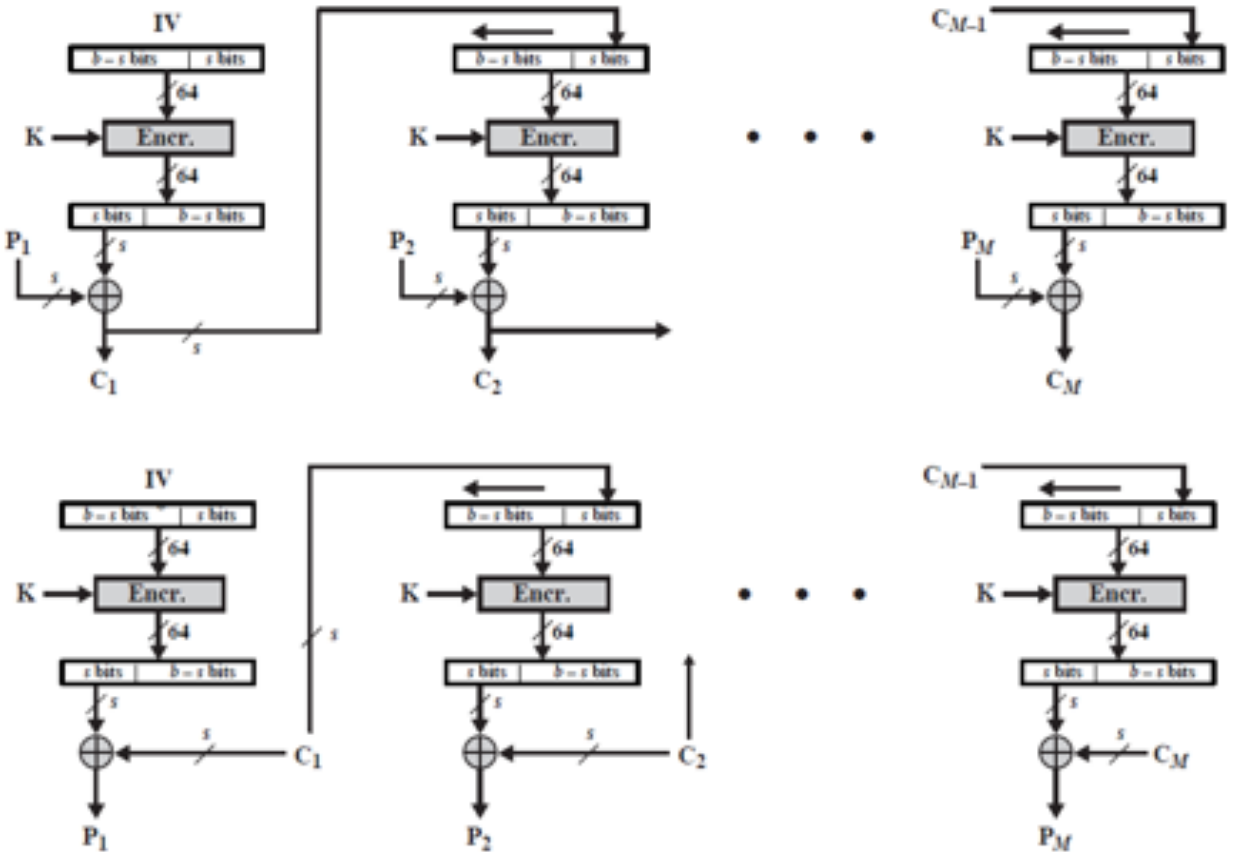
Modo CBC


IV
P1
P2
PN
CN–1
K
Encr.
K
Encr.
K
Encr.
C1
C2
CN
C1
C2
CN
K
Decr.
K
Decr.
K
Decr.
IV
CN–1
P1
P2
PN

Modo CFB

- Cipher Feedback Mode
- Conversão de uma cifra de bloco em uma espécie de cifra de stream
  - s bits de saída são combinados com s bits da mensagem plana para gerar s bits da mensagem cifrada
    - esses s bits cifrados ainda realimentam a entrada
  - b-s bits de saída são descartados

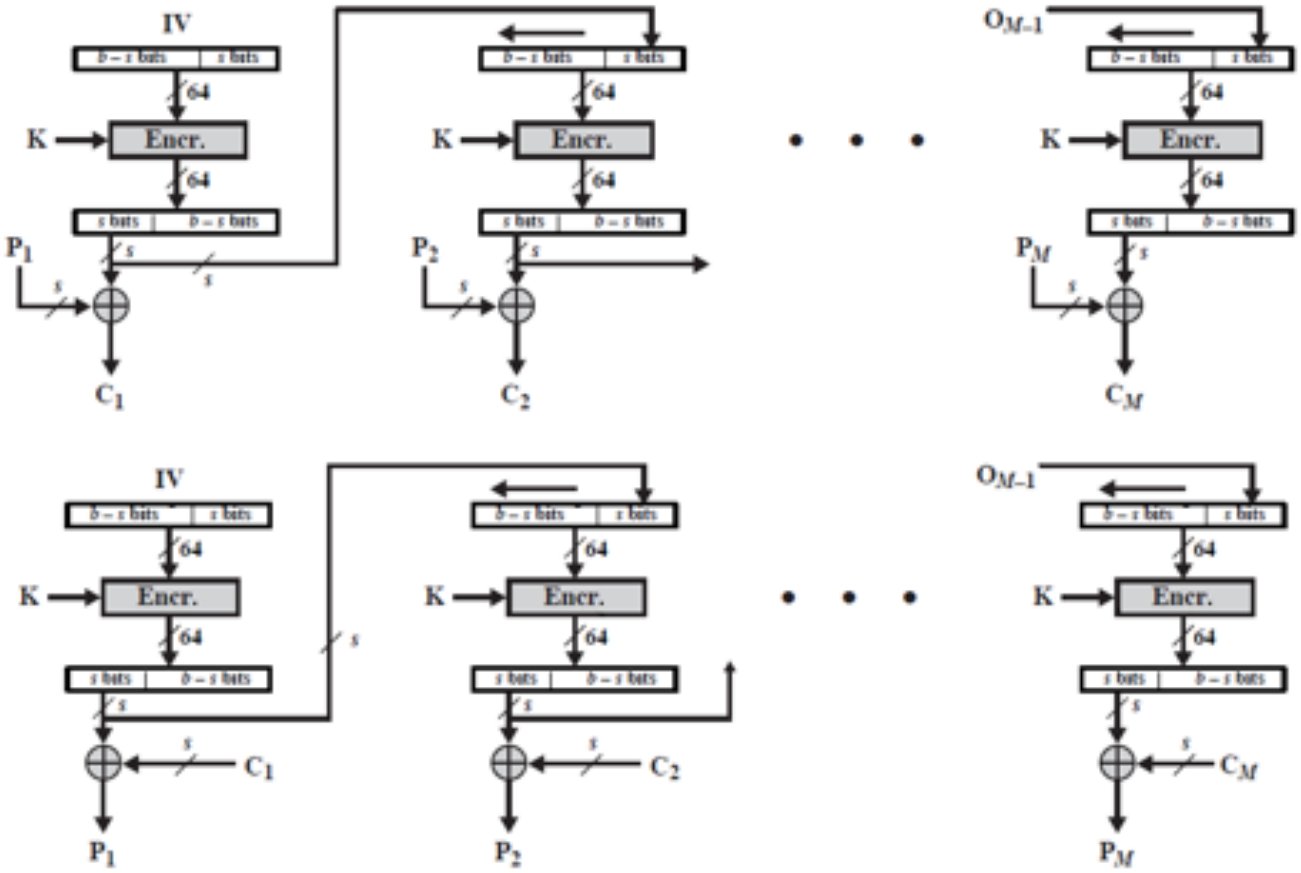
Modo CFB



Modo OFB

- Output Feedback Mode
- Muito parecido com CFB
  - Diferença: os s bits que realimentam a entrada são tomados antes de serem combinados com a mensagem plana

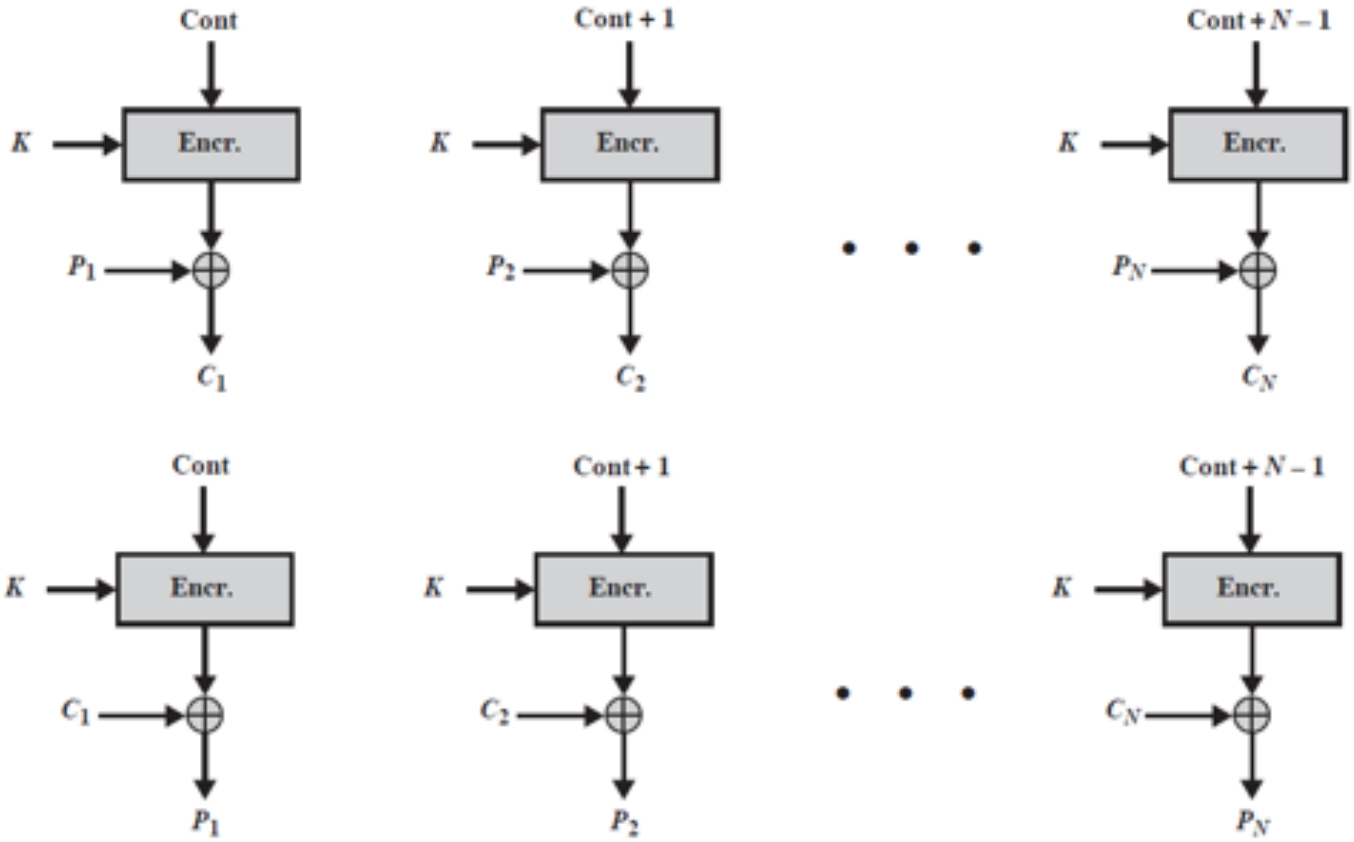
Modo OFB



Modo CTR

- Counter Mode
- Um contador é incrementado, encriptado e combinado com a mensagem plana
- Diversas vantagens em relação aos outros modos:
  - Paralelismo / eficiência (hardware e software)
  - Preprocessamento
  - Acesso aleatório
  - Segurança "demonstrável"
  - Simplicidade

Modo CTR


Cont
K
Encr.
P_1
C_1
Cont + 1
K
Encr.
P_2
C_2
Cont + N – 1
K
Encr.
P_N
C_N
Cont
K
Encr.
C_1
P_1
Cont + 1
K
Encr.
C_2
P_2
Cont + N – 1
K
Encr.
C_N
P_N

Aplicações típicas dos modos

- Electronic Codebook
  - Transmissão de mensagens curtas
- Cipher Block Chaining
  - Uso geral orientado a bloco, autenticação
- Cipher Feedback
  - Uso geral orientado a stream
- Output Feedback
  - Uso orientado a stream em canais ruidosos (satélite)
- Counter Mode
  - Uso orientado a bloco com requisitos de alta velocidade

