Criptografia

Histórico, Técnicas e Aplicações

Tópicos Históricos de Criptografia

Cifras Antigas - Egito

- Tumba de Khnumhotep II
  - Substituição deliberada de alguns símbolos
  - Transformação da escrita com diversos possíveis objetivos
    - Ex.: criar aura de mistério
  - Mais antigo texto conhecido contendo modificação deliberada de linguagem

Cifras Antigas – Esparta – Scytale (Cítala)

- Cifrador espartano de transposição (400AC)
- Fita de papel era enrolada em uma vareta
- Mensagem escrita enquanto a fita está enrolada; depois o papel é removido, ficando a fita com uma seqüência de letras aparentemente aleatória
- A chave é definida pela circunferência da vareta e a espessura do papel

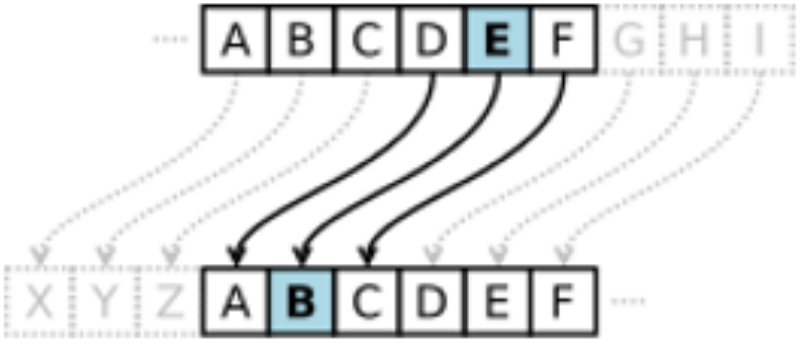
	A	J	U	D	E	
	M	E	S	T	O	
	U	S	O	B	A	
	T	A	Q	U	E	

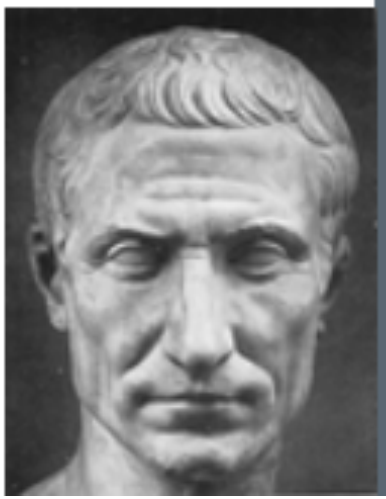
Cifras Antigas – Persas

- Heródoto registra o uso de esteganografia pelos persas (400 AC)
  - Damaratus avisa gregos sobre Xerxes (cera sobre madeira)
  - Histaiaeus envia mensagem a Aristagoras de Miletus (tatuagem no couro cabeludo)

Cifras Antigas – Roma

- Cifras antigas
  - Cifra de substituição supostamente utilizada por Júlio César (100 AC)
  - Basicamente, um "deslocamento" do alfabeto





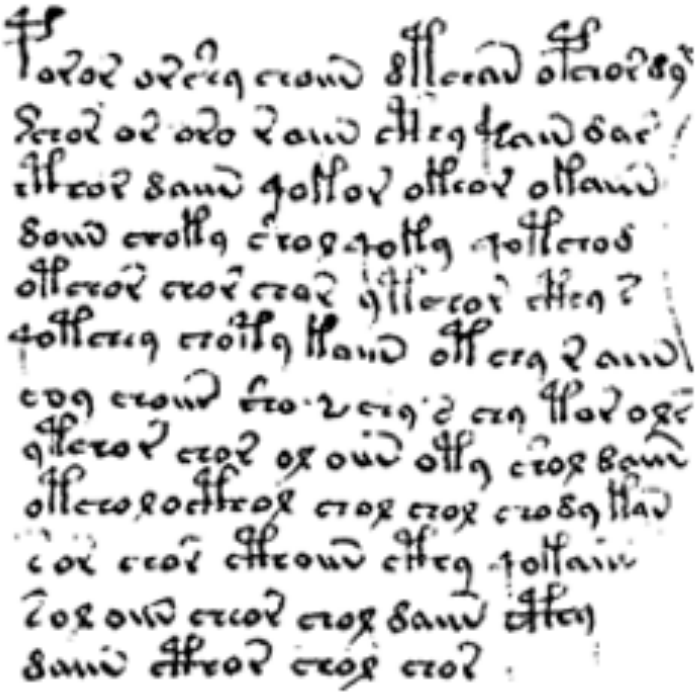
Cifras Antigas – Al-Kindi

- Al-Kindi estuda estatísticas das linguagens e desenvolve primeiras técnicas de criptanálise (séc. IX)
- O "tratado" de criptografia foi redescoberto em 1987, em Istambul
  - Chama-se "Manuscrito sobre 'deciframento' de mensagens criptografadas"

Ahmad al-Qalqashandi

- Ahmad al-Qalqashandi (1355-1418 DC)
- Escreve enciclopédia (Subh al-a 'sha, 14 volumes) com seção dedicada à criptografia (1412 DC)

Manuscrito de Voynich (1450-1520 DC)



Cifra Leon Alberti (1466 DC)

Vigenère

- Livro de Vigenère sobre cifras (1585)
  - Traicte de Chiffres
  - Idéia baseada em Giovan Batista Belaso

Cilindro de Jefferson

Cilindro de Jefferson

- Desenvolvido em 1795
  - 26 discos, cada um com uma ordem aleatória do alfabeto
  - Os discos podem ser reordenados
    - Emissor e receptor "combinam" uma ordenação
  - Emissor gira os discos de maneira a formar a mensagem plana em uma linha
    - Ele transmite a seqüência de caracteres de outra linha
  - Receptor gira os discos de maneira a formar a mensagem cifrada em uma linha
    - A mensagem plana irá aparecer em outra linha

Disco de Wheatstone

- Disco de Wheatstone
  - Originalmente inventado por Wadsworth em 1817
  - Desenvolvido por Wheatstone em 1860
  - Dois discos concêntricos geral uma cifra polialfabética

Urkryptografen: versão do disco de Wheatstone usado pelo exército dinamarquês de 1936 a 1948

Desiderata de Kerchhoff

› Kerchhoff e as leis da criptografia (1883)

JOURNAL

DES

SCIENCES MILITAIRES

DES

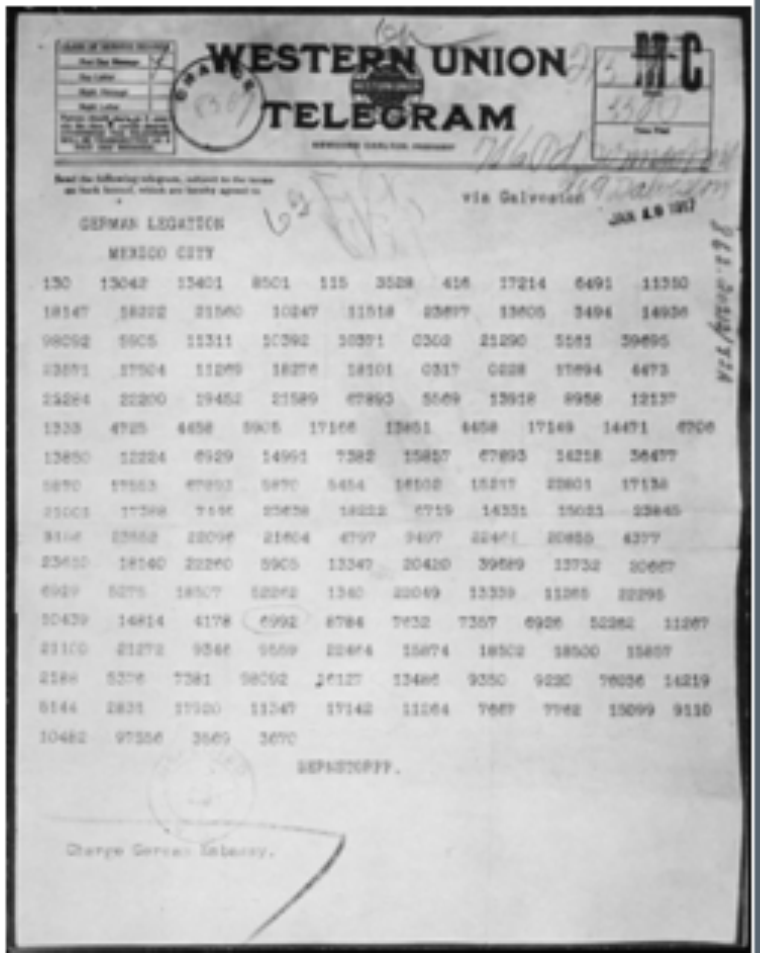
ARMÉES DE TERRE ET DE MER,

PUBLIÉ

SUR LES DOCUMENTS FOURNIS PAR LES OFFICIERS DES ARMÉES FRANÇAISES ET ÉTRANGÈRES,

Telegrama de Zimmerman

› Entrada dos EUA na I Guerra



WESTERN UNION TELEGRAM

via Galveston

GERMAN LEGATION

MEXICO CITY

Enigma

- Importante classe de máquinas cifradoras
- Bastante utilizada durante a Segunda Guerra Mundial
- Discos contendo conexões internas gerando substituições com alfabetos modificados continuamente

Criptografia ”at a glance”

- Ferramentas básicas
  - Esquema geral de criptografia
  - Cifra simétrica/secreta (bloco e fluxo)
  - Hash (função unidirecional/one-way)
  - Autenticação de mensagem (HMAC e CMAC)
  - Cifra de assimétrica/pública
  - Assinatura digital
- Aplicações ”simples”
  - Certificado Digital
  - Troca de chave / envelope de mensagem
- TLS, DH vs RSA, FPS
- Números aleatórios
- Ataques e modelos de segurança (visão geral)

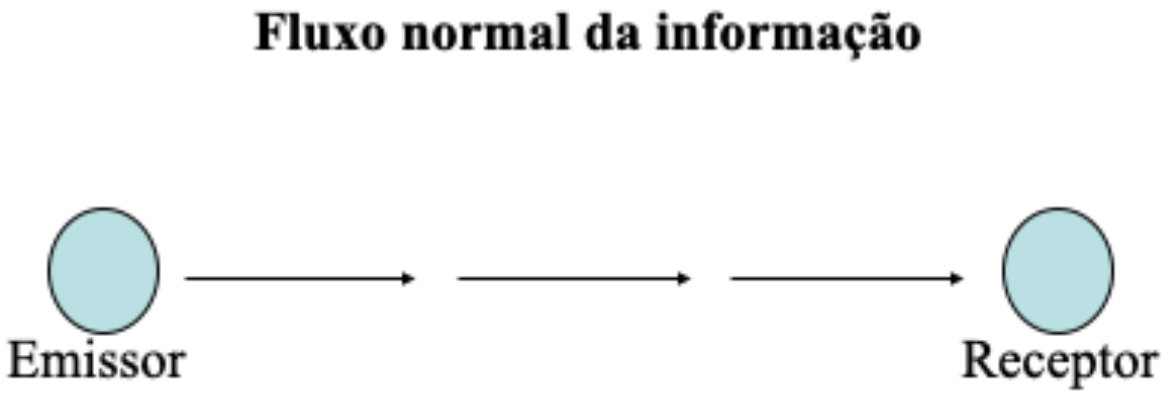
Criptografia

Apresentação

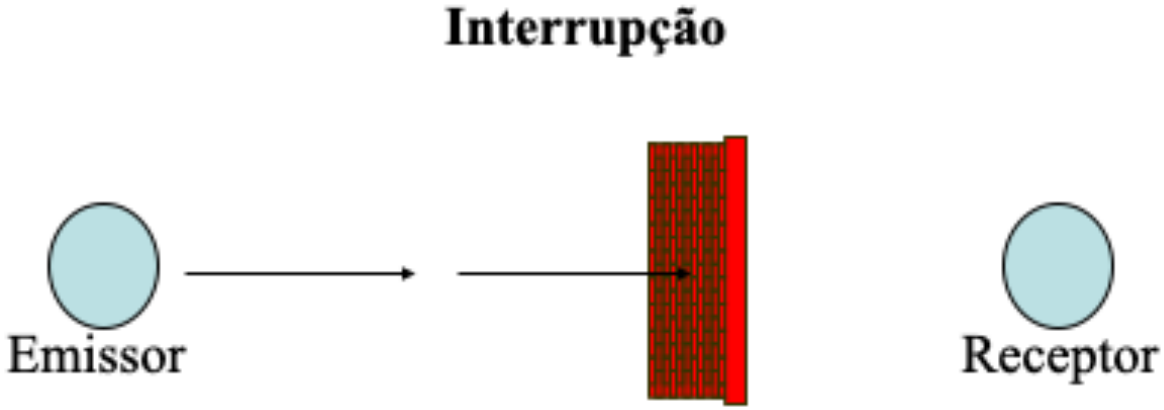
Por que estudar criptografia

- Ferramenta fundamental para atingir objetivos (prover serviços) de segurança
  - Importante para compreender soluções reais de segurança
- Modelos de funcionamento e de ataques simples e bem-caracterizados
  - Bom ponto de partida para entender arquiteturas de segurança
  - Sempre ter em mente que o mundo real é mais complexo do que os diagramas simplificados que veremos a seguir – lembre dos diversos ataques até agora estudados