›Criptografia de chave pública; funções hash; assinatura digital

›Criptografia de chave pública

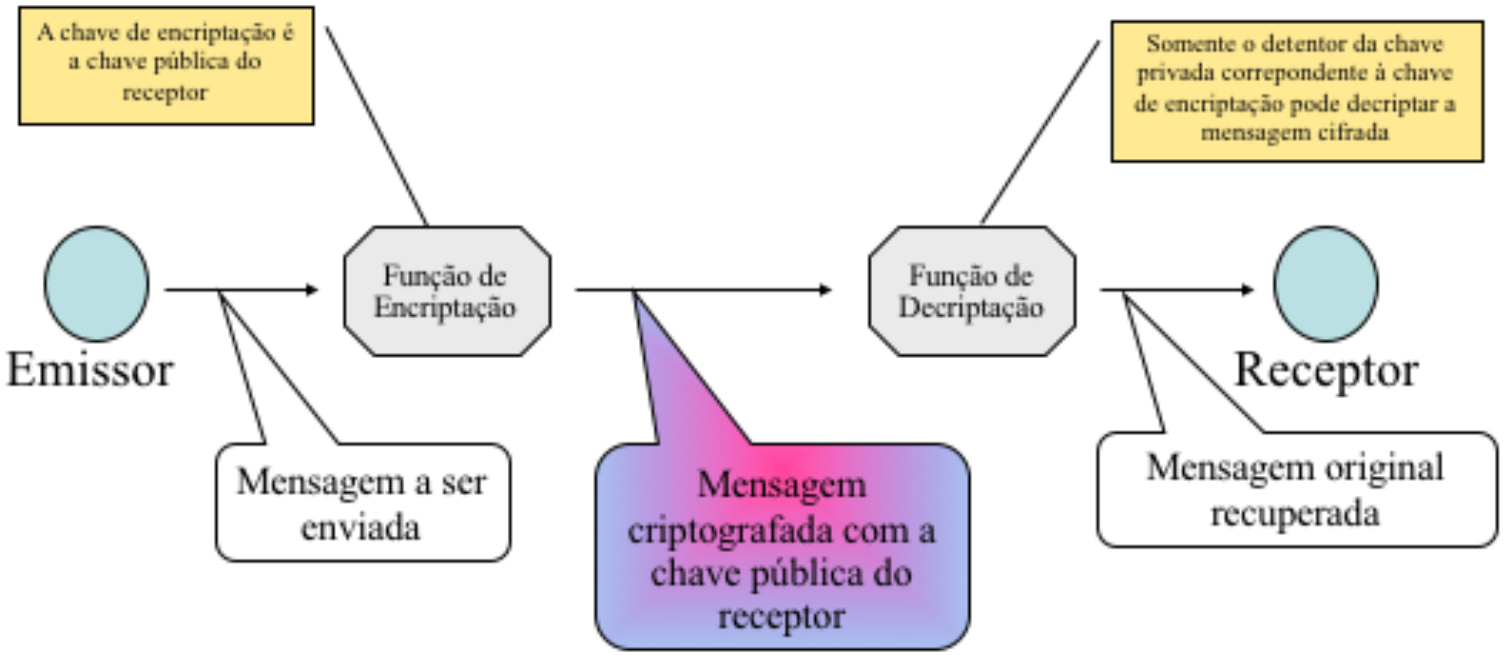
Criptografia "convencional"

- Criptografia tradicional (chave simétrica ou secreta) usa um par de chaves conhecido por emissor e receptor
  - De certa forma, pode-se entender que a chave é única, já que, conhecida a chave de encriptação, determina-se facilmente a chave e a transformação de decriptação
- Se uma chave é divulgada, toda a comunicação é comprometida
- É dita simétrica, no sentido de que ambas as partes conhecem o par de chaves

Criptografia de chave pública

- Avanço mais significante em 3000 anos de criptografia
- Funciona – de fato – com duas chaves
  - Não se pode determinar a chave de decriptação a partir da chave de encriptação
- É dita assimétrica, no sentido de que os participantes possuem papéis diferentes
  - Quem encripta não é capaz de decriptar
- É baseada em funções construídas a partir de problemas computacionalmente difíceis, de campos como Álgebra e Teoria dos Números
- Complementa – em vez de substituir – a criptografia de chave secreta

Funcionamento básico de um esquema criptográfico



Breve histórico da criptografia de chave pública

- Descoberta oficialmente (publicamente) por Whitfield Diffie & Martin Hellman (Stanford 1977)
  - Conhecida anteriormente em comunidades restritas
    - Serviço de Segurança britânico já conhecia em 1970
    - Segundo NSA, já era conhecida desde meados de 1960s
- Na verdade, um protocolo de troca de chaves
- Em 1977, Rivest, Shamir e Adleman desenvolvem cifra de chave pública
- Até hoje, o RSA permanece como a cifra mais usada (?)

Características da criptografia de chave pública

- Os algoritmos de chave pública baseiam-se em duas chaves/transformações criptográficas com as seguintes características:
  - é computacionalmente simples encriptar (resp. decriptar) mensagens quando a chave relevante de encriptação (resp. decriptação) é conhecida
  - é computacionalmente inviável encontrar a chave de decriptação conhecendo-se apenas o algoritmo e a chave de encriptação

Aplicações da criptografia de chave pública

- Três aplicações freqüentes:
  - encriptação/decriptação (provêem confidencialidade)
  - assinaturas digitais (provêem autenticação)
  - troca de chaves simétricas (protocolo de troca de chaves para uma sessão)

Segurança de esquemas baseados em criptografia de chave pública

- Como o oponente conhece a chave de encriptação, é sempre possível tentar explorar a busca exaustiva
  - Por esse motivo, usam-se chaves bastante extensas (>512bits)
- A segurança baseia-se na grande diferença entre a dificuldade entre
  - A operação fácil de encriptar/decriptar conhecendo-se a chave apropriada;
  - O difícil problema de criptanalisar uma mensagem (decriptar sem a chave apropriada)

Funções one-way e trapdoor one-way

- Função one-way
  - Facilmente calculável em todo o seu domínio
  - Cálculo da inversa computacionamente difícil
  - Ex.: multiplicação X fatoração
- Função trapdoor one-way
  - Facilmente calculável em todo o seu domínio
  - Cálculo da inversa computacionamente difícil
    - Entretanto, possuir determinada informação torna o cálculo da inversa mais fácil
  - Ex.: cubo X raiz cúbica modular

Função *one-way* - exemplo

- Seja $X=\{1,\ldots,16\}$ e $f(x)$ o resto da divisão de 3^x por 17
  - Dado $x \in X$, é relativamente fácil obter $f(x)$
  - Entretanto, não é tão fácil obter, por exemplo, o valor de x tal que $f(x)=7$.
    - Provavelmente teremos que tentar todas as 16 possibilidades

x	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$f(x)$	3	9	10	13	5	15	11	16	14	8	7	4	12	2	6	1

Função trapdoor one-way - exemplo

- Selecione dois números primos
  - Exemplo: p=488611 e q=53993
- Calcule $n=pq$
  - No exemplo, n=2624653723
- Seja $f(x)$ o resto da divisão de x^3 por n
  - No exemplo, $f(2489991)$ = 1981394214
- Dado $f(x)$, é difícil obter x (raiz cúbica modular com módulo n)
- Entretanto, conhecidos os fatores p e q de n, existem algoritmos eficientes para obter a raiz cúbica modular
  - Este é um exemplo de função *trapdoor one-way*

RSA

- Descrito por Rivest, Shamir & Adleman (MIT) em 1977
- Cifra de chave pública mais conhecida e utilizada
- Baseado na operação de exponenciação (módulo um número primo)
  - Exponenciação leva $O((\log n)^3)$ operações (fácil)
- Usa grandes números inteiros (ex.: 1024 bits)
- Segurança baseada no custo de fatorar grandes primos
  - Fatoração leva $\Theta(e^{\log n \log \log n})$ operações (difícil)

Estabelecimento de chaves no RSA

- Um usuário gera par de chaves da seguinte forma:
  - Seleciona dois números primos p e q aleatorios
  - Computa o módulo $N=pq$
    - Notação: $\phi(N)=(p-1)(q-1)$
  - Seleciona a chave de encriptação e aleatoriamente
    - onde $1<e<\phi(N)$, $gcd(e,\phi(N))=1$
  - Resolve a seguinte equação para obter a chave de decriptação d
    - $ed=1 \ mod \ \phi(N)$ and $0 \leq d \leq N$
  - Publica sua chave pública, o par (e,N)
    - E mantém privada a chave (d,p,q)

Uso do RSA

- Para encriptar uma mensagem M, o emissor
  - Obtém a chave pública (e,N) do receptor
  - Computa $C=M^e \bmod N$, onde $0 \leq M < N$
- Para decriptar C, o receptor
  - Usa sua chave privada(d,p,q) para computar $M=C^d \bmod N$
- Note que a mensagem M é menor que o módulo N
  - caso necessário, a mensagem M é dividida em blocos.

Demonstração do funcionamento do RSA

- Como $ed=1 \ mod \ ø$, existe k tal que $ed=1+kø$.
- Pequeno Teorema de Fermat:
  - se $gcd(a,p)=1$, então $a^{p-1}=1 \ mod \ p$
  - Ou seja, se $gcd(m,p)=1$, então $m^{p-1}=1 \ mod \ p$
- Elevando a $k(q-1)$ e multiplicando por m
  - $m^{k(p-1)(q-1)}+1=m \ mod \ p$
- A relação também vale se $gcd(m,p)=p$, pois ambos os lados são nulos módulo p.
- Assim, $m^{ed}=m \ mod \ p$; analogamente $m^{ed}=m \ mod \ q$
- Como p e q são primos distintos, $m^{ed}=m \ mod \ n$
- Finalmente, $c^d = (m^e)^d = m \ mod \ n$

Exemplo RSA

1. Selecione primos: p=17 & q=11
2. Compute $n = pq$ =17×11=187
3. Compute $\phi(n)=(p-1)(q-1)=16\times10=160$
4. Selecione e : gcd(e,160)=1; escolha e=7
5. Determine d tal que de=1 mod 160 e $d < 160$
   - O valor é d=23, pois 23×7=161= 10×160+1
6. Divulgue a chave pública (7,187)
7. Guarde a chave privada (23,17,11)

Exemplo RSA (cont.)

- Relembrando:
  - Chave pública: (7,187)
  - Chave privada: (23,17,11)
- Exemplo de encriptação/decriptação:
  - Mensagem $M = 88$ (observe que $88<187$)
  - encriptação:

$$C = 88^7 \bmod 187 = 11$$

  - decriptação:

$$M = 11^{23} \bmod 187 = 88$$

”Invertendo” a cifra

- Observe que as transformações $E_e(m)$e $D_d(m)$ (encriptação/decriptação) uma cifra de chave pública são bijeções.
  - Logo, $D_d(E_e(m))=E_e(D_d(m))$
- Suponha que invertemos a seqüência das operações:
  - Primeiro, um usuário *A* usa sua chave privada para aplicar a operação de decriptação em *m*, obtendo $D_d(m)$
  - Qualquer entidade poderá usar a chave pública de *A* para obter a mensagem original *m*
  - Se o espaço das mensagens válidas for “bem gerenciado”, esse esquema pode ser usado para gerar assinaturas
  - Veremos essa abordagem mais à frente, no curso

Fundamentos de Criptografia

›Autenticação de mensagens:
›funções hash e MAC

Função hash

- Primitiva criptográfica fundamental na criptografia moderna.
- Função que mapeia strings de tamanho arbitrário em strings de algum tamanho fixo
  - As strings de tamanho fixo são chamadas resumo, valor-hash ou simplesmente hash
  - O mapeamento deve ser eficiente computacionalmente
- A idéia é que o hash funcione como uma "impressão digital" de uma string

Aplicações de funções hash

- Integridade de dados
  - Hash de determinada informação é computado em algum momento;
  - O valor do hash é mantido protegido de alguma forma;
  - Em um momento posterior, recalcula-se o hash da informação e compara-se o novo hash com o antigo
  - Aplicação típica: integridade de software
- Assinatura digital/autenticação
  - Mensagem longa é hashed e apenas o resumo (hash) é assinado/autenticado
    - Economia de tempo e espaço

Requisitos de uma função hash

- Para que seja útil para autenticação, uma função hash H deve possuir as propriedades
  - H pode ser aplicado a strings de comprimento arbitrário
  - H produz saída de comprimento fixo
  - H é facilmente computável
  - Propriedade "one-way"
    - Dado h qualquer, é inviável encontrar x tal que H(x)=h
  - Resistência fraca a colisão
    - Dado x, é inviável encontrar y tal que H(x)=H(y)
  - Resistência forte a colisão
    - É inviável encontrar um par (x,y) tal que H(x)=H(y)

Secure Hash Algorithm (SHA)

› Desenvolvido pelo NIST (FIPS 180 de 1993)

› Várias revisões posteriores...