Modelo de comunicação ”segura”

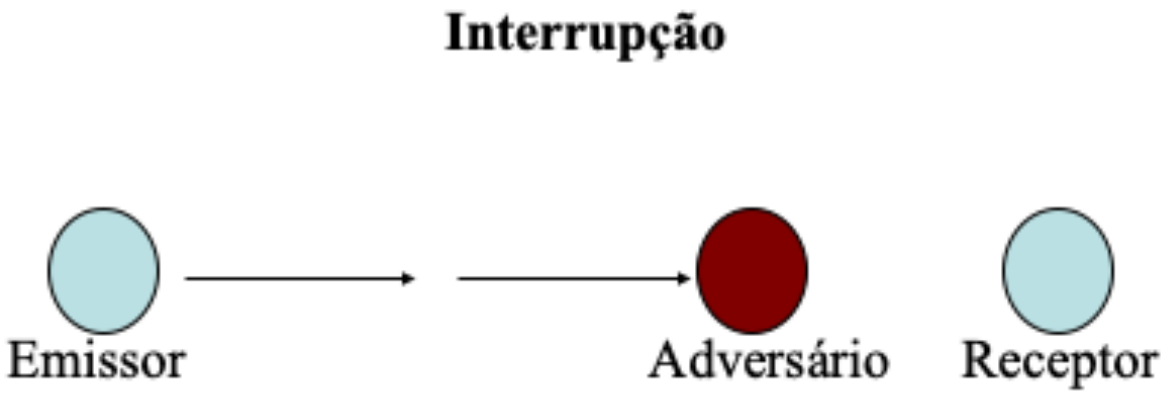


Ataques à segurança: disponibilidade



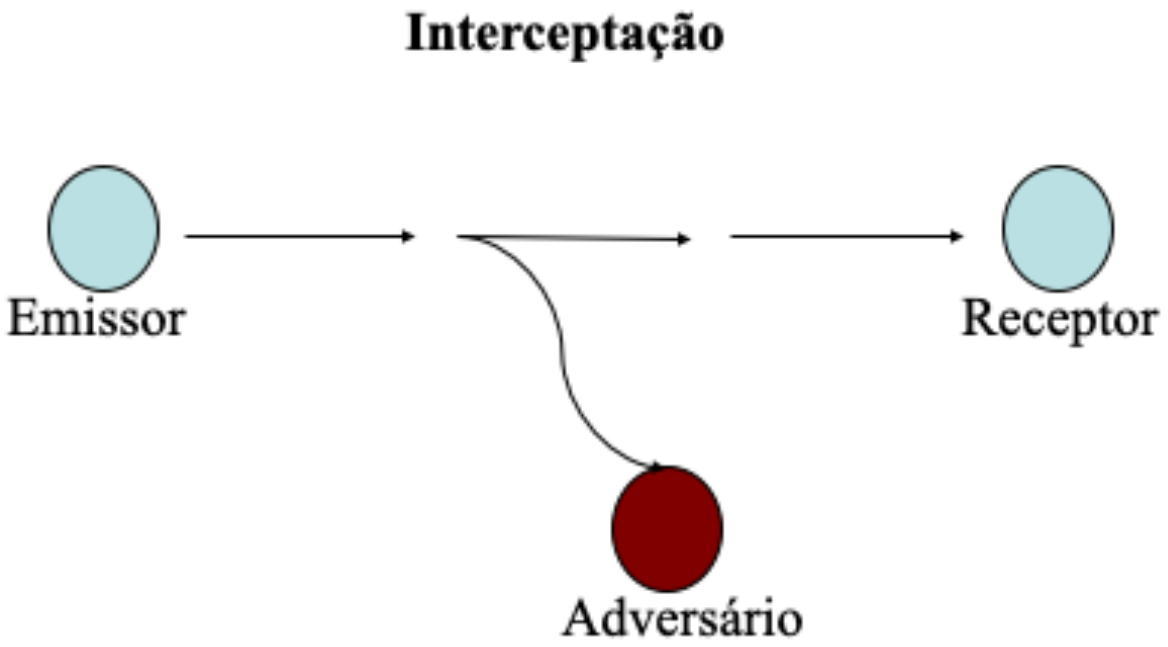
É um ataque à disponibilidade da informação

Ataques à segurança: disponibilidade



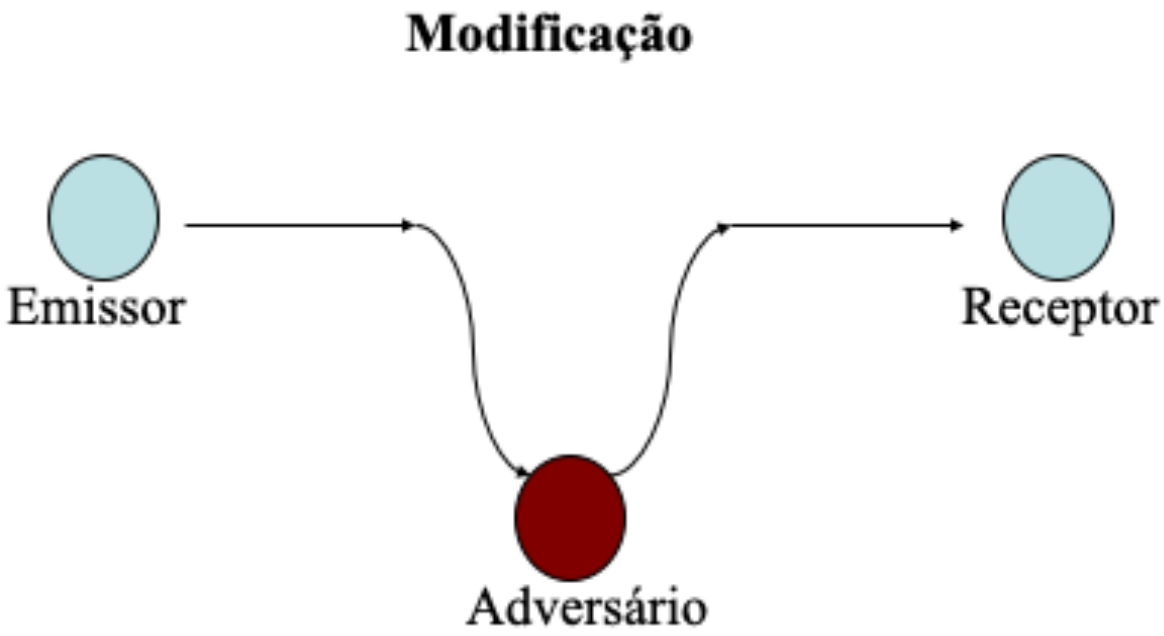
É um ataque à disponibilidade da informação

Ataques à segurança: confidencialidade



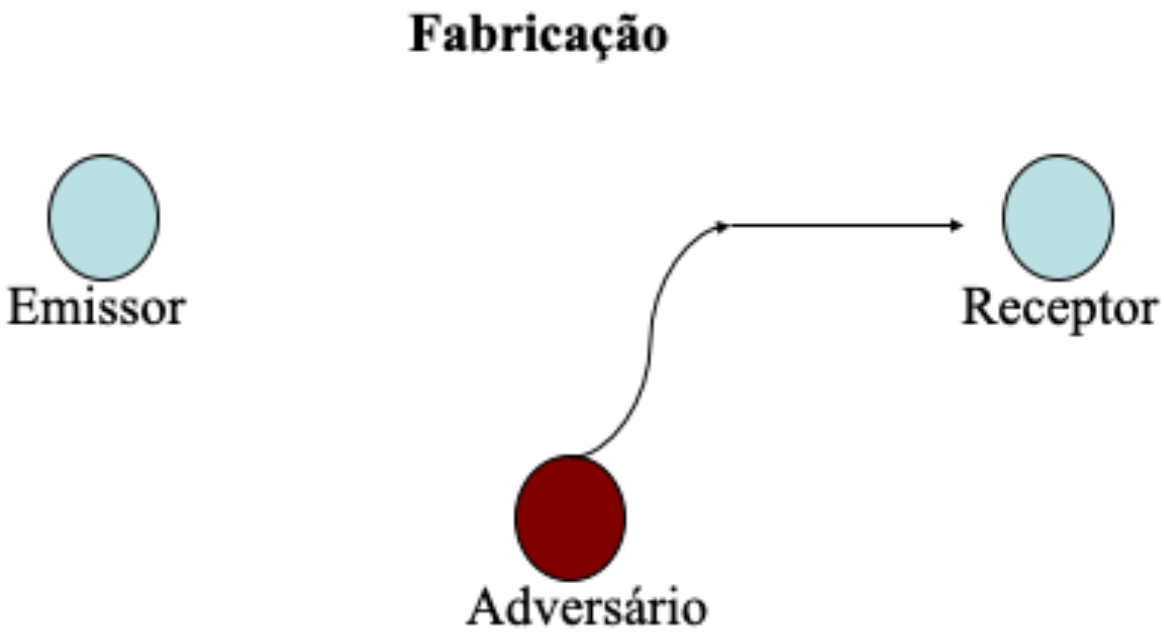
É um ataque à confidencialidade da informação

Ataques à segurança: integridade



É um ataque à integridade da informação

Ataques à segurança: autenticidade



É um ataque à autenticidade da informação

O que é criptografia?