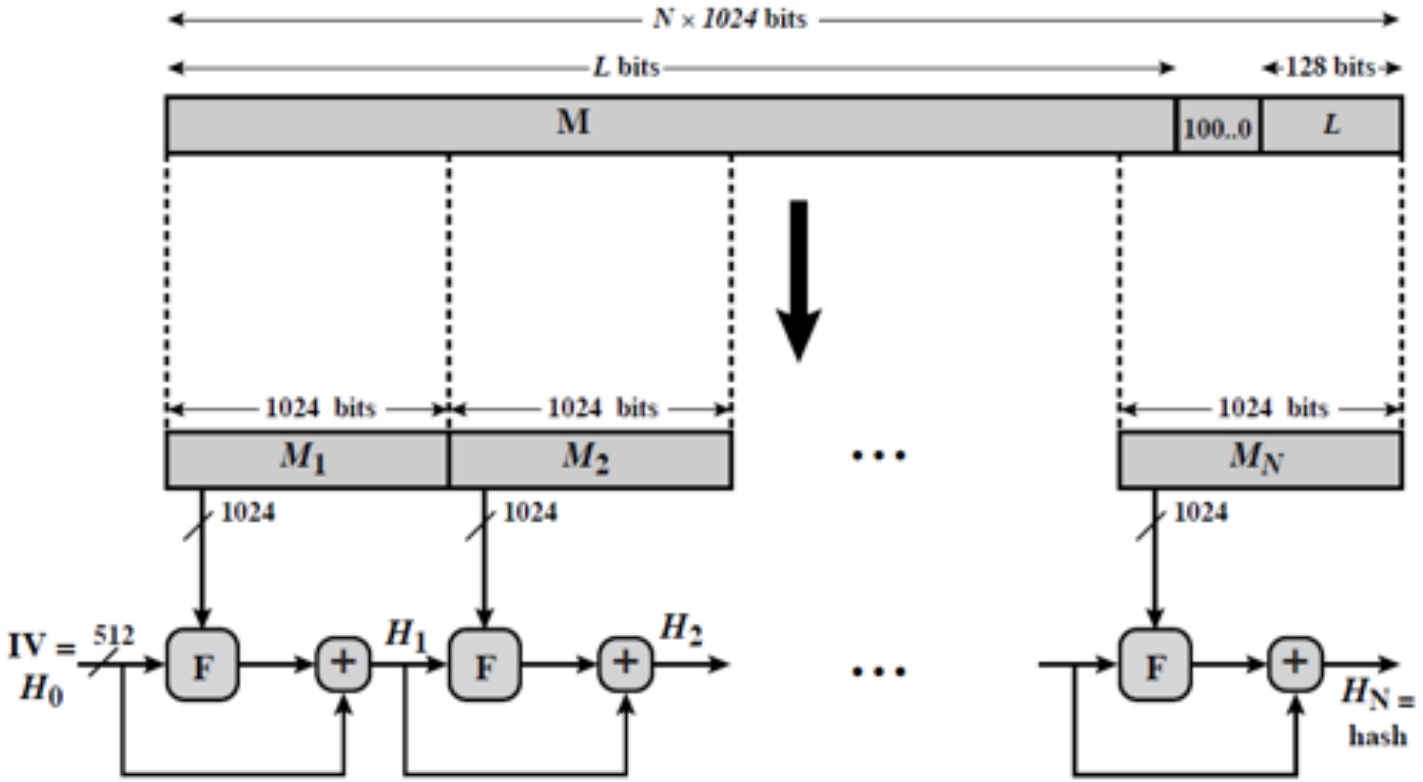
Padrões SHA

- SHA-0: Nome retroativo aplicado à versão original da função hash de 160 bits publicada em 1993 sob o nome "SHA". Ele foi retirado logo após a publicação devido a uma "falha significativa" não revelada e substituído pela versão revisada SHA-1.
- SHA-1: Uma função de hash de 160 bits que se assemelha ao algoritmo MD5. Este foi concebido pela Agência Nacional de Segurança (NSA) para fazer parte do algoritmo de assinatura digital. Fraquezas criptográficas foram descobertas no SHA-1, e o padrão não foi mais aprovado para a maioria dos usos criptográficos após 2010.
- SHA-2: Uma família de duas funções hash similares, com diferentes tamanhos de bloco, conhecidas como SHA-256 e SHA-512. Eles diferem no tamanho da palavra; O SHA-256 usa palavras de 32 bits em que o SHA-512 usa palavras de 64 bits. Existem também versões truncadas de cada padrão, conhecidas como SHA-224, SHA-384, SHA-512/224 e SHA-512/256. Estes também foram projetados pela NSA.
- SHA-3: Uma função hash anteriormente chamada Keccak, escolhida em 2012 após uma competição pública entre criptógrafos não pertencentes à NSA. Ele suporta os mesmos comprimentos de hash que o SHA-2 e sua estrutura interna difere do restante da família SHA.

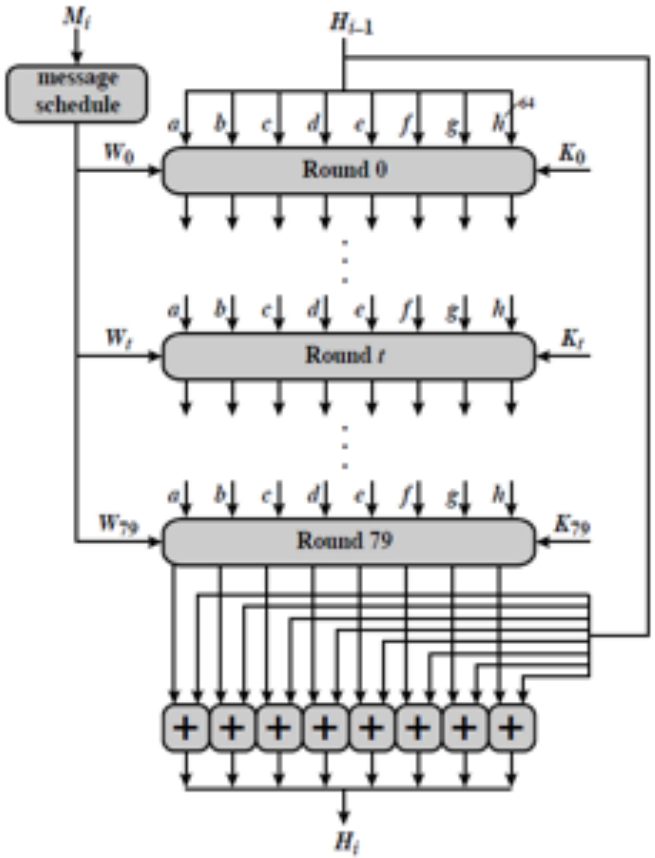
Comparação de funções SHA

Algorithm and variant		Output size (bits)	Internal state size (bits)	Block size (bits)	Max message size (bits)	Rounds	Operations	Security bits (Info)	Capacity against length extension attacks	Performance on Skylake (median cpb)[1]		First Published
										long messages	8 bytes	
MD5 (as reference)		128	128 (4 × 32)	512	Unlimited[2]	64	And, Xor, Rot, Add (mod 2^{32}), Or	<64 (collisions found)	0	4.99	55.00	1992
SHA-0		160	160 (5 × 32)	512	$2^{64} - 1$	80	And, Xor, Rot, Add (mod 2^{32}), Or	<34 (collisions found)	0	≈ SHA-1	≈ SHA-1	1993
SHA-1								<63 (collisions found[3])		3.47	52.00	1995
SHA-2	*SHA-224*	224	256 (8 × 32)	512	$2^{64} - 1$	64	And, Xor, Rot, Add (mod 2^{32}), Or, Shr	112	32	7.62	84.50	2004
	SHA-256	256						128	0	7.63	85.25	2001
	SHA-384	384	512 (8 × 64)	1024	$2^{128} - 1$	80	And, Xor, Rot, Add (mod 2^{64}), Or, Shr	192	128 (≤ 384)	5.12	135.75	
	SHA-512	512						256	0	5.06	135.50	
	SHA-512/224	224						112	288	≈ SHA-384	≈ SHA-384	
	SHA-512/256	256						128	256			
SHA-3	*SHA3-224*	224	1600 (5 × 5 × 64)	1152	Unlimited[4]	24[5]	And, Xor, Rot, Not	112	448	8.12	154.25	2015
	SHA3-256	256		1088				128	512	8.59	155.50	
	SHA3-384	384		832				192	768	11.06	164.00	
	SHA3-512	512		576				256	1024	15.88	164.00	
	SHAKE128	*d* (arbitrary)		1344				min(*d*/2, 128)	256	7.08	155.25	
	SHAKE256	*d* (arbitrary)		1088				min(*d*/2, 256)	512	8.59	155.50	

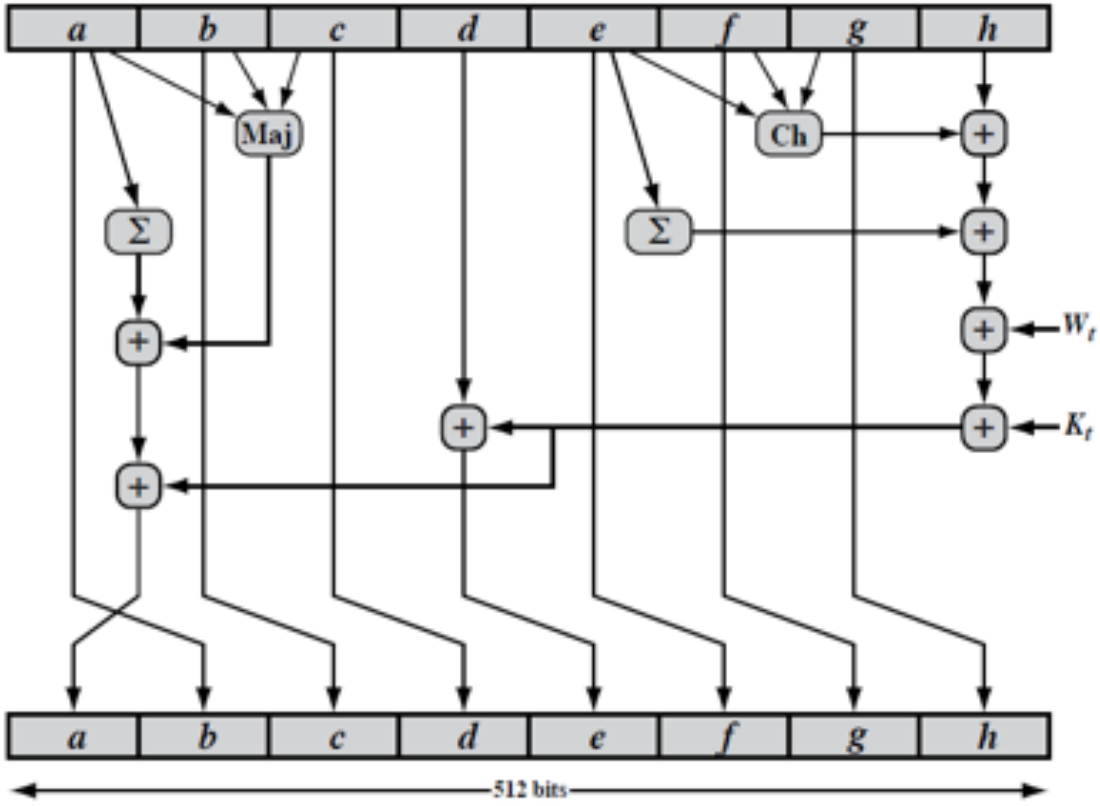
Estrutura do SHA-512



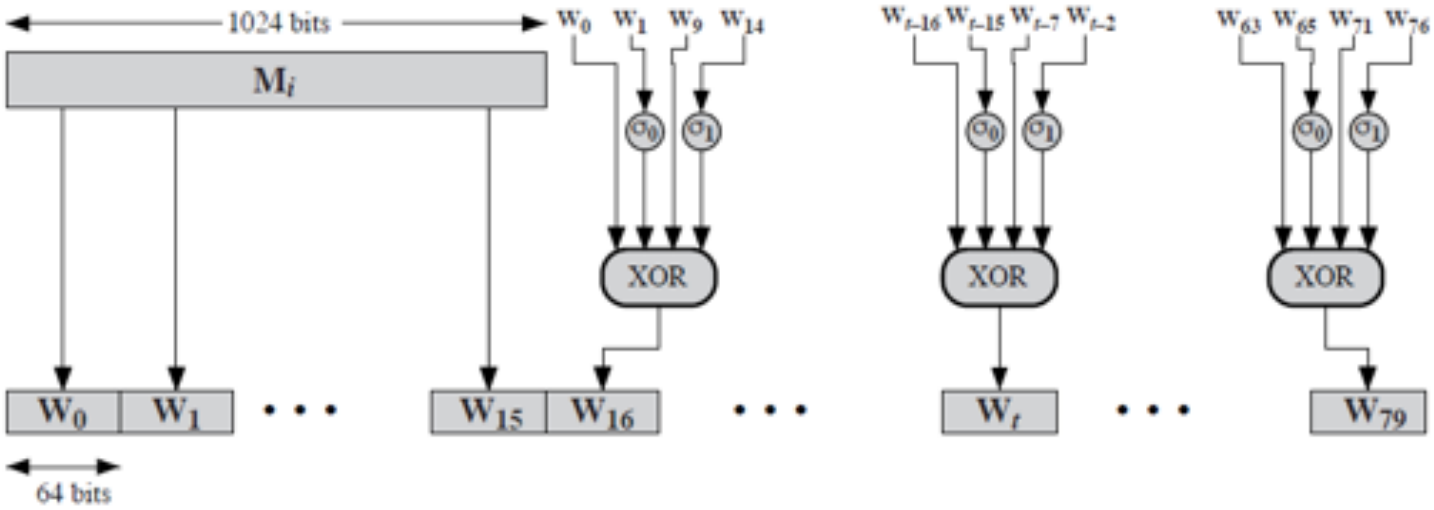
Estrutura do SHA-512



Estrutura do SHA-512



Estrutura do SHA-512



Message Authentication Codes (MAC)

- Assim como o hash, o MAC é um resumo criptográfico
  - Depende de chave: $MAC_k(m)=C(k,m)$ ou $H(k|m)$
- Aplicação típica:
  - Chave secreta k, compartilhada por entidades A e B
  - A envia m (mensagem) e MAC_k (m) (assinatura)
  - B recebe m, gera MAC_k (m), e compara com assinatura recebida

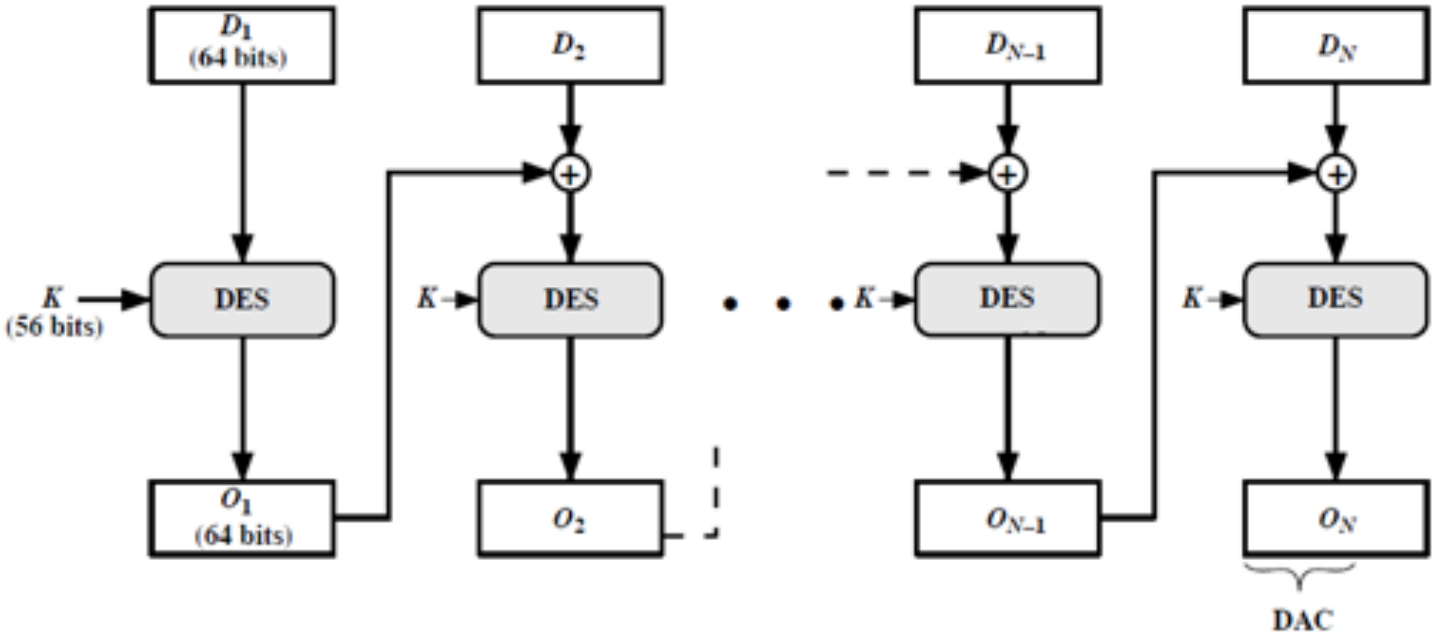
Requisitos de MAC

- Dados m e $MAC_k(m)$, é inviável encontrar m' tal que $MAC_k(m)=MAC_k(m')$
- MAC_k (m) deve ser distribuída uniformemente
  - $Pr[MAC_k(m)=MAC_k(m')]=2\text{-}n$
- Não-correlação
  - $Pr[MAC_k(m)=MAC_k(f(m))]= 2\text{-}n$
  - Para qualquer função conhecida (diferente da identidade)

MAC baseado no DES

- Data Authentication Algorithm (DAC)
  - FIPS PUB 113
  - ANSI X9.17
- Já substituído por novos algoritmos
- Cipher Block Chaining – CBC Mode
  - Mensagem em blocos de 64 bit: $D_1,\ldots,D_n$
    - $O_1=DES_k(D_1)$
    - $O_2=DES_k(D_2)$
    - ...
    - $O_n=DES_k(D_{n-1}) \equiv DAC$

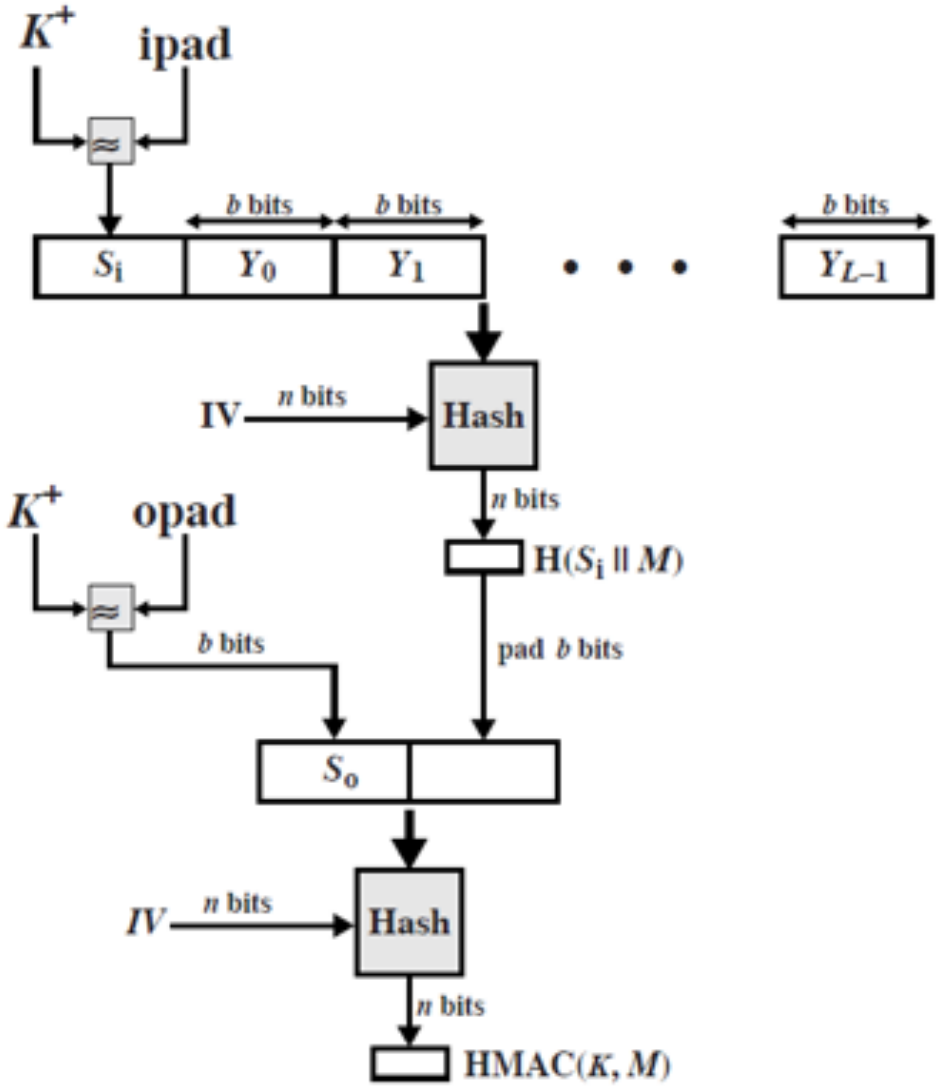
MAC baseado em uma cifra



MAC baseado em hash

- HMAC
  - RFP 2104
  - FIPS 198
  - Usado no SSL
- Objetivos do HMAC (RFC 2104)
  - Uso das funções hash disponíveis
  - Fácil subsituição da função hash, que estará "encapsulada"
  - Preservação da performance da função hash
  - Uso "simples" das chaves
  - Entendimento da "força" do mecanismo de autenticação dadas hipóteses acerca da função hash usada

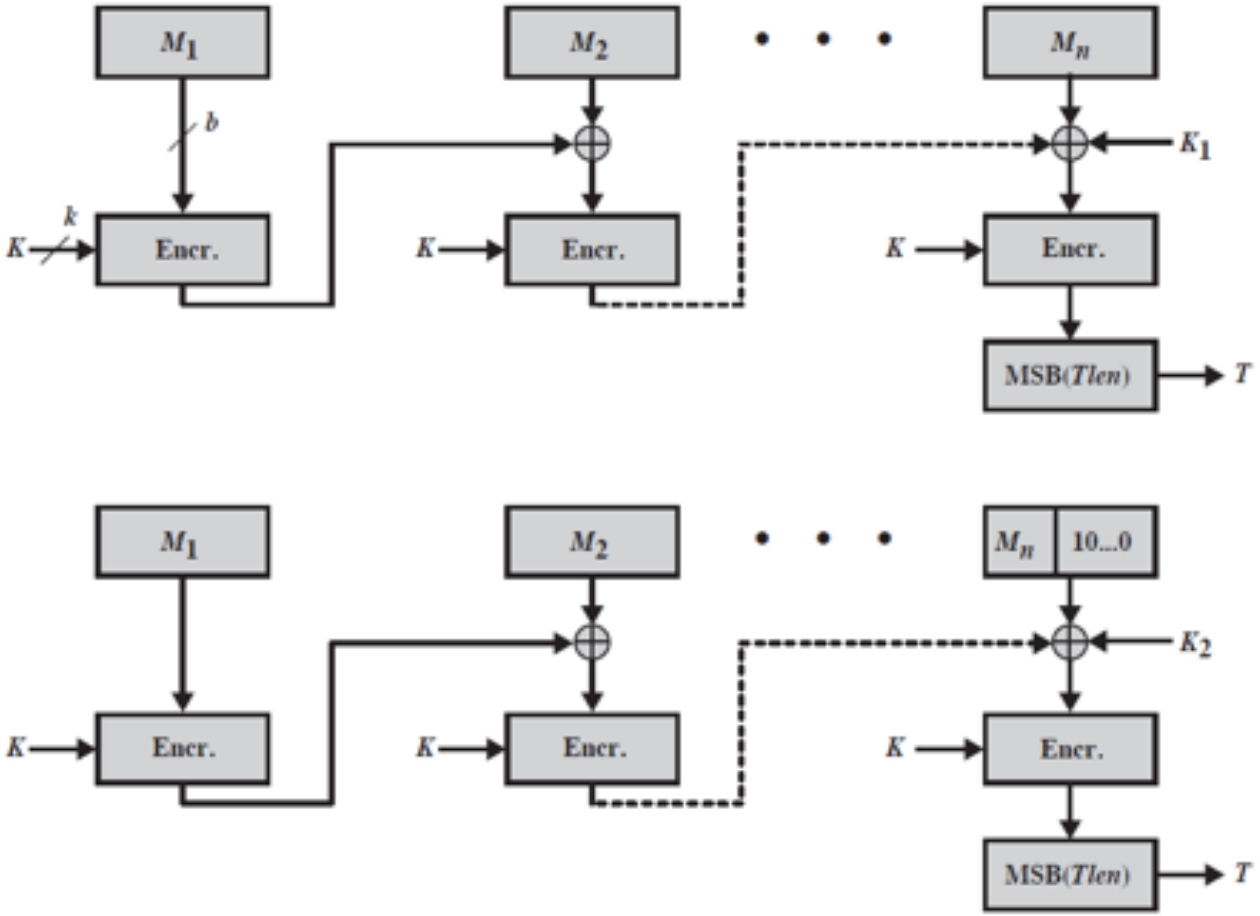
HMAC


K^+
ipad
b bits
b bits
b bits
S_i
Y_0
Y_1
Y_{L-1}
IV
n bits
Hash
K^+
opad
n bits
$H(S_i \| M)$
b bits
pad b bits
S_o
IV
n bits
Hash
n bits
HMAC(K, M)

CMAC

- O CBC-MAC (FIPS PUB 113), baseado no DES mostrou algumas fraquezas
  - Ex.: Dada mensagem x e MAC(k,x), sabemos construír uma mensagem diferente com mesmo MAC: x||x⊕MAC(k,x) (|| significa concatenação)
- Black e Rogaway: propõem do uso de três chaves
  - Uma com k bits, usada nos estágios do CBC
  - Uma com n bits (tamanho do bloco do cifrador)
- Iwata e Kurosawa: refinam o método
  - As duas chaves de n bits derivam da chave K de k bits
  - NIST Special Publication 800-38B

CMAC



Autenticação de mensagens

- Usando encriptação
  - Mensagem cifrada (inteira) funciona como autenticador
- Usando MAC
  - Chave secreta produz resumo criptográfico que é enviado juntamente com a mensagem
- Usando hash
  - Resumo criptográfico deve ser enviado de modo seguro

Assinaturas digitais

- Meio de associar identidade e informação
- Processo de assinatura de uma mensagem
  - usar alguma informação privada que, combinada com a informação, gera uma string – a assinatura

Assinaturas digitais

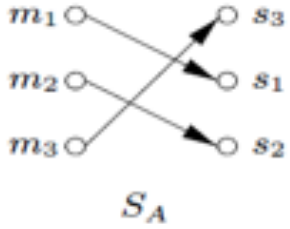
- Nomenclatura
  - $\mathcal{M}$ é o conjunto das mensagens que podem ser assinadas
  - $\mathcal{S}$ é o conjunto dos elementos chamados assinaturas, em geral strings de tamanho fixo
  - S_A é a transformação de assinatura da entidade A
    - leva uma mensagem a uma assinatura (deve ser mantida secreta por A)
  - V_A é a transformação de verificação de assinaturas de A
    - verifica se uma assinatura de mensagem foi gerada pela entidade A (divulgada publicamente)
- Os conjuntos $\mathcal{M}$, $\mathcal{S}$ e as transformações S_A e V_A provêem um *esquema de assinatura digital* (ou *mecanismo de assinatura digital*) para A.