- Uma ferramenta matemática para atingir objetivos de segurança da informação
  - De fato, é elemento fundamental, presente na maioria das soluções de segurança
  - Frequentemente não será a única ferramenta
  - Outras ferramentas:
    - Assinatura (clássica X digital)
    - Lacres
    - Lei (exemplo da violação de correspondência)
    - Políticas de segurança
    - Controle do acesso físico

Criptografia

- **escrita** (-grafia) **secreta** (cripto-)
- Terminologia
  - Texto plano ou mensagem plana – mensagem original
  - Texto cifrado ou mensagem cifrada – mensagem codificada
  - Transformação criptográfica – função que leva textos planos a cifrados (trans. de encriptação) ou textos cifrados a planos (transf. de decriptação)
  - Chave - informação que determina uma transformação criptográfica a ser utilizada

Criptografia

- Terminologia
  - Criptografar (encriptar) – converter texto plano em cifrado
  - Cifra – conjunto de transformações criptográficas indexadas por chaves
  - Descriptografar (decriptar) – converter texto cifrado em plano

Criptografia

- Terminologia
  - criptografia – estudo dos princípios e métodos de encriptação
  - criptanálise – estudo dos princípios e métodos para descriptografar sem o conhecimento da chave
  - criptologia – campo de estudo da criptografia e criptanálise
  - código – algoritmo que transforma uma mensagem compreensível em uma incompreensível a partir de um livro de códigos (ex.: códigos militares)

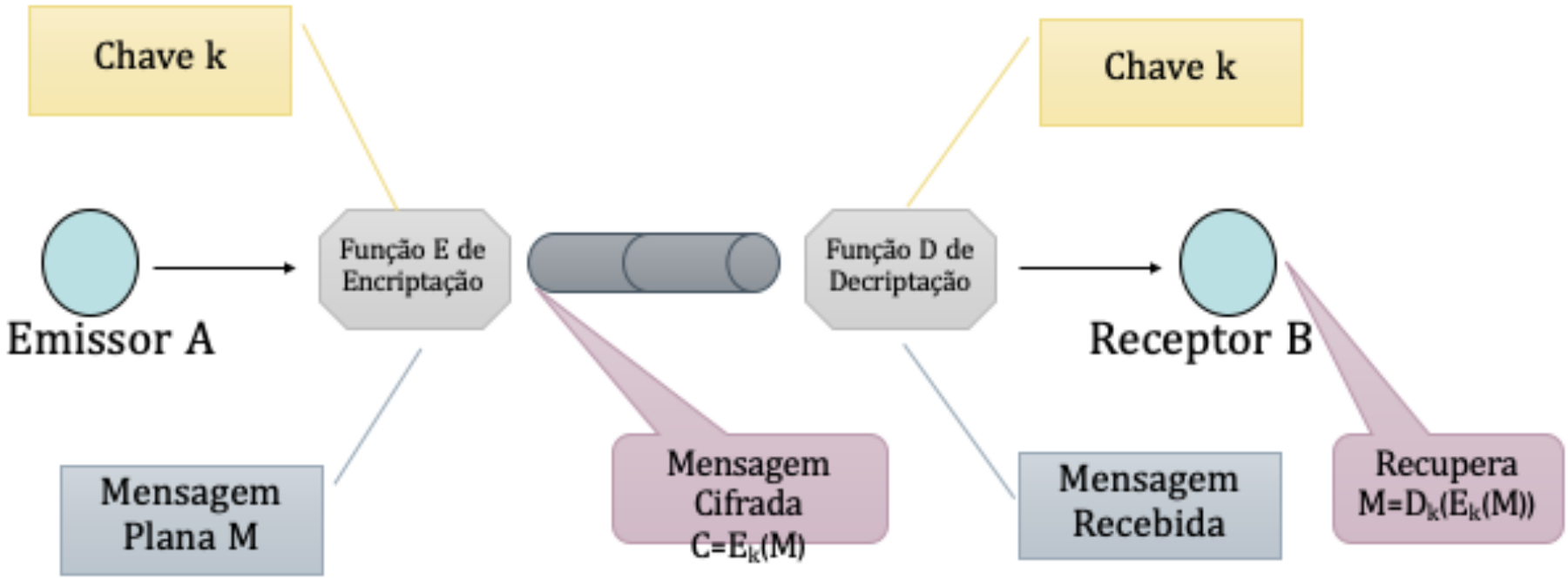
Ferramentas criptográficas Básicas

- Cifras (chave simétrica, bloco)
  - Objetivo: "confidencialidade" da informação
  - Chave "simétrica": mesma chave para encriptar e decriptar
- Hash (Resumo Criptográfico)
  - Objetivo: integridade da informaçao
  - Não usa chaves
- Códigos de Autenticação de Mensagem (MAC)
  - Objetivo: autenticação de origem da informação
  - Chave "simétrica": mesma chave para autenticar e verificar
- Acordo de Chaves Diffie-Hellman
  - Estabelecimento de chaves sem transmissão pelo canal
- Cifras (chave assimétrica)
  - Cifra com chaves distintas para encriptar e decriptar
  - Cada usuário possui uma chave privada e uma chave pública
- Assinatura Digital
  - Ferramenta para autenticidade e irrefutabilidade
  - Cada usuário possui uma chave privada e uma chave pública
- Cifras de Stream
  - Encripta no nível do bit
  - Modelo de chave simétrica
- Geradores de Números Aleatórios
  - Números aleatórios são usados extensivamente em criptografia
- Gerenciamento de chaves
  - Estabelecimento de Chaves
  - Certificação Digital (PKI)

Cifras (chave simétrica, em bloco)

- Aplicação clássica da criptografia
  - Objetivo: proteger mensagens em trânsito
- Tradicionalmente, assumia sigilo do algoritmo
  - Conceito moderno: apenas a chave fica secreta
- Cifra em bloco: opera sobre blocos de bits
  - Em oposição a cifras de fluxo, que operam bit-a-bit

Modelo para cifra simétrica



Cifras Clássicas

Cifra de César (Substituição)

- k=3
- Msg Plana:

VIM VI VENCI

- Mensagem cifrada

YLP YL YHQFL

Cifra de Cítala (Transposição)

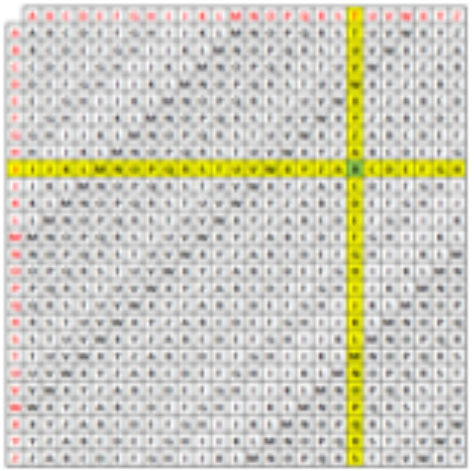
- k=2
- Msg Plana: ENCONTRE FRONT

E C N R F O T
N O T E R N

- Mensagem cifrada

E C N R F O T N O T E R N

Cifra de Gronsfeld-Vigenàre (Mistura com Chave)



- k=GREEN
- Msg Plana:

HELLO HOW ARE YOU

- Mensagem cifrada

NVPPB NFA EEK PSY

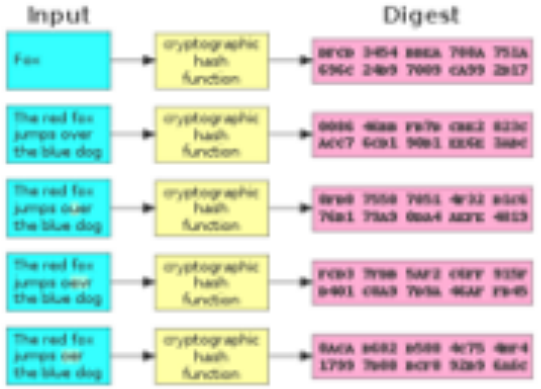
Kerchhoff desiderata (1883)

- (Requisitos para um esquema de encriptação)
- O esquema deve ser inquebrável, se não na teoria,então, na prática
- Divulgação dos detalhes do esquema (algoritmos) não deve causar problemas (redução de segurança)
- Chave deve ser memorizável sem auxílio de anotações e facilmente (freqüentemente) modificada
- Criptograma deve ser transmissível por telégrafo
- Aparato de encriptação deve ser transportável e operável por uma única pessoa
- O esquema deve ser facilmente utilizável, não necessitando-se de conhecimento prévio ou elevado poder mental

Modelos de Ataque

- Ataques criptanalíticos versus ataques a protocolos
- Exemplo de ataque a protocolo: transferência bancária
  - Mensagem de transferência contendo vários campos criptografados separadamente
- Tipos de ataques criptanalíticos
  - Ciphertext-only
  - Known-plaintext
  - Chosen (plain/cipher)text

Hash (Resumo Criptográfico)



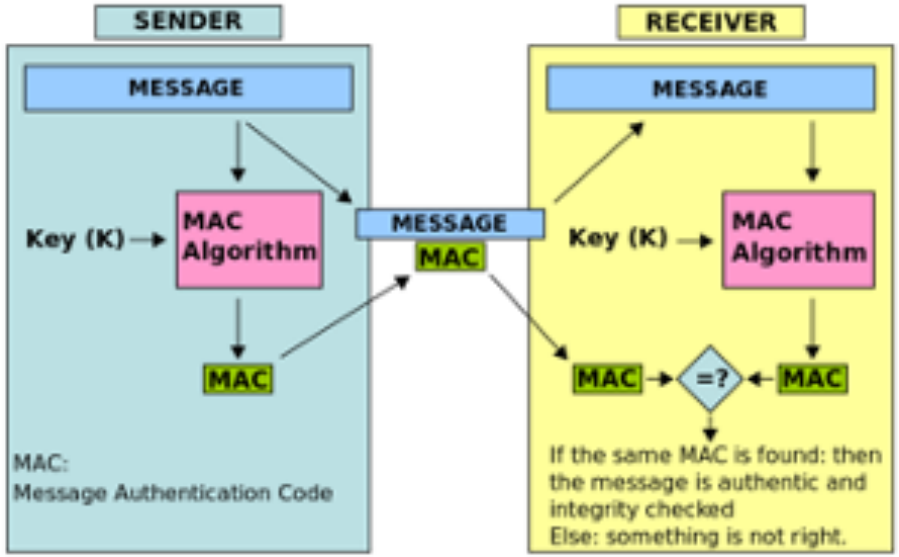
- Função criptográfica sem chave
  - Entrada: mensagem de tamanho "arbitrário"
  - Saída: mensagem de tamanho fixo (centenas de bits)
- Funciona como identificador "seguro" da mensagem
- Baseia-se nas propriedades de resistência a colisão:
  - Resistência a ataque de pré-imagem (one-way)
  - Resistência fraca a colisão (segunda pré-imagem)
  - Resistência forte a colisão
- Hashes ingênuos (não-cripto): truncar, cifrar XOR de blocos
- Hashes modernos construídos com operações de substituição, permutação etc.
- Aplicações: versão de software, imagem de disco em forense, controle de acesso em Unix

Códigos de Autenticação de Mensagem (MAC)

› Espécie de resumo criptográfico dependente de chave
  – Resumo associado à mensagem, chave autentica origem

› Enviado junto à mensagem, identificando origem

› Resistência a colisões e inversões (assim como hash)



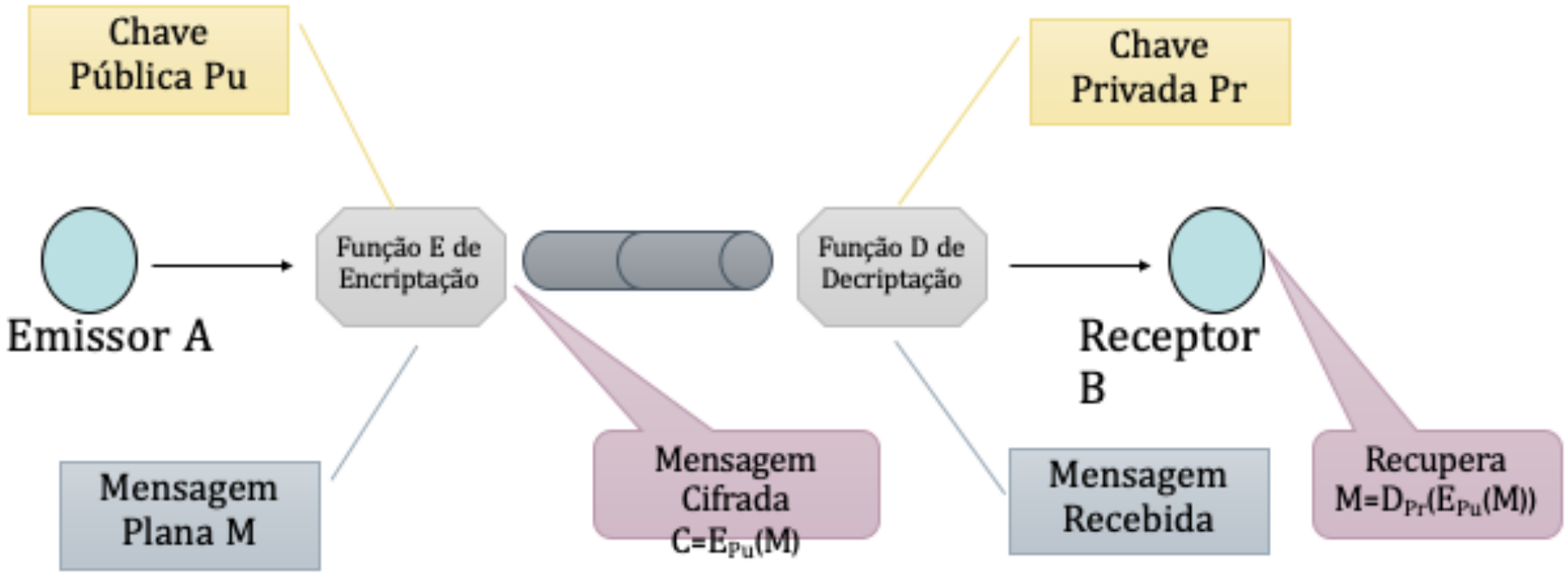
Acordo de Chaves Diffie-Hellman

- Permite que duas partes concordem remotamente sobre uma chave criptográfica
- Chave criptográfica é construída a partir de informações trafegadas na rede
  - Alice sorteia k_A e envia $f(k_A)$
  - Bob sorteia k_B e envia $f(k_B)$
  - Alice constrói $g(k_A,f(k_B))$ igual a $g(k_B,f(k_A))$ construído por Bob
- Alice e Bob passam a usar $k=g(k_A,f(k_B))=g(k_B,f(k_A))$
- Não é possivel recuperar k a partir das informações trafegadas $f(k_A)$ e $f(k_B)$