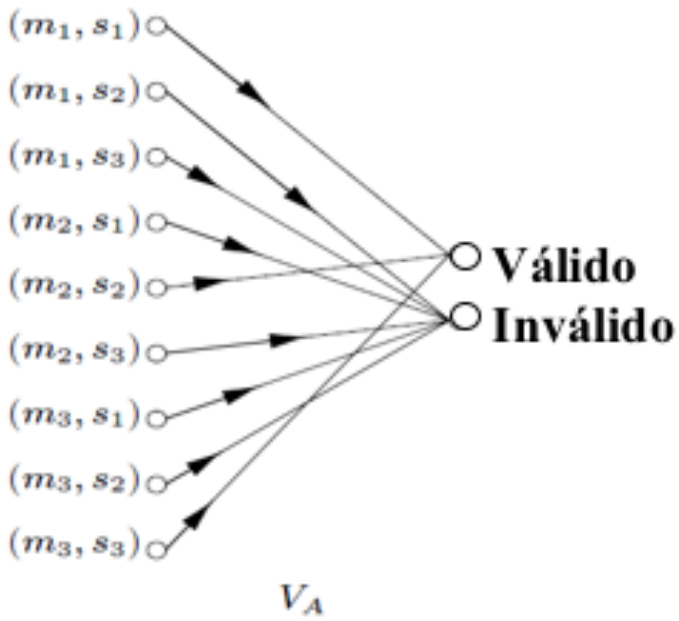
Funcionamento do mecanismo de assinatura

- Procedimento de assinatura
  - Entidade *A* (assinadora) cria assinatura para mensagem m da seguinte forma:
    - Computa $s=S_A(m)$
    - Transmite o par (m,s) – s é a assinatura de *A* para *m*
- Procedimento de verificação
  - Para verificar se a assinatura s na mensagem m foi realmente criada por *A*, o receptor procede da seguinte forma
    - Obtém a transformação de verificação V_A de *A*
    - Computa $u=V_A(m,s)$
    - Aceita a mensagem se *u*=válido e rejeita se *u*=inválido

Exemplo

- $M=\{m_1,m_2,m_3\}$ e $S=\{s_1,s_2,s_3\}$. Na figura da esquerda mostramos uma transformação de assinatura S_A. Na direita, mostramos a transformação de verificação V_A correspondente.





Propriedades de esquemas de assinatura

- s será uma assinatura válida de A para a mensagem m se e só se $V_A(m,s)$=válido.
- É computacionalmente inviável para qualquer outra entidade além A encontrar, para qualquer mensagem m em $\mathcal{M}$, uma assinatura s em $\mathcal{S}$ tal que $V_A(m,s)$=válido
- O processo de verificação de assinatura é computacionalmente eficiente

Autenticação versus assinatura

- Autenticação garante a origem de uma mensagem
  - Se eu recebo mensagem com autenticador, sei que apenas o detentor da chave secreta pode ter enviado
- Autenticação não proporciona irrefutabilidade
  - Emissor alega não ter enviado mensagem
    - Receptor poderia ter forjado MAC, já que compartilha da chave secreta
- Assinatura
  - Caso a assinatura do emissor seja verificada, ele não pode alegar não ter enviado
    - Apenas ele poderia gerar uma assinatura válida

Assinatura digital a partir de cifra de chave pública

- Esquema reversível de chave pública
  - Espaço das mensagens planas = espaço das mensagens cifrada
  - Então $D_d(E_e(m))=E_e(D_d(m))=m$ para toda mensagem m em $\mathcal{M}$
- Construção do esquema de assinatura digital
  - M é o espaço das mensagens do esquema de assinatura digital
  - C é o espaço das assinaturas do esquema de assinatura digital
  - A transformação de assinatura é $S_A:=D_d$
  - A transformação de verificação é
    - $V_A(m,s)$ = válido, se $E_e(s)=m$
      inválido, caso contrário

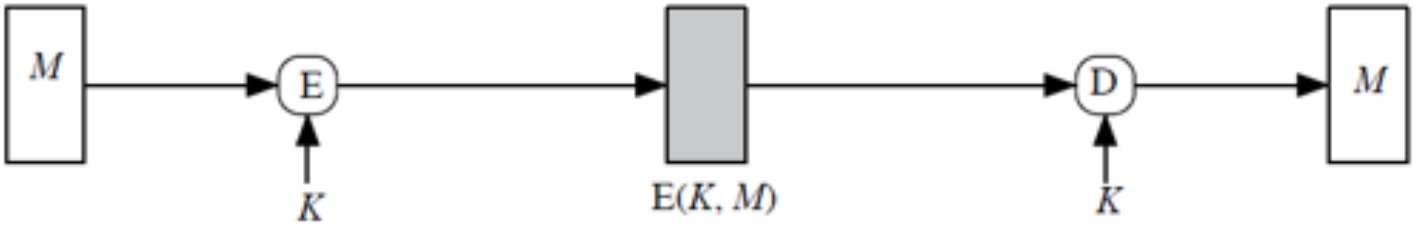
Digital Signature Standard (DSS)

- Padrão NIST – FIPS 186
  - Usa Secure Hash Algorithm
  - Apresenta nova técnica de assinatura digital
  - Proposto em 199, revisado em 1993
  - Pequenas alterações em 1996
- Nova versão em 2000 – FIPS 186-2
  - Incorpora algoritmos baseados em RSA e curvas elípticas
- FIPS 186-3: junho de 2009
- FIPS 186-4: julho de 2013

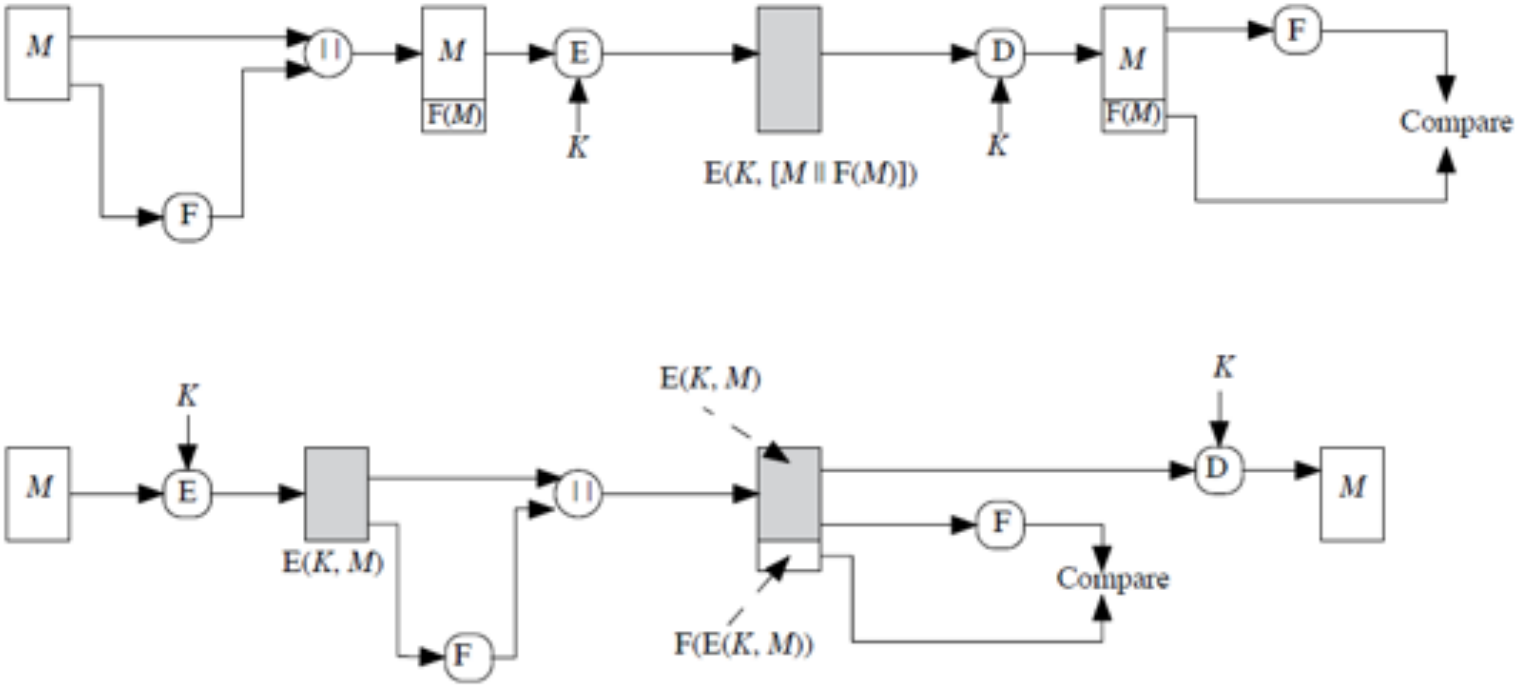
Protocolos básicos

Cifra de chave simétrica

› Confidencialidade (e autenticação)

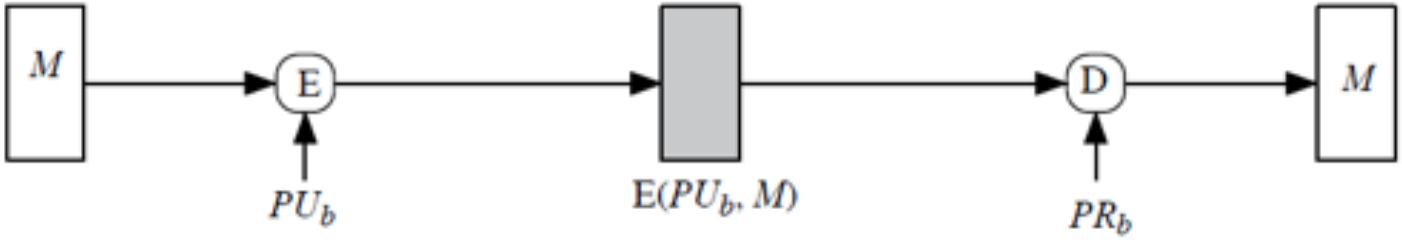


Controle de erros (interno/externo)