Cifras de chave pública (chave assimétrica)

- Ferramenta de confidencialidade
- Assimetria entre encriptação e decriptação
  - Uma chave (pública) encripta e outra (privada) decripta
- Maior quebra de paradigma da história da criptografia
  - Primeiro algoritmo prático desenvolvido por Rivest, Shamir e Adleman em 1976
  - Algoritmo era conhecido pela inteligência britânica desde 1973 (Cocks)
- Baseia-se na ”dificuldade computacional”
  - Assimetria na complexidade de computar f e f^{-1}
  - Exemplos: multiplicação de primos, exponenciação discreta

Modelo para cifra assimétrica



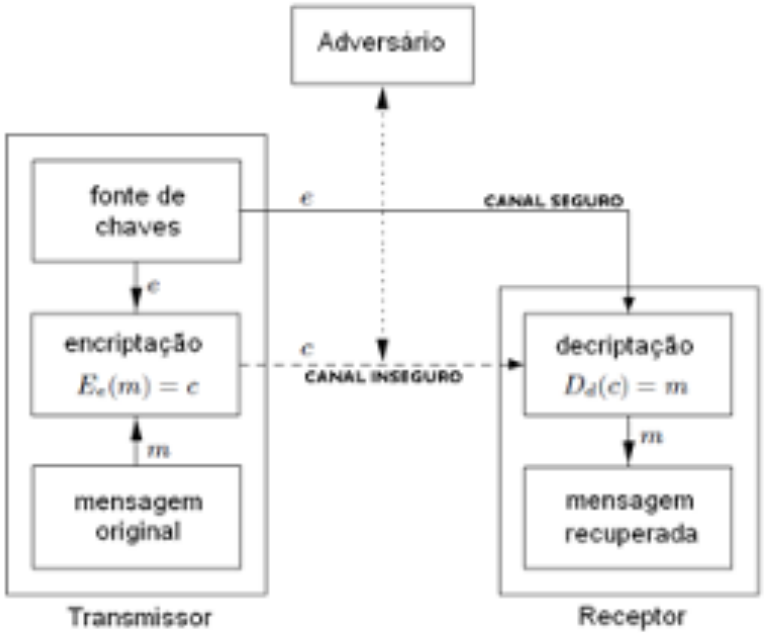
Analogias: chave simétrica vs chave pública

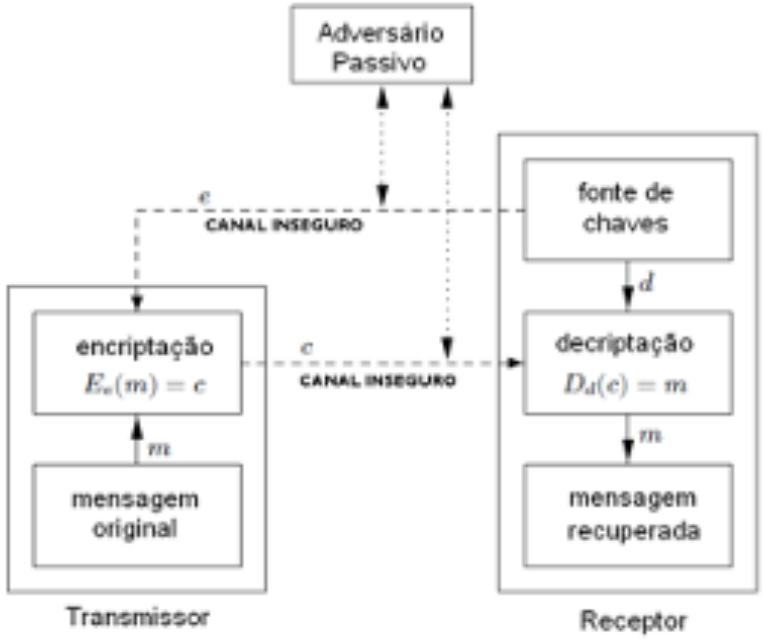
- Chave simétrica: cadeados e chaves
  - Conjunto de pessoas possui chave de cadeado de caixa
  - Apenas essas pessoas podem inserir algo na caixa e trancar
    - Equivalente a encriptar mensagem
  - Exatamente essas pessoas podem retirar da caixa trancada
    - Equivalente a decriptar mensagem
- Chave pública: cadeados e chaves
  - Cada pessoa possui um cadeado e sua respectiva chave
  - Cada pessoa pode disponibilizar cadeados abertos
    - Ou seja, divulgar sua chave pública
  - Qualquer um pode trancar uma caixa com o cadeado
    - Ou seja, encriptar mensagens
  - Apenas o dono da chave pode abrir o cadeado (e a caixa)
    - Ou seja, decriptar mensagens

Analogias: chave simétrica vs chave pública

- Chave simétrica: cofre
  - Conjunto de pessoas possui segredo do cofre
  - Apenas essas pessoas podem inserir algo
    - Equivalente a encriptar mensagem
  - Exatamente essas pessoas podem retirar algo
    - Equivalente a decriptar mensagem
- Chave pública: caixa de correio
  - Cada pessoa possui uma caixa
  - Qualquer pessoa pode inserir mensagem
    - Ou seja, encriptar a mensagem
  - Apenas o dono da caixa pode retirar as mensagens
    - Ou seja, decriptar as mensagens

Comparando os modelos

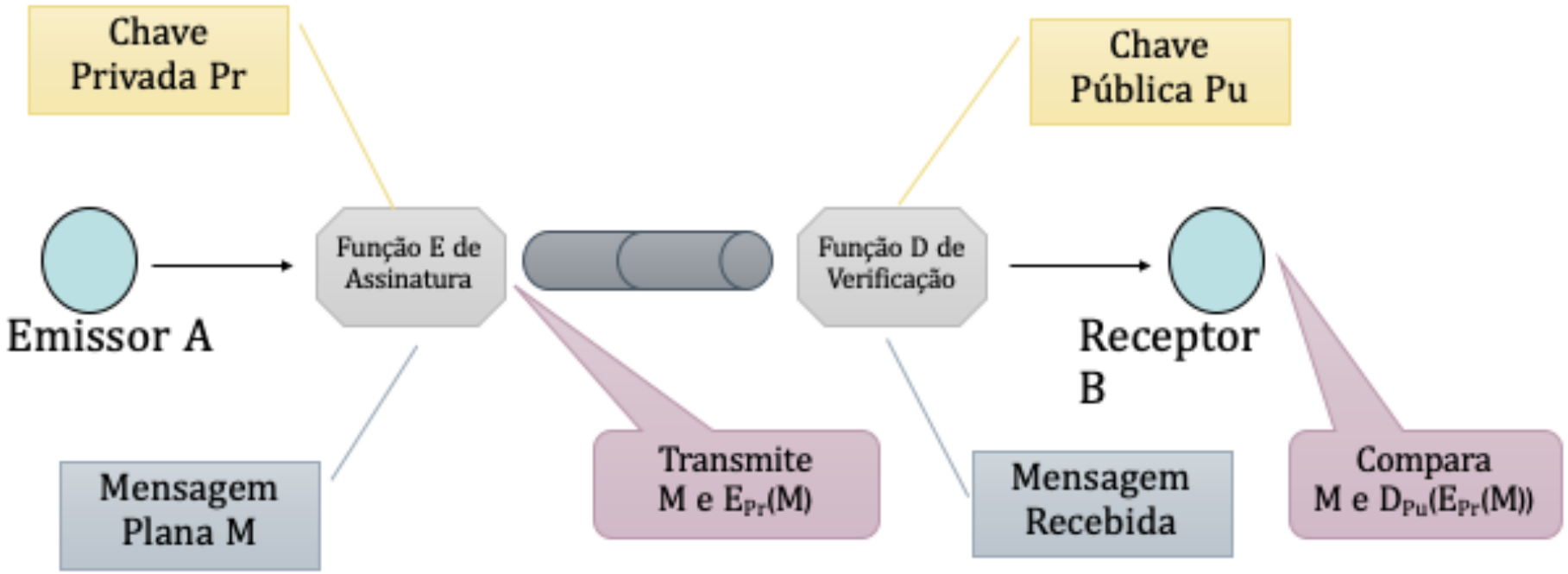




Assinatura Digital

- Um tipo de criptografia assimétrica
  - Inversa da cifra de chave pública
- Espécie de resumo criptográfico da mensagem construído a partir da chave privada
  - Assim, provê não-repúdio
- Mensagem enviada acompanhada da assinatura digital
  - Não provê confidencialidade

Modelo para assinatura digital



Cifras de Stream

- Cifras de stream criptografam bit a bit – em oposição à cifra de bloco
- Exemplo da Cifra de Vernam – one-time pad
  - Única cifra com segurança incondicional
  - Desvantagem do tamanho da chave
- Cifras de stream práticas: keystream gerada a partir uma semente compartilhada entre as partes

Geradores de Números Aleatórios

- Conceito de "aleatoriedade"
- True Random Number Generator (TRNG)
  - Números aleatórios a partir de eventos físicos
- Pseudo Random Number Generator (PRNG)
  - Números "aparentemente" aleatórios gerados por algoritmos determinísticos – possibilita reprodutibilidade
- Estrutura típica
  - TRNG como fonte de entropia que é inserida em um PRNG
- Aplicações
  - Criação de chaves criptográficas
  - Cifras de stream (fluxo de bits a serem XORed com mensagem)
  - Criação de nounces/IV em protocolos e modos de operação
  - Desafios em protocolos challenge-response
  - Aplicações em protocolos diversos (de sorteio a assin. contratos)

Gerenciamento de Chaves

- Estabelecimento de chaves: acordo versus transporte
- Problema prático: como as partes podem concordar a respeito de chaves criptográficas?
  - Difícil resolver se não temos um canal autenticado
- Modelo "mais seguro": as partes se encontram pessoalmente e definem suas chaves
  - Não é prático se temos redes globais envolvendo bilhões de pessoas e dispositivos
- Certificado digital: associação entre identidade e chave pública certificada por TTP
- PKI: estrutura hierárquica de TTPs para certificação digital
  - Alguns TTPs emitem certificados de chave pública
  - Outros TTPs autorizam a "operação" de TTPs na rede
  - Um TTP é a "raiz" da estrutura e origem da confiança

Criptografia

Visão do Livro =)

Funções unidirecionais e hash criptográfico

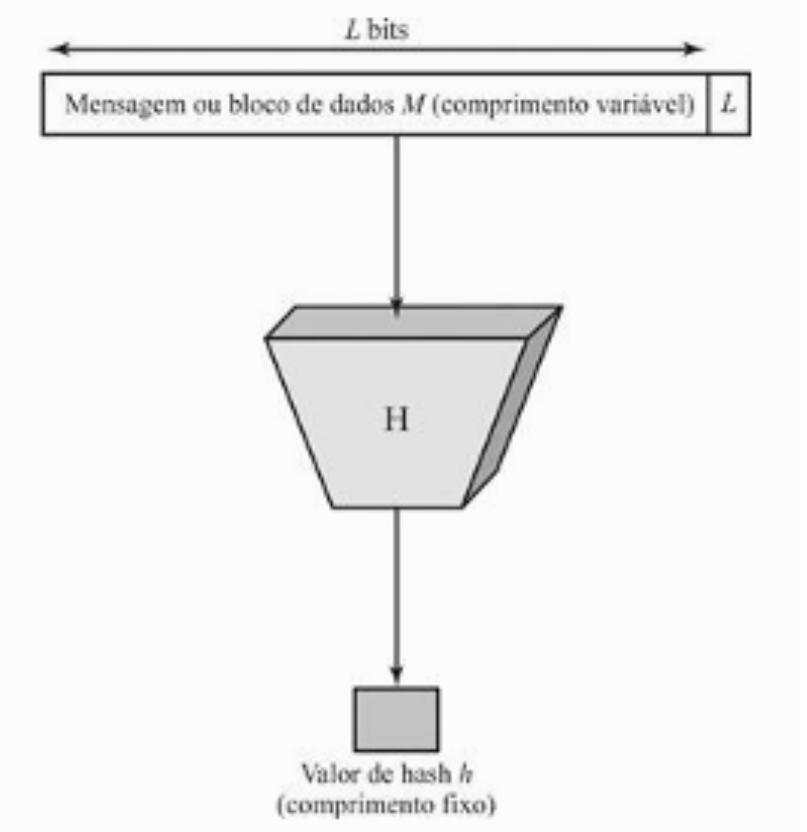
- Função unidirecional: fácil computar, difícil inverter
  - "A" ferramenta fundamental da criptografia
  - Exemplo: $f(x)=3^x \bmod 17$
- Hash criptográfico: recebe como entrada uma mensagem e gera um pequeno "resumo"
  - É uma função unidirecional: inviável encontrar mensagem a partir de um resumo
  - Resistência a colisões: inviável encontrar duas mensagens com o mesmo resumo
  - Não confundir com tabela de dispersão e hash não-criptográfico

Exemplo de função unidirecional

- Seja $X=\{1,...,16\}$ 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

Lógica do hash criptográfico

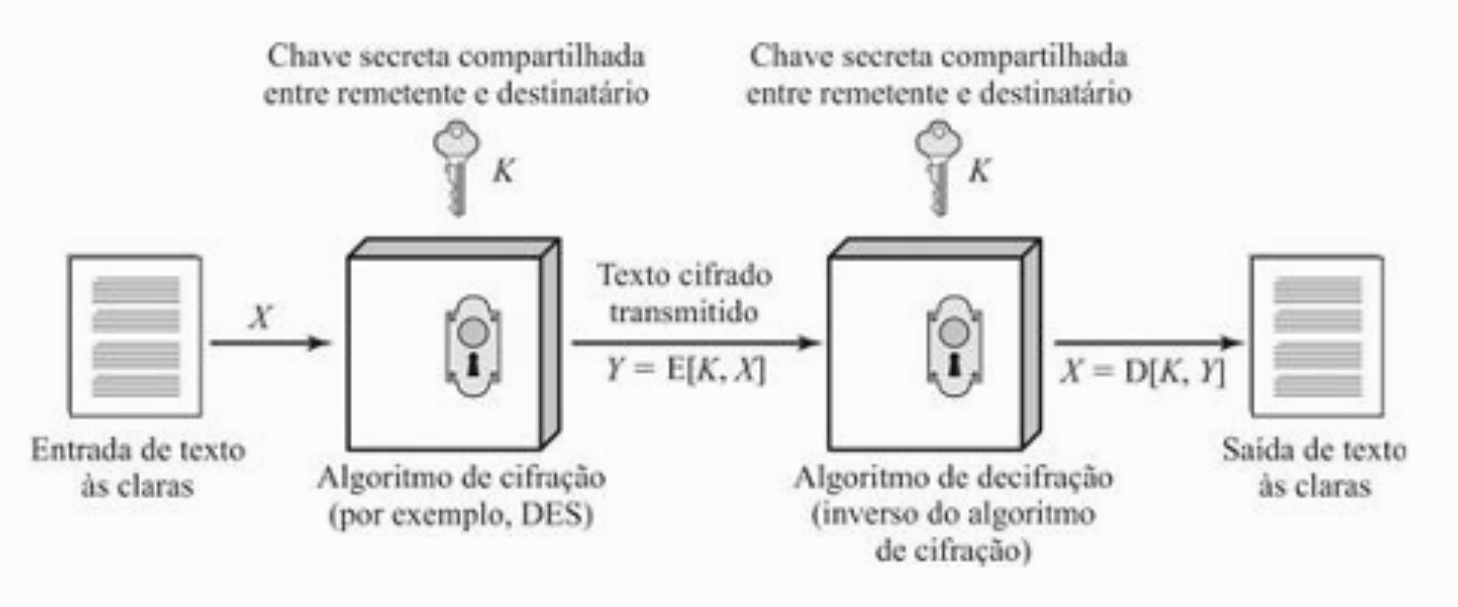


Segurança da função hash

- Resistência a identificação de pré-imagem
  - Para qualquer código dado h, é inviável em termos computacionais achar x tal que H(x) = h.
- Resistência fraca a colisão
  - Para qualquer mensagem dada x, é inviável em termos computacionais achar $y \neq x$ tal que H(y) = H(x).
- Resistência forte a colisão
  - É inviável, em termos computacionais, achar qualquer par (x, y) tal que H(x) = H(y).

Cifração Simétrica

- Texto em claro / às claras
- Algoritmo de cifração
- Chave secreta
- Texto cifrado
- Algoritmo de decifração



Algoritmo de cifração de bloco

- Cifra mensagem processando-a em ”blocos” de tamanho fixo.
- Algoritmos mais importantes
  - Data Encryption Standard (FIPS PUB 46)
  - Triple DES (ANSI X9.17 e FIPS PUB 46-3)
  - Advanced Encryption Standard (FIPS PUB 197)
- Modos de operação – como trabalhar com blocos em mensagens longas
  - Exemplo mais simples: ECB

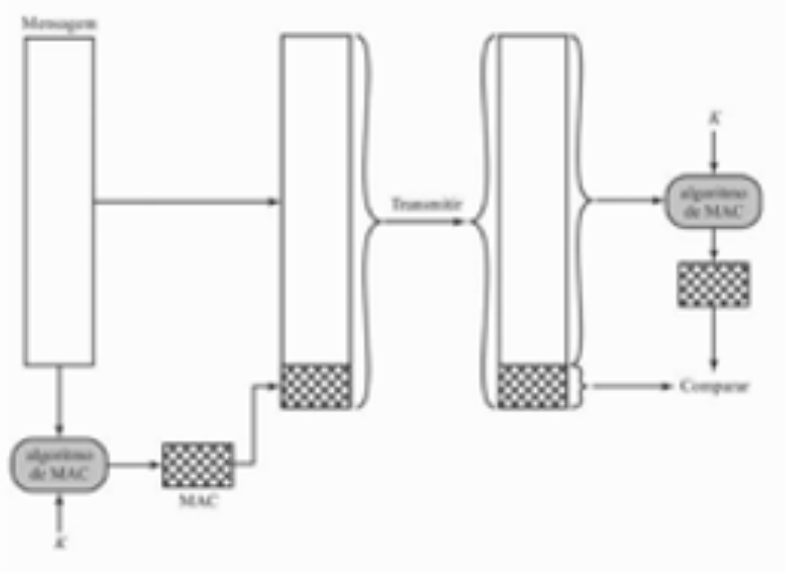
Cifra de fluxo

- Processa mensagem bit-a-bit (ou byte-a-byte)
- Estreitamente relacionada a geradores de números aleatórios
  - Exemplo de abordagem: XOR entre os bits da mensagem e os bits de um PRNG

Autenticação de mensagem e funções hash

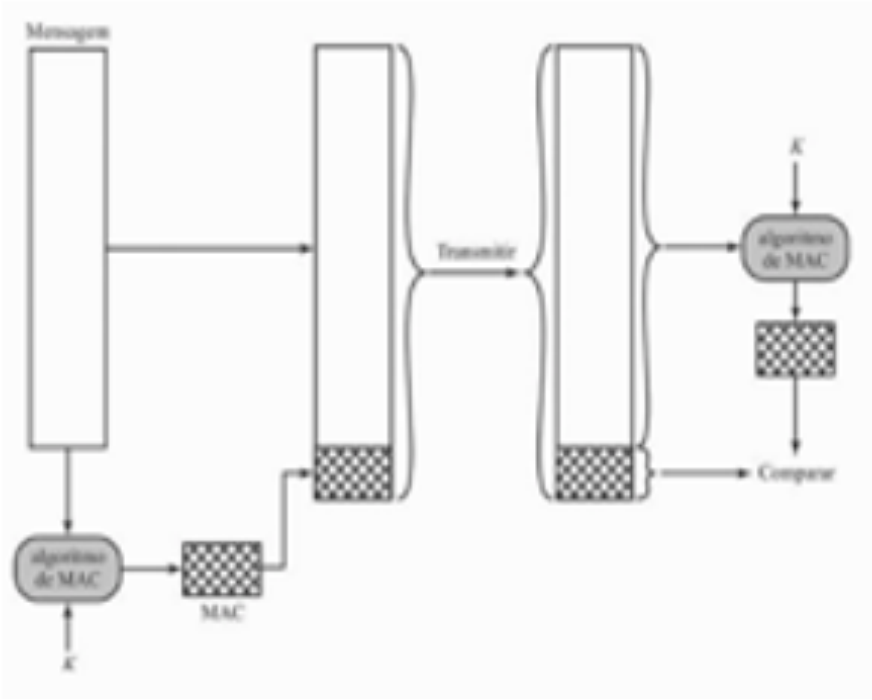
› Autenticação com cifra simétrica
  – Mensagens válidas devem compor um código

› Autenticação sem cifra simétrica
  – Mensagem segue junto com código de autenticação

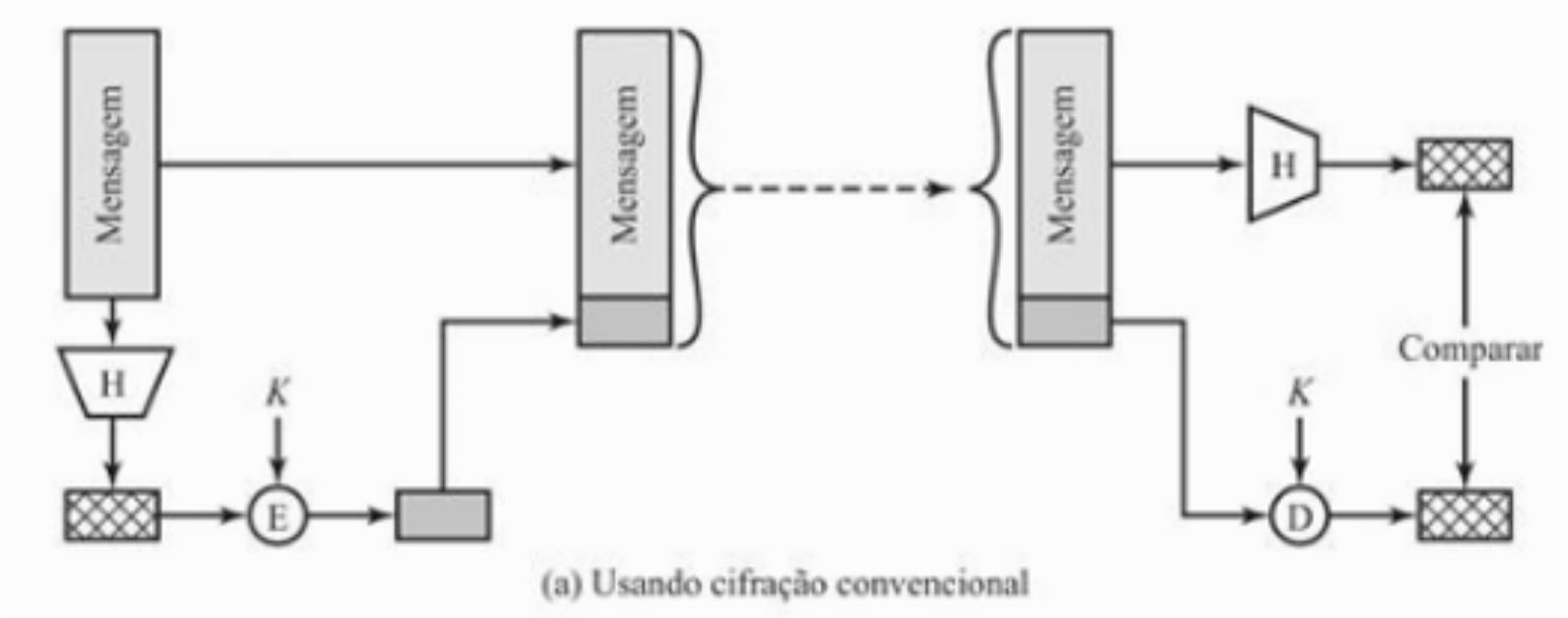


Message authentication code (MAC)