Cifra de chave pública

› Confidencialidade

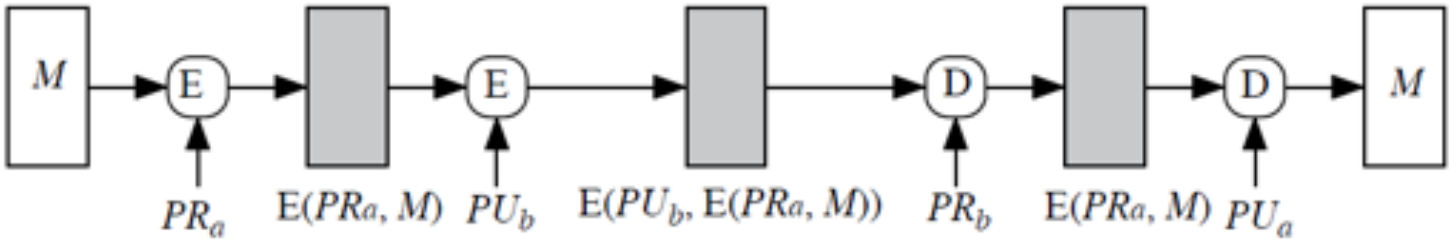


Cifra de chave pública

› Autenticação e assinatura

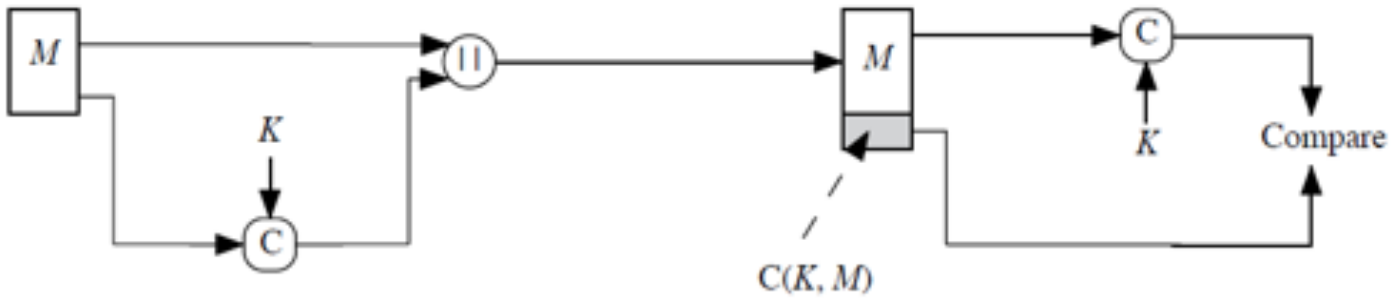
Cifra de chave pública

› Confidencialidade, autenticação e assinatura



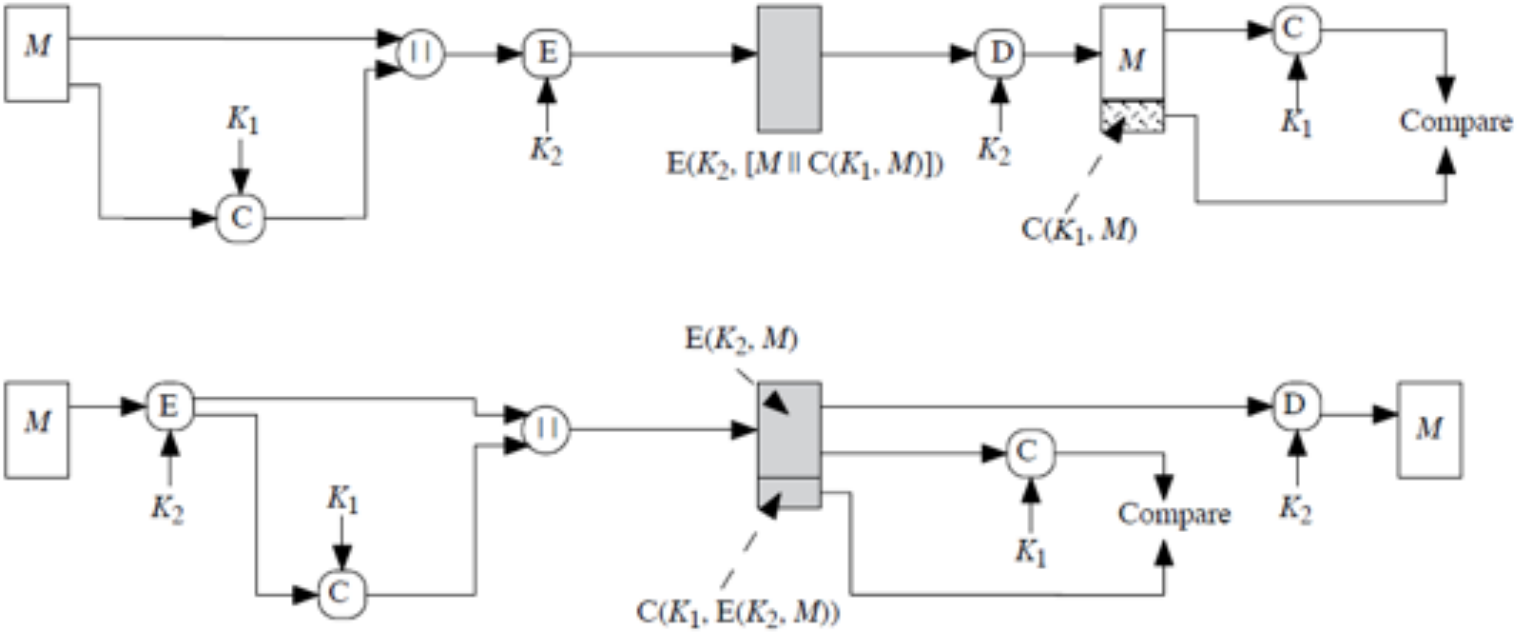
MAC

› Autenticação



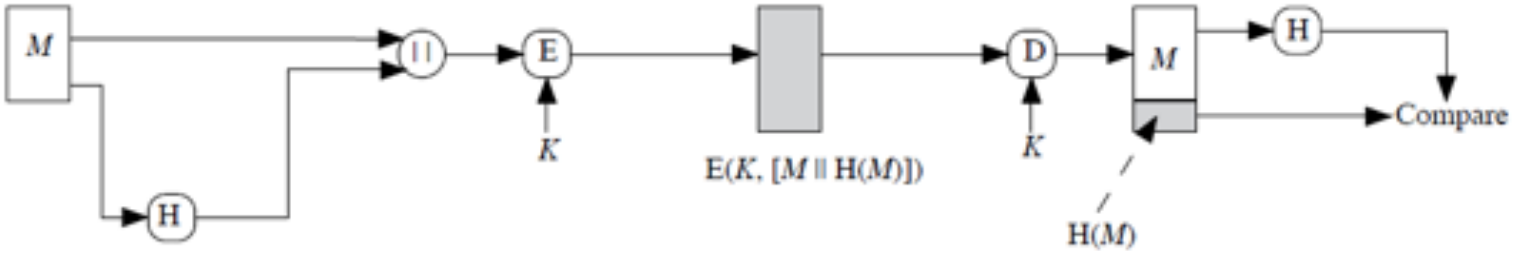
MAC

› Autenticação e confidencialidade



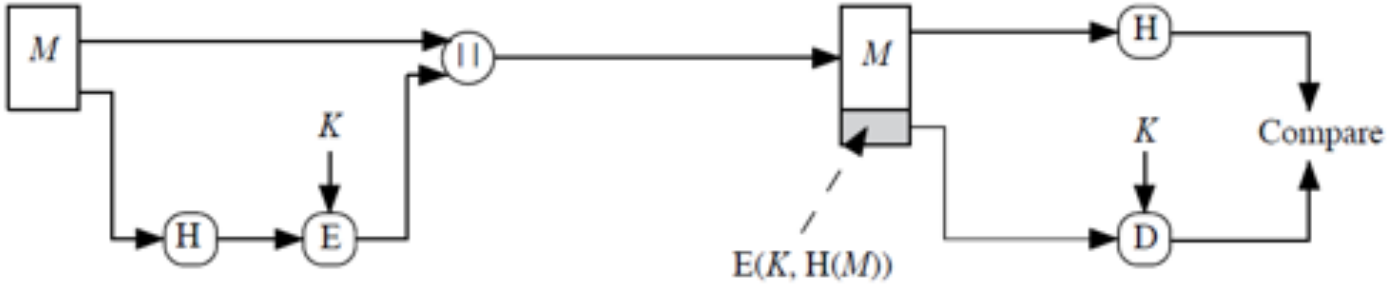
Hash

› Autenticação e confidencialidade



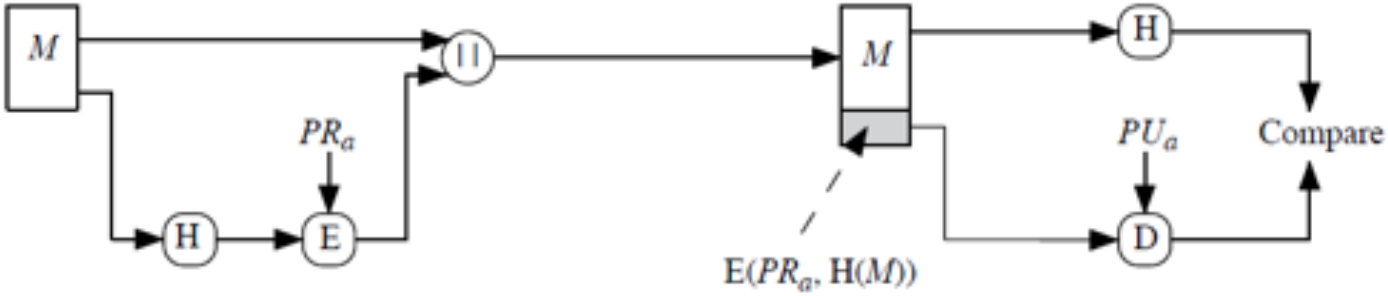
Hash

› Integridade



Hash

› Autenticação e assinatura

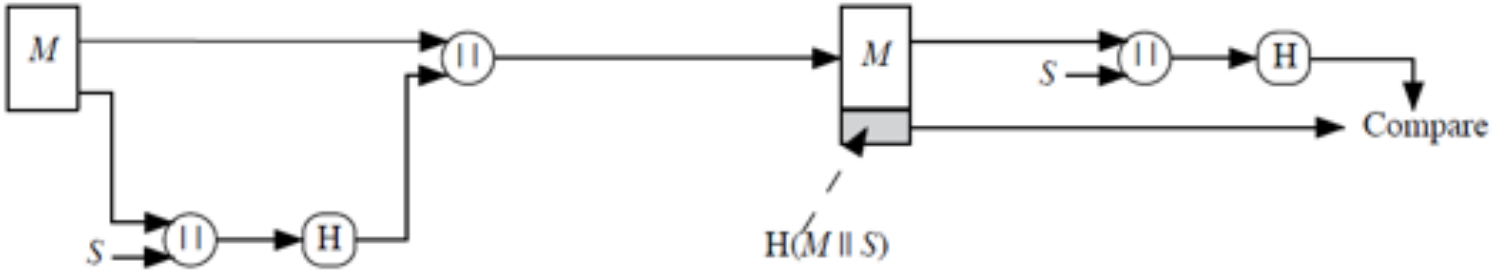


Hash

› Autenticação, assinatura e confidencialidade

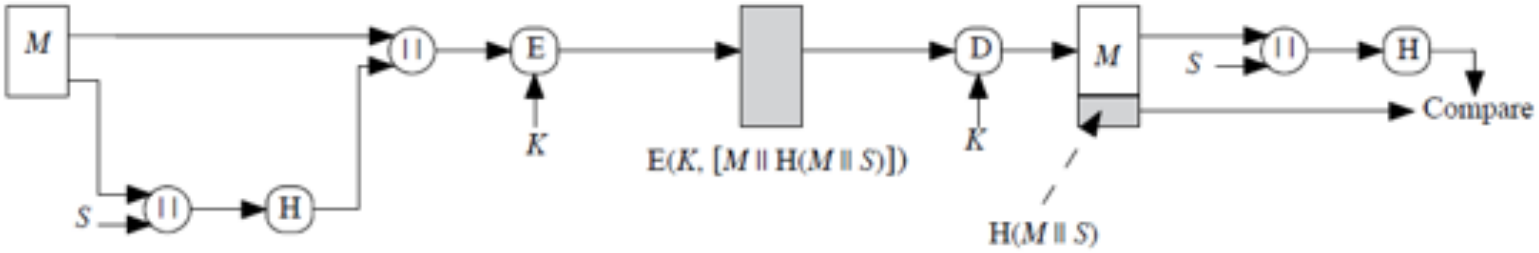
Hash

› Autenticação baseada em “segredo”

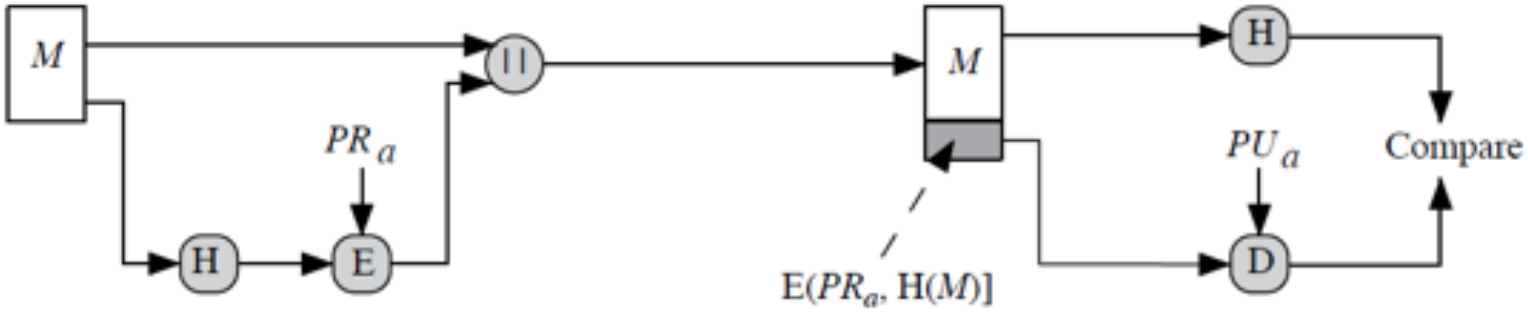


Hash

› Autenticação baseada em "segredo" e confidencialidade



Assinatura baseada em hash + cifra assimétrica



›Protocolos criptográficos; gerenciamento de chaves

Fundamentos de Criptografia

›Protocolos criptográficos

Protocolo criptográfico

- Algoritmo distribuído definido por seqüência de passos que especificam ações a serem tomadas por duas ou mais entidades para alcançarem um objetivo de segurança
- Protocolo exercem papel central em criptografia, sendo essenciais para se alcançar os objetivos de segurança
- Esquemas de encriptação, assinaturas digitais, funções hash e geradores de números aleatórios são primitivas que podem ser utilizadas para se construir um protocolo criptográfico.

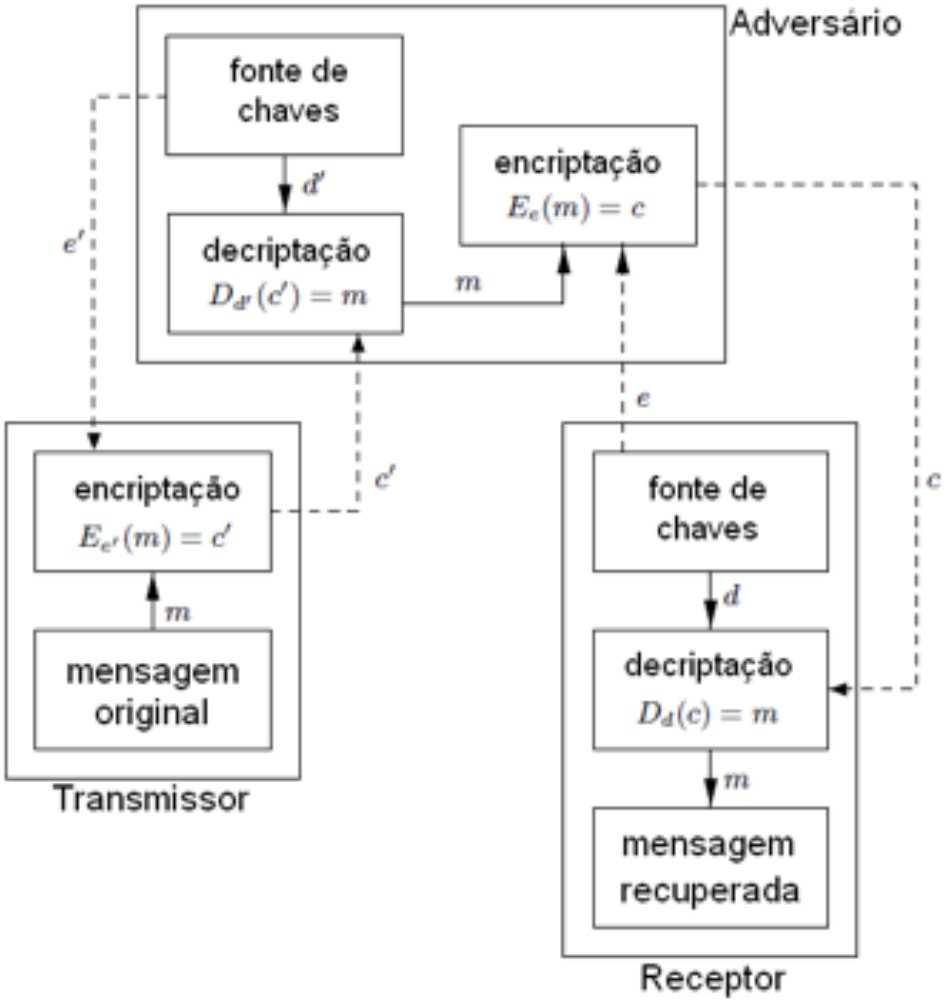
Exemplo de protocolo criptográfico

- Esquema de criptografia de chave simétrica para comunicação em canal inseguro
  - Bob constrói esquema de criptografia de chave pública e envia a Alice sua chave pública através do canal inseguro
  - Alice gera chave secreta para um esquema de criptografia de chave simétrica
  - Alice encripta a chave secreta usando a chave pública de Bob
  - Bob decripta a mensagem de Alice e obtém a chave secreta
  - Alice e Bob passam a comunicar-se usando a chave secreta.
- As primitivas básicas utilizadas são os esquemas de criptografia de chave pública e de chave secreta.

Ataque ao protocolo anterior: impersonation/personificação

- Eva "personifica" Bob enviando sua chave pública a Alice
- Alice assume (incorretamente) ter a chave pública de Bob e envia uma chave secreta para Eva
- Eva passa a interceptar mensagens cifradas de Alice para Bob e decriptá-las
  - Em seguida, re-encripta a mensagem com a chave pública de Bob e para ele envia a mensagem cifrada correspondente

Impersonation (MitM)



Falha de protocolo/mecanismo

- Ocorre quando o protocolo/mecanismo falha em atingir seus objetivos de segurança.
- O adversário obtém vantagem não pela quebra das primitivas criptográficas, mas pela manipulação do protocolo/mecanismo.

Falha de mecanismo: outro exemplo

- Alice e Bob comunicam-se usando cifra de stream (encriptação bit a bit)
- As mensagens encriptadas tem o seguinte formato:
  - Os primeiros vinte bits carregam informação monetária (encriptada)
- Um adversário ativo pode simplesmente modificar esses primeiros vinte bits
- O adversário não foi capaz de ler a informação, mas pôde alterá-la
- O problema foi que se assumiu incorretamente que a criptografia proveria garantia de integridade

Forward search

- Em uma transação bancária, 32 bits do campo "valor da transação" são encriptados de forma a prover confidencialidade.
- Entretanto, o protocolo falha no seu objetivo
  - O espaço de mensagens planas é pequeno: 232 mensagens.
  - O adversário pode encriptar cada uma delas (a chave de encriptação é pública) e comparar com a mensagem cifrada.
- O esquema de encriptação de chave pública não foi comprometido
  - A chave não foi descoberta
  - Entretanto, a forma como o esquema foi usado permitiu descobrir a mensagem plana

Causas de falhas de protocolo

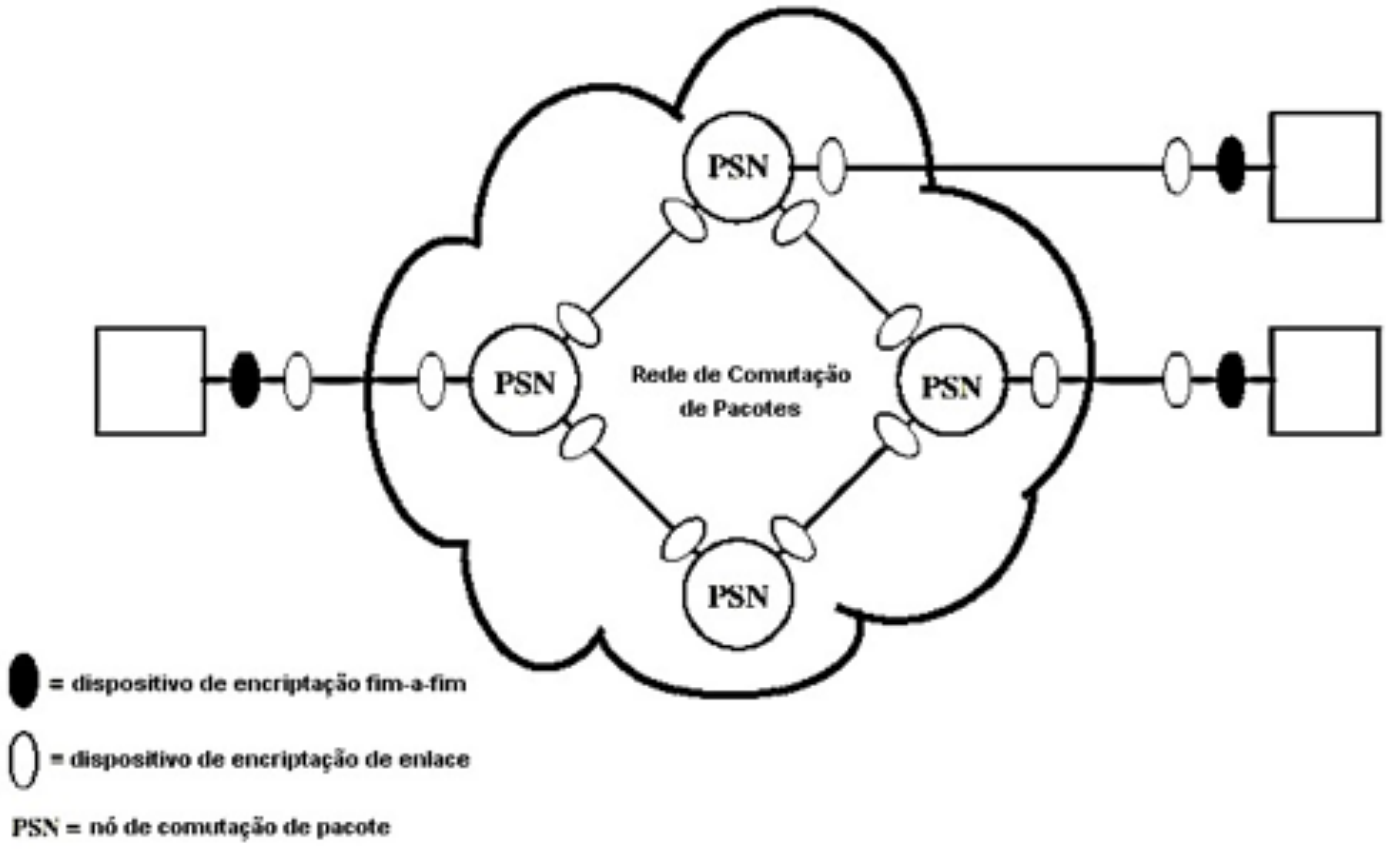
- Fraquezas de determinada primitiva criptográfica podem ser "ampliadas" por um protocolo ou mecanismo inconveniente
- Incorreto entendimento de algum princípio associado a determinada primitiva criptográfica

- Projeto de protocolos
  - 1. Identifique todas as hipóteses utilizadas no projeto de protocolo ou mecanismo; e
  - 2. para cada hipótese, determine o efeito nos objetivos de segurança, caso a hipótese seja violada.

O “local” dos dispositivos criptográficos

- Criptografia de enlace (link de dados)
- Criptografia pode ser usada em diversas camadas de um protocolo de comunicação
- Necessita de diversos dispositivos criptográficos
  - Encripta cabeçalhos de camadas superiores
- Criptografia fim-a-fim
  - O transmissor encripta e o receptor decripta
  - As informações de cabeçalho passam em claro
- Em situações em que é necessária alta segurança, ambas as formas devem ser usadas

O “local” dos dispositivos criptográficos



› Gerenciamento de chaves

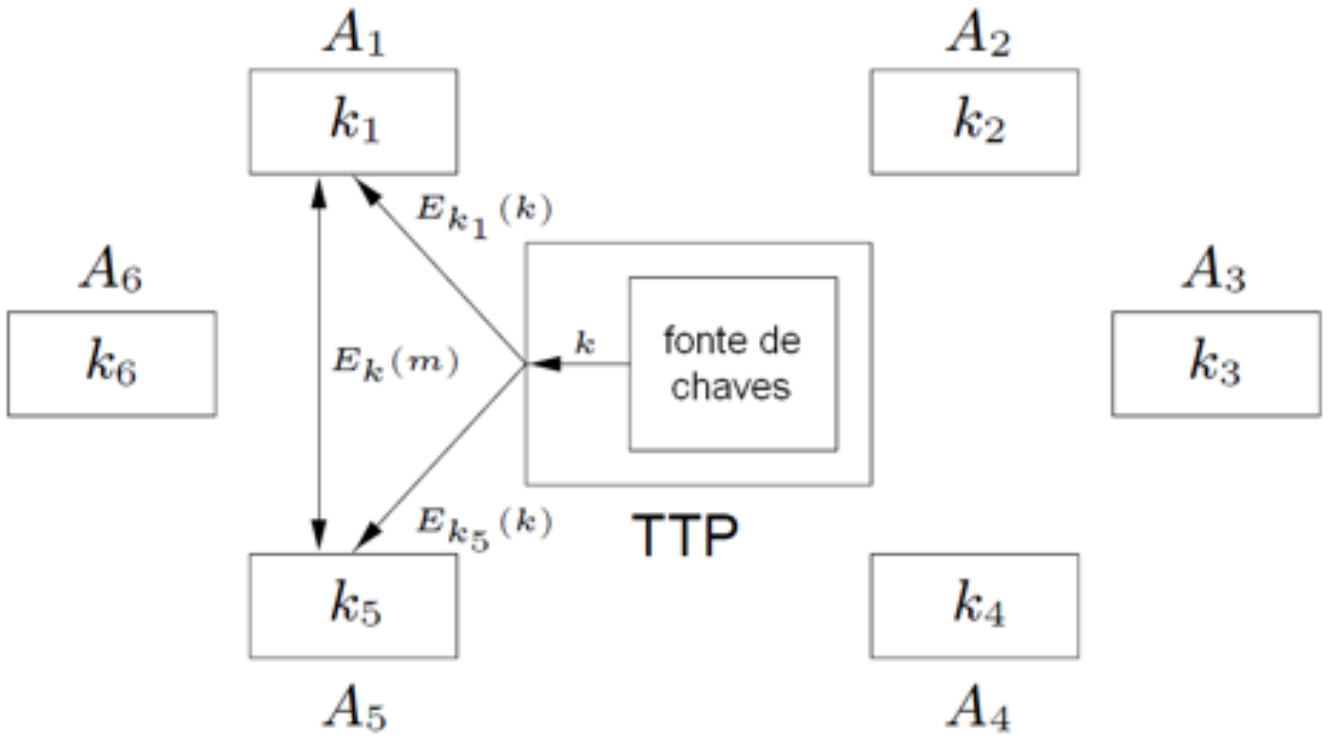
Gerenciamento de Chaves

- Objetivo: distribuição segura de chaves criptográficas
- Estabelecimento de chave: processo pelo qual uma chave se torna disponível para uso criptográfico
- Gerenciamento de chave: conjunto de processos que apóia o estabelecimento e a manutenção de chaves
  - Inclui revogação e substituição de chaves

Gerenciamento de chaves em ambientes de chave simétrica

- Trusted third party (TTP): entidade que tem a confiança de todas as outras entidades
  - Cada entidade Ai compartilha uma chave secreta ki com o TTP
  - Assume-se que estas chaves foram distribuídas através de canal seguro
- Se duas entidades desejam comunicar-se:
  - O TTP gera uma chave (chave de sessão) e as envia encriptadas (através das chaves secretas fixas) a estas entidades