- Recebe como entrada uma mensagem M e uma chave secreta K_{AB} e gera como saída um pequeno bloco de dados $MAC_M=F(K_{AB},M)$, o cód. aut. mensagem
- Trasmissor envia M e MAC_M
- Receptor recalcula $F(K_{AB},M)$ e compara com MAC_M recebido
- Funciona porque só se pode gerar MAC_M conhecendo-se a chave secreta K_{AB}

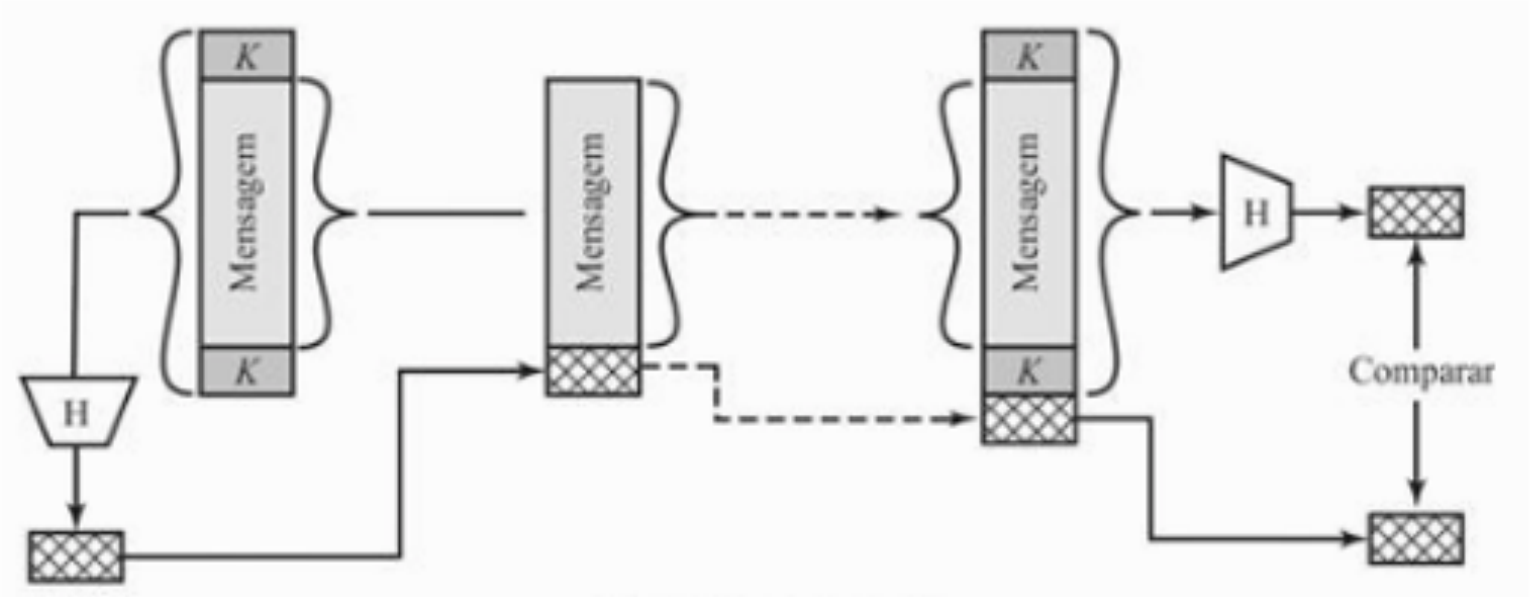


Autenticação com cifra simétrica

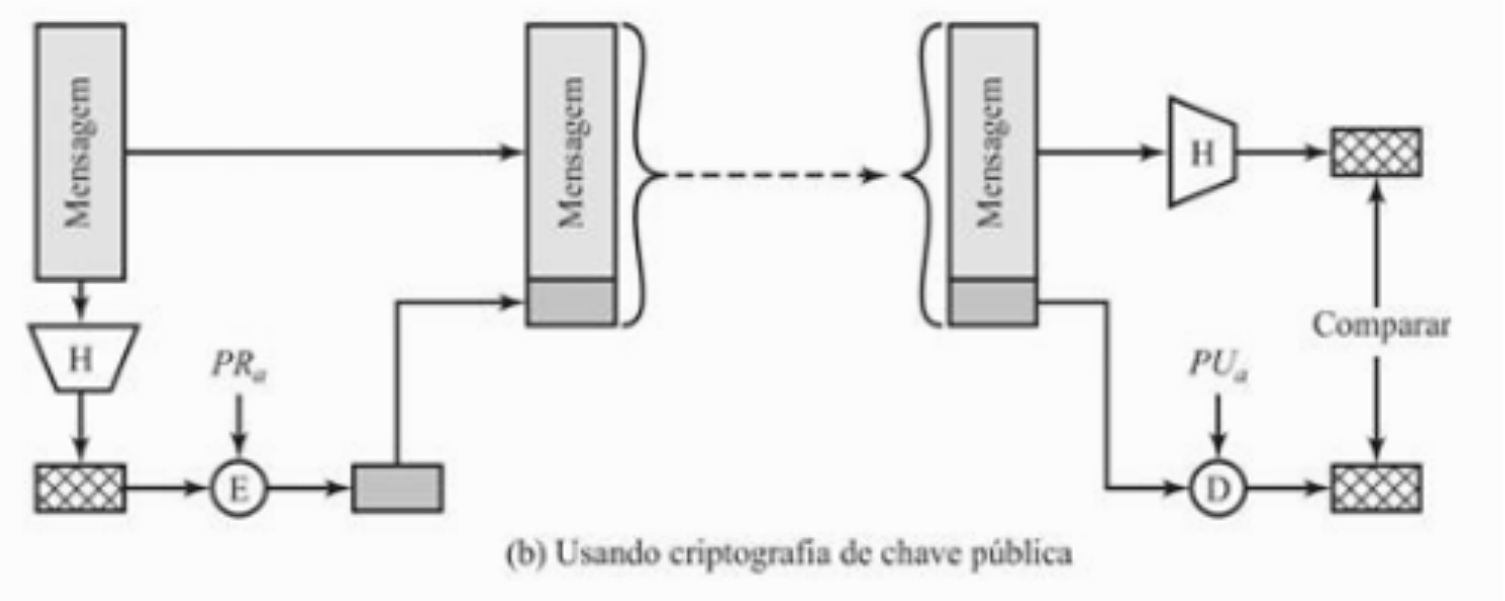


(a) Usando cifração convencional

Autenticação com hash/MAC



Autenticação com chave pública



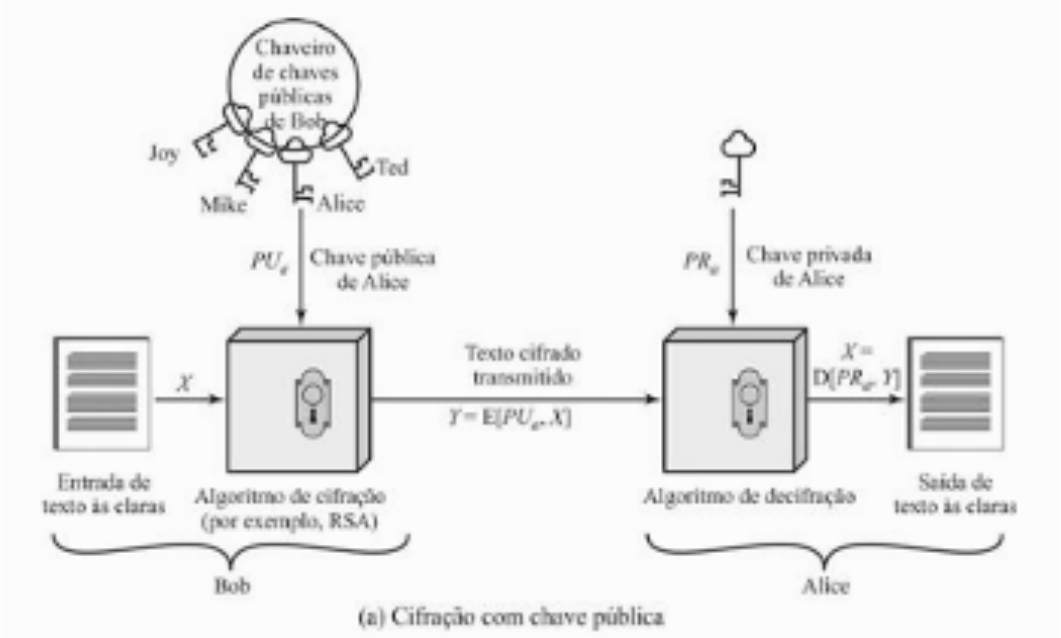
(b) Usando criptografia de chave pública

Criptografia de chave pública

› Acordo de chaves: Diffie e Hellman, 1976

› Cifra de chave pública: Rivest Shamir e Adleman, 1977

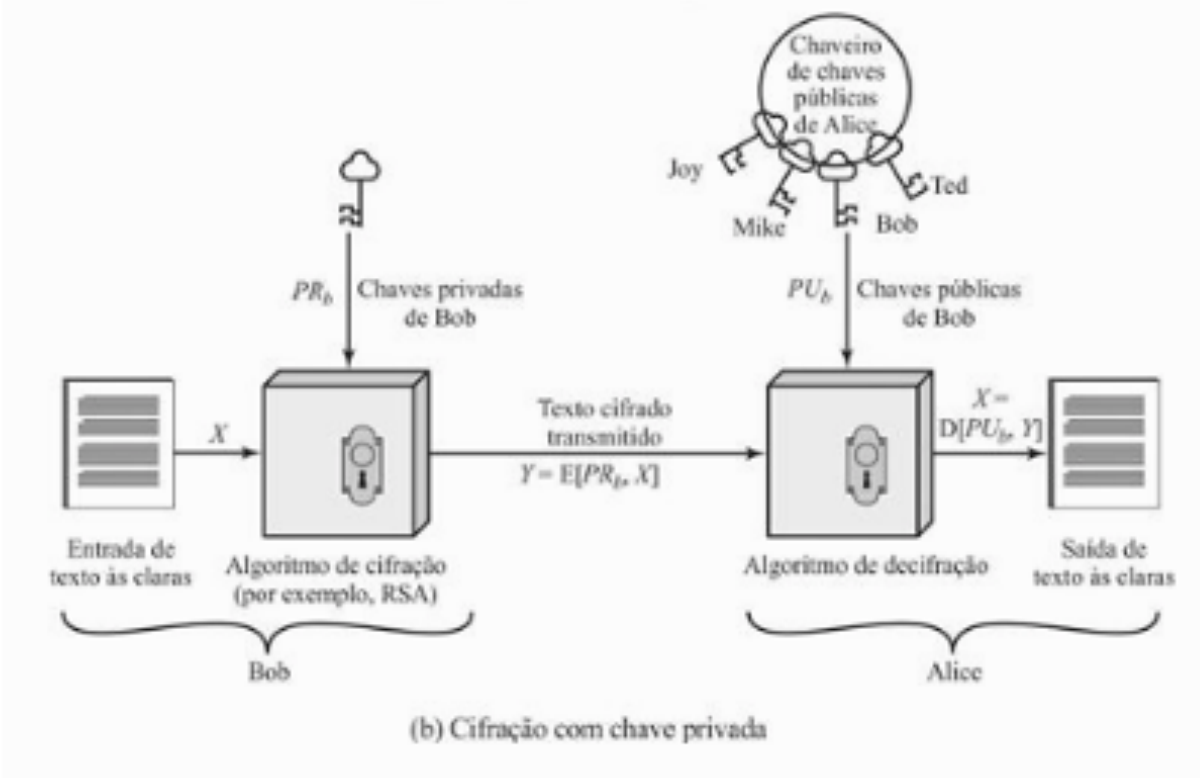
› Supostamente conhecido pelo GCHQ em 1973



(a) Cifração com chave pública

Assinaturas digitais

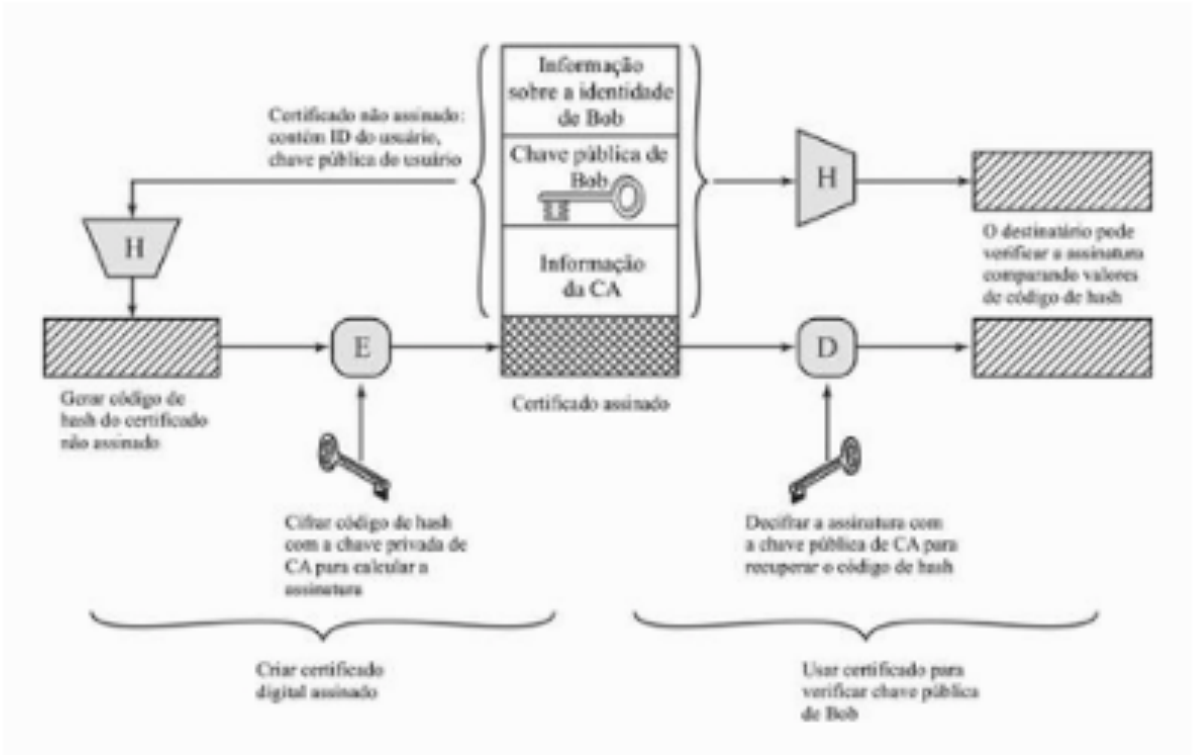
› Uso da chave privada para atestar a origem



(b) Cifração com chave privada

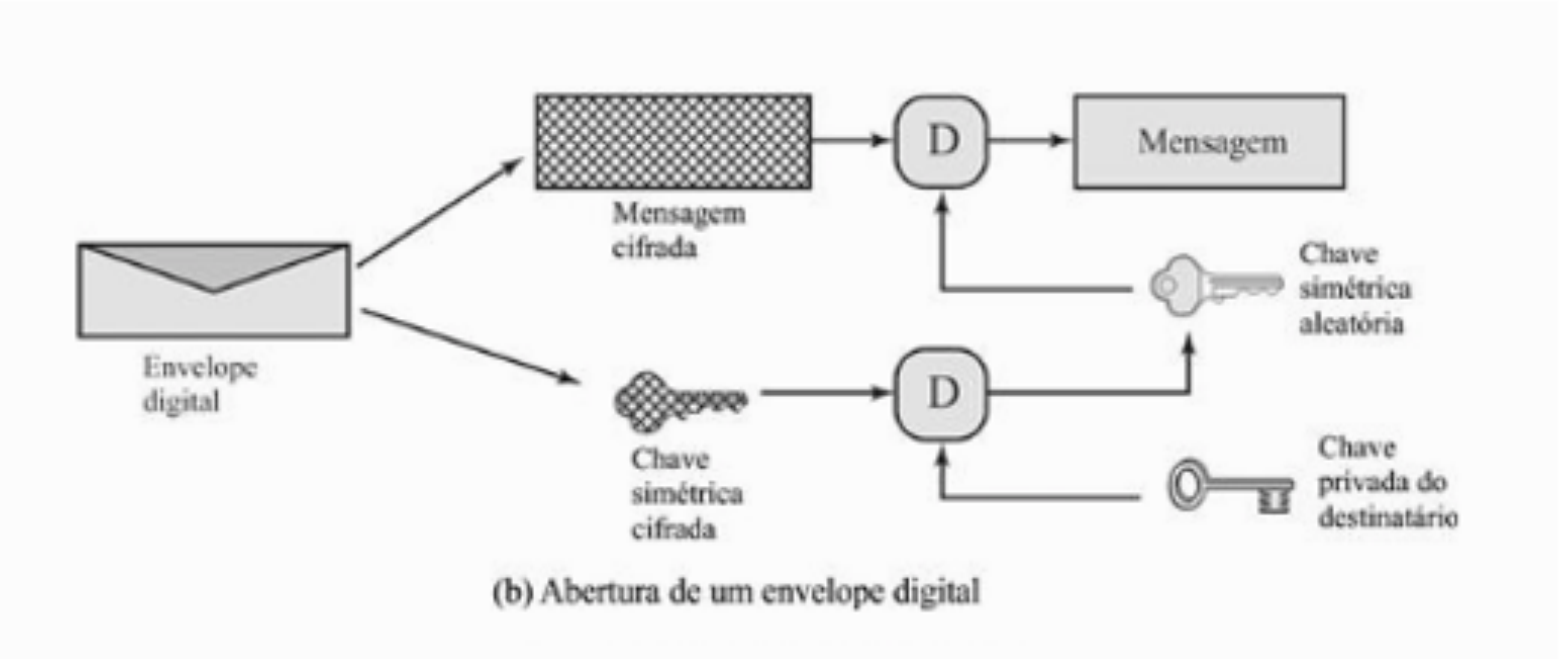
Certificados de chave pública

› Associação entre ”identidade” e ”chave pública” realizada por terceiras partes confiáveis



Envelope digital

› Mensagem é cifrada simetricamente e a chave é cifrada com a chave pública do destinatário



(b) Abertura de um envelope digital

Cifras modernas de bloco com chave simétrica

Cifras de bloco

- Cifras de bloco quebram a mensagem e a encriptam bloco a bloco
- É como uma substituição de caracteres “longos” (64-bits ou mais)
- A maioria das cifras mais populares na atualidade são deste tipo

Princípios das cifras de bloco

- Cifras de bloco funcionam como uma grande substituição
  - Para blocos de 64 bits, cada uma das 2^{64} mensagens planas é levada, de forma bijetiva, em uma de 2^{64} mensagens cifradas
- Construir tal tabelas de substituições seria impraticável
- Muitos dos cifradores de bloco modernos são baseados na chamada Estrutura de Cifra de Feistel
- Utilizam-se blocos menores de construção
- Então, usa-se a idéia de composição de cifras

Cifras de bloco iteradas

- Envolve a repetição de funções internas chamadas rounds
- São parâmetros da cifra
  - Número de rounds
  - O tamanho do bloco
  - O tamanho da chave, de onde serão tiradas as subchaves de cada round
- Cada round deve ser uma função bijetiva

Estrutura das cifras de Feistel

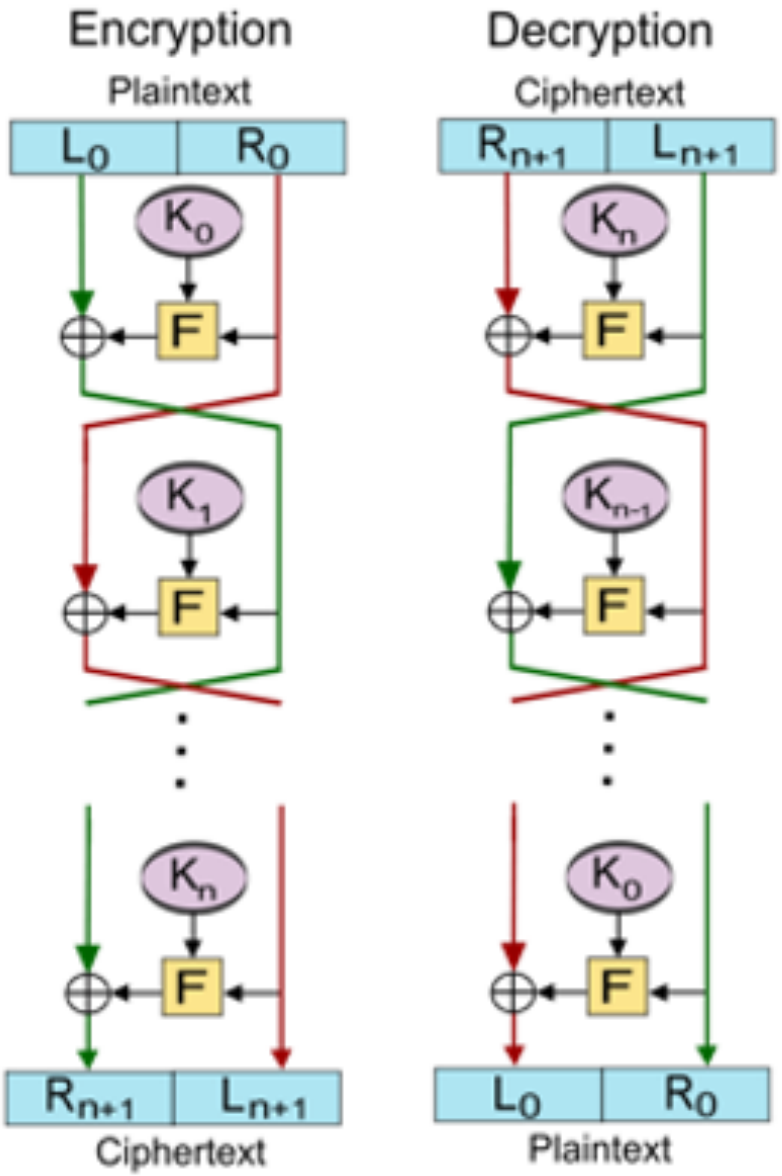
- Desenvolvida por Horst Feistel
- Particiona o bloco de entrada em duas partes de mesmo tamanho
- Processa em rounds nos quais
  - Aplica substituição, na metade esquerda, baseada no conteúdo da metade direita e em subchave derivada da chave
  - Então, permuta as duas partes
- Formalmente:
  - $L_i = R_{i-1}$, $R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$

Decriptação da Cifra de Feistel

- Tipicamente, $r \geq 3$ rounds
- Ordena a saída como (R_r, L_r)
- Decriptação: apenas aplicar os rounds na ordem reversa

Estrutura da cifra de Feistel

- Cifras de Feistel ou modificações da cifra de Feistel: Blowfish, Camellia, CAST-128, DES, FEAL, ICE, KASUMI, LOKI 97, Lucifer, MARS, MAGENTA, MISTY1, RC5, TEA, Triple DES,Twofish, XTEA, GOST 28147-89
- Generalizações da cifra de Feistel: CAST-256, MacGuffin, RC2, RC6, Skipjack, SMS4, CLEFIA



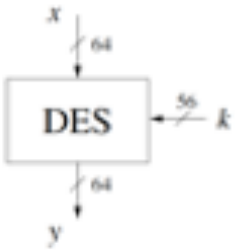
Data Encryption Standard (DES)