Gerenciamento de chaves com TTP



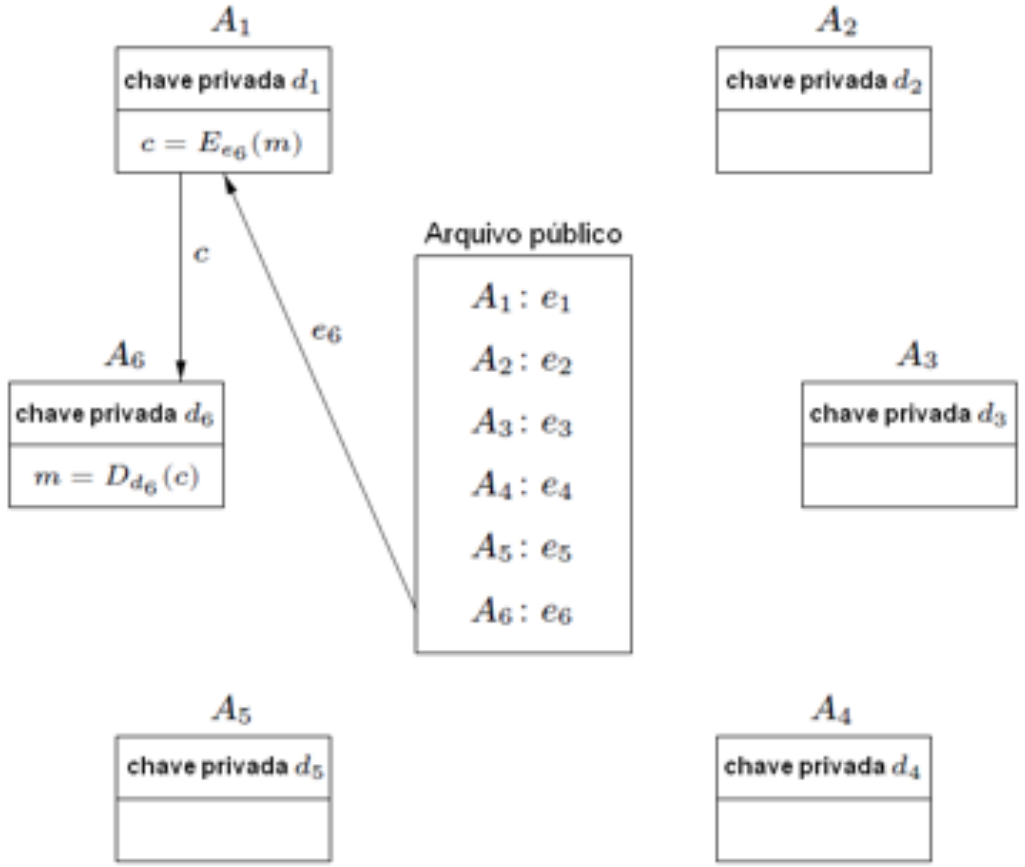
Gerenciamento de chaves com TTP

- Vantagens
  - É fácil adicionar e remover entidades da rede
  - Cada entidade precisa armazenar apenas uma chave secreta de longa duração
- Desvantagens
  - Todas as comunicações exigem interação inicial com o TTP
  - O TTP deve armazenar n chaves secretas de longa duração
  - O TTP tem a possibilidade de ler todas as mensagens
  - Se o TTP é comprometido, todas as comunicações são inseguras

Gerenciamento de chaves em ambientes de chave pública

- Arquivo público (public file): repositório central de chaves
- Suponha que uma entidade A1 quer se comunicar com uma entidade A6
  - A1 obtém a chave pública e6 de A6 no arquivo público
  - A1 encripta a mensagem usando e6, e
  - Envia a mensagem cifrada a A6

Gerenciamento de chaves em ambientes de chave pública

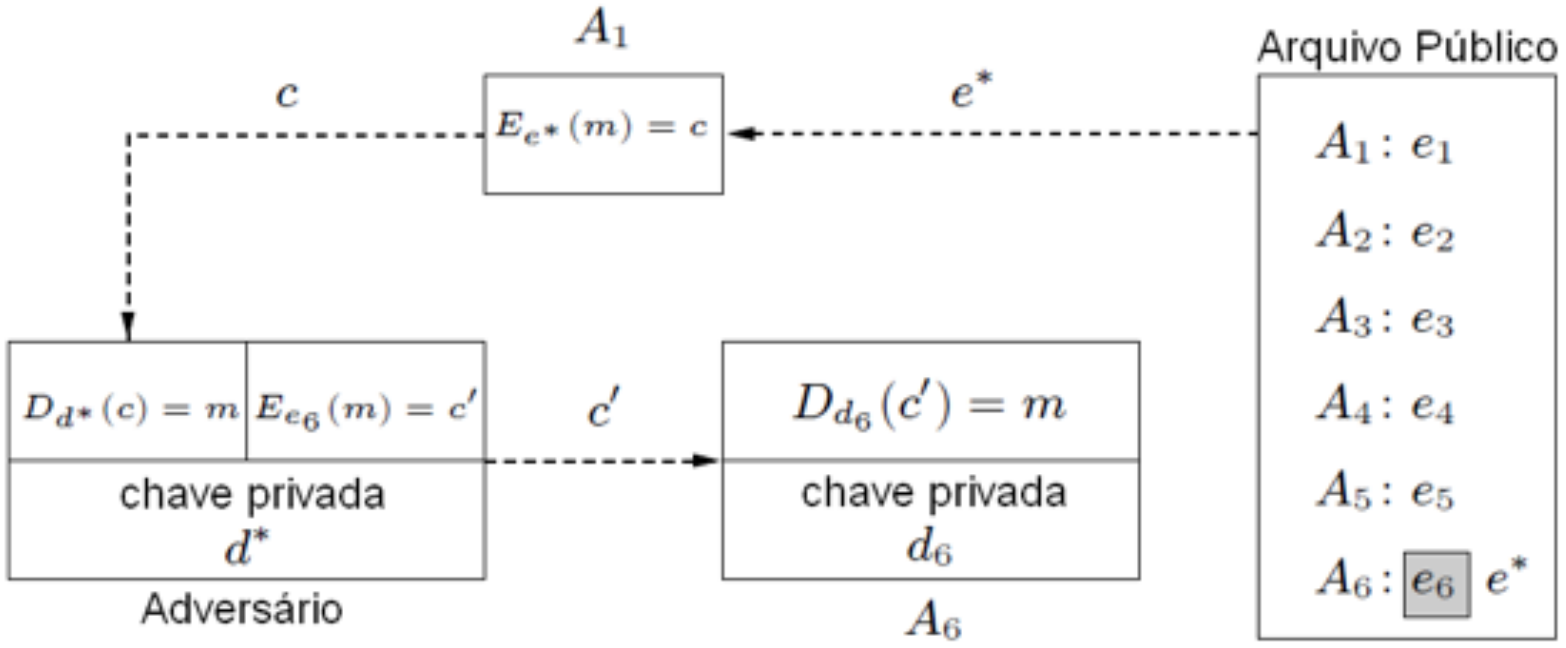


Gerenciamento de chaves em ambientes de chave pública

- Vantagens
  - Não é necessário um TTP "todo-poderoso".
  - O arquivo público pode ser replicado em cada entidade
  - Apenas n chaves públicas precisam ser armazenadas
    - Assumindo que apenas ataques passivos são possíveis

Adversário ativo

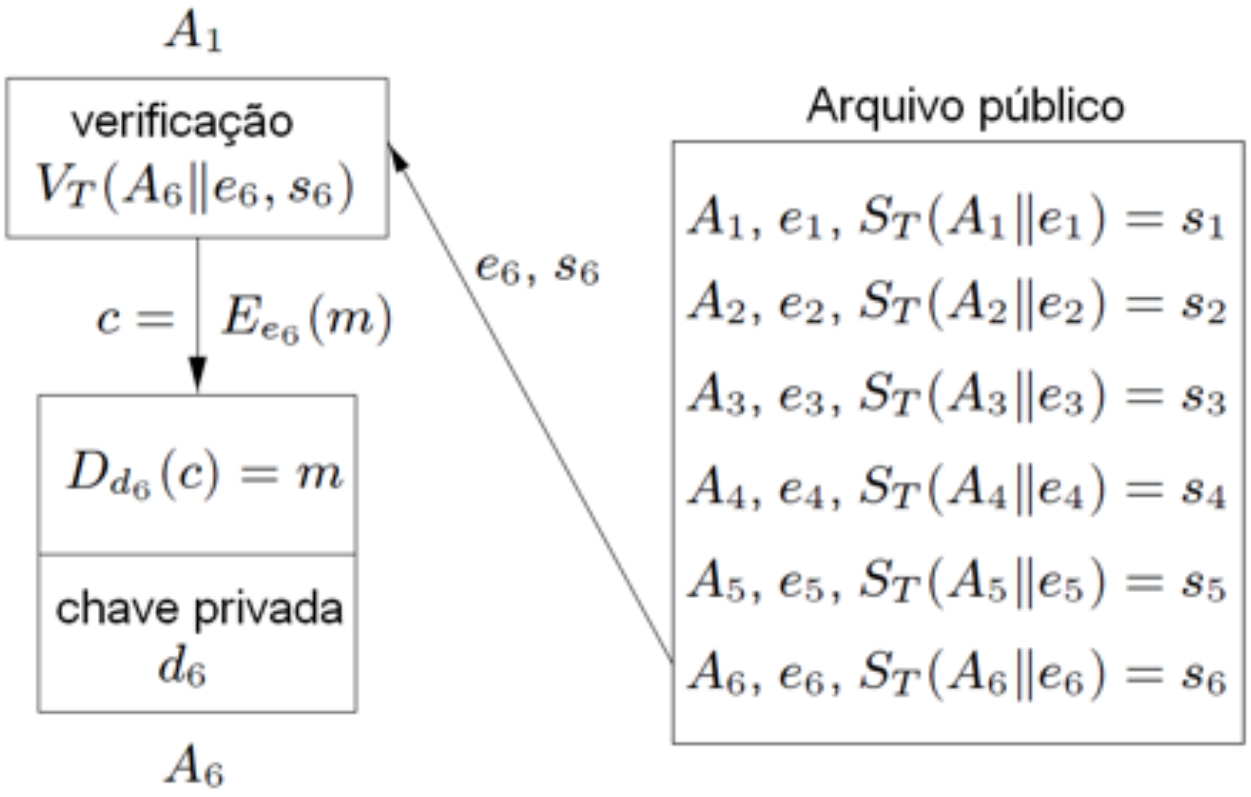
› Adversário pode alterar o arquivo público



Uso de TTP como certificador

- Entidades podem usar um TTP para certificar a chave pública de cada entidade
- O TTP possui um algoritmo privado de assinatura ST e divulga um algoritmo de verificação VT
- O TTP verifica "cuidadosamente" a identidade de cada entidade
- O TTP assina uma mensagem que consiste no identificador da entidade e sua chave pública
- Este é um exemplo simples de certificado, associando a identidade de uma entidade à sua chave pública

Autenticação com TTP



Funcionamento do certificado de chave-pública

- Para verificar a autenticidade da chave pública de uma entidade A, a entidade B deve possuir uma cópia autêntica da função de verificação de assinatura do TTP
  - Assuma que essa função é fornecida a B diretamente pelo TTP
- B executa os seguintes passos
  - 1. Obtém o certificado de A (de uma base de dados ou diretamente de A)
  - 2. Usa a função de verificação do TTP para verificar a assinatura do TTP no certificado de A
  - 3. Se a assinatura é verificada verdadeira, aceita a chave pública contida no certificado, caso contrário, rejeita

Autenticação com TTP

- Vantagens de usar um TTP para manter a integridade do arquivo público
  - Previne a possibilidade de impersonation por um adversário ativo.
  - O TTP não pode monitorar a comunicação
    - as entidades confiam no TTP apenas para associar identidades a chaves públicas
  - A interação com o arquivo público pode ser reduzida se as entidades armazenarem certificados localmente
- Alguns problemas permanecem
  - Se o algoritmo de assinatura do TTP é comprometido, toda comunicação se torna insegura
  - Toda confiança é depositada em um único lugar

Grau de confiança no TTP

- TTP incondicionalmente confiável (unconditionally trusted)
  - Confiável em todos os sentidos
    - Acesso a chaves secretas
    - Acesso a chaves privadas
    - Encarregado da associação entre chaves públicas e identificadores
- TTP funcionalmente confiável (functionally trusted)
  - Entidade é considerada honesta
  - No entanto, não tem acesso a chaves secretas ou privadas

Lição: você precisa de canais confiável

- Em qualquer esquema de comunicação, por mais avançado que seja, precisaremos, em algum momento, de um canal seguro:
  - Apresentação de documentação pessoal
  - Verificação de dados biométricos
  - Contato “em pessoa”
- As técnicas de criptografia permitem que, uma vez que se tenha estabelecido uma comunicação confiável, todas as comunicações posteriores estarão asseguradas
  - A segurança passa a residir na chave que fora trocada por um canal seguro

Lição: não existe canal totalmente confiável

- Utilize canais de confiabilidade adequada à sua necessidade de segurança
- A segurança acerca da identidade de uma pessoa ou entidade é, no máximo, a segurança estabelecida no canal mais confiável já estabelecido
- A criptografia é um instrumento para perpetuar a confiança estabelecida neste canal mais confiável
- Veja um estudo de caso “informal” no próximo slide