- Cifra de bloco por muito tempo mais usada no mundo
  - Após ter recomendação retirada, passou a ser usada na forma do triple-DES (que está para ser aposentado tb=)
- Pode-se dizer que é a mais estudada e conhecida
- Padronizada em 1977 pelo NBS (agora NIST)
  - FIPS PUB 46
- Encripta blocos de 64 bits usando chaves de 56 bits
- Enorme importância histórica
- Já não é considerada segura
  - Substituída pelo DES
  - Ainda em uso na forma de Triple-DES

História do DES

- IBM desenvolve a cifra Lucifer em 1971
  - Equipe liderada por Feistel
  - Bloco de 48, 32 or 128 bits
  - Chave de 48, 64 or 128 bits
- Em 1973, o NBS solicitou propostas para um novo padrão nacional de cifras
- A IBM submeteu uma versão revisada do Lucifer, que finalmente seria aceita como DES

Estrutura do DES

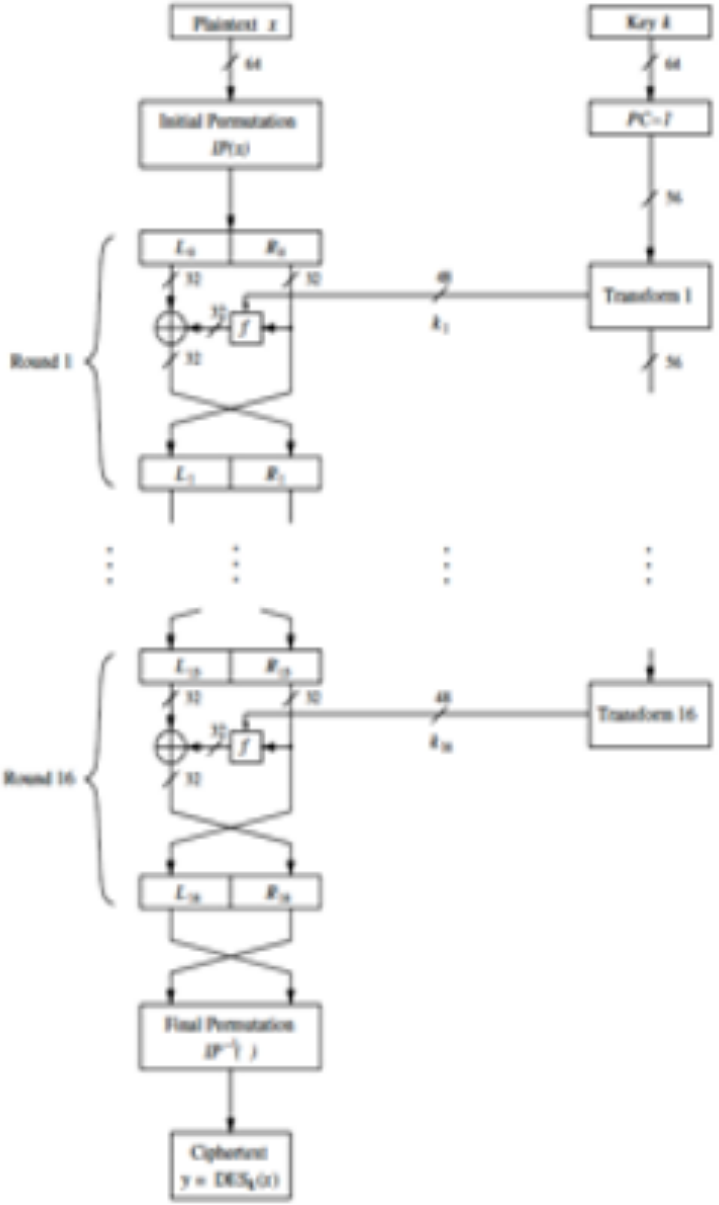


- Tamanho de bloco: 64 bits
- Tamanho de chave: 56 bits
  - (64 bits com 8 de paridade)
- Número de estágios: 16 rounds
  - 16 subchaves de 48 bits
  - Cada round é Feistel:

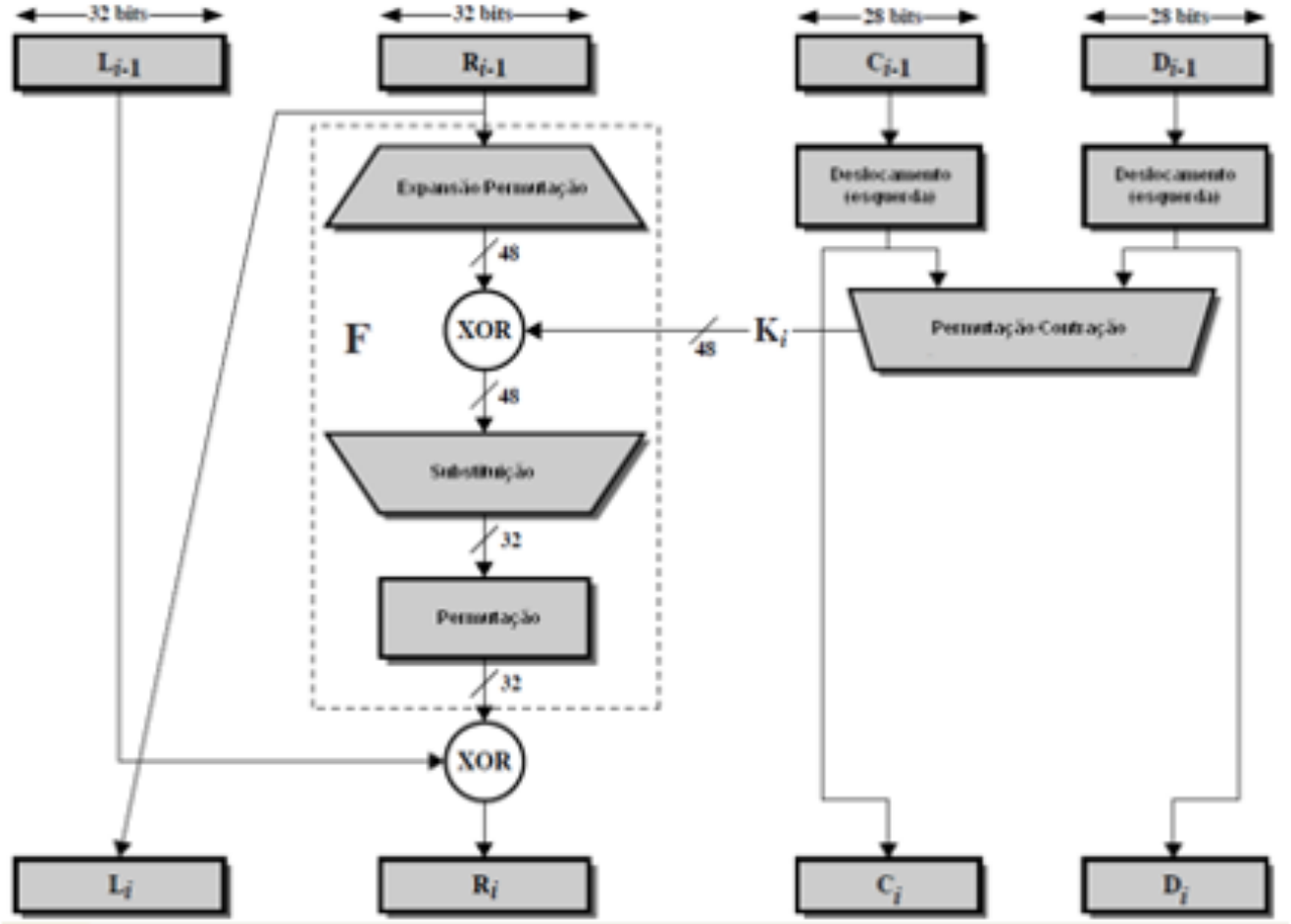
    $L_i = R_{i-1}$;

    $R_i = L_{i-1}$ XOR $f(R_{i-1},K_i)$, onde

    $f(R_{i-1},K_i) = P(S(E(R_{i-1}) \text{ XOR } K_i))$



Round do DES


32 bits
32 bits
28 bits
28 bits
L_{i-1}
R_{i-1}
C_{i-1}
D_{i-1}
Expansão-Permutação
Deslocamento esquerda
Deslocamento esquerda
48
F
XOR
48
K_i
Permutação-Contração
48
Substituição
32
Permutação
32
XOR
L_i
R_i
C_i
D_i

Triple DES

- Necessidade de substituição do DES
  - Diversos ataques demonstrados
  - Ataques de busca exaustiva de chave
- Possibilidade: usar múltiplas repetições do DES
- Double DES permite ataque “meet-in-the-middle”
- Três encriptações oferecem bem mais segurança

Triple DES

- Variação do DES – tripla encriptação com duas ou três chaves
- Padrão estabelecido em ANSI X9.17 & ISO 8732
- Ataques práticos ainda desconhecidos
  - Força-bruta bastante inviável
  - Ataque meet-in-the-middle com três chaves precisa de 2^{112} operações e 2^{56} memória
- Alternativa ainda popular

Dupla e tripla encriptações

- Dupla encriptação: $E(x) = E_{K_2}(E_{K_1}(x))$
- Tripla encriptação: $E(x) = E'_{K_3}(E'_{K_2}(E'_{K_1}(x)))$
  - E'_K pode denotar E_K ou $D_K = E_K^{-1}$
  - O caso $E(x) = E_{K_3}(D_{K_2}(E_{K_1}(x)))$ é denominado E-D-E tripla encriptação
    - O subcaso $K_1 = K_3$ é denominado tripla encriptação de duas chaves

Triple DES (cont.)

- Duas chaves
  - Seqüência E-D-E
  - $E(x) = E_{K_1}(D_{K_2}(E_{K_1}(x)))$
  - Padronizado em ANSI X9.17 e ISO8732
  - Sem ataques práticos conhecidos
- Três chaves
  - $E(x) = E_{K_3}(D_{K_2}(E_{K_1}(x)))$
  - Oferece maior segurança
  - Adotado por algumas aplicações Internet, como PGP e S/MIME

AES – Advanced Encryption Standard

- Uma substituição do DES mostrava-se necessária
  - Diversos ataques teóricos demonstrados
  - Diversos ataques de busca exaustiva de chave
- Triple-DES podia ser usado – mas era lento
- NIST efetuou uma "chamada de cifras" em 1997
- 15 candidatos aceitos em Junho de 1998
- 5 selecionados para fase seguinte em Agosto de 1999
- Rijndael selecionado como AES em Outubro de 2000
- Publicado como padrão FIPS PUB 197 em Novembro de 2001

Requisitos do AES

- Cifra de chave simétrica
- Blocos de 128 bits, chaves de 128/192/256 bits
- Mais rápido e forte que Triple-DES
- Vida útil de 20 a 30 anos
- Especificações completas e detalhes de projeto
- Implementações em Java e C

Critérios de avaliação AES

- Critério inicial:
  - segurança – esforço para criptanalisar
  - custo – computacional
  - Algoritmo e características de implementação
- Critério final:
  - Segurança geral
  - Facilidade de implementação (software e hardware)
  - flexibilidade

O selecionados do AES

- Lista de Agosto de 1999:
  - MARS (IBM) - complexo, rápido, alta margem de segurança
  - RC6 (EUA) - muito simples, muito rápido, pequena margem de segurança
  - Rijndael (Bélgica) - limpo, rápido, boa margem de segurança
  - Serpent (Europa) – limpo, lento, altíssima margem de segurança
  - Twofish (EUA) - complexo, muito rápido, alta margem de segurança

O selecionados do AES

- Diferenças-chave entre os selecionados
  - Estratégia de rounds
    - Poucos rounds complexos versus muitos rounds simples
  - Inovação
    - Redefinições de cifras existentes versus novas propostas

O AES – Rijndael

- Projetado por Rijmen-Daemen na Belgica
- Cifra iterativa, em vez de Feistel
  - Manipula dado em 4 grupos de 4 bytes
  - Opera o bloco inteiro em cada round
- Objetivos de projeto
  - Resistencia contra ataques conhecidos
  - Velocidade e código compacto em diversas CPUs
  - Projeto simples

Rijndael

- Processa blocos em quatro grupos de 4 bytes
- possui 9/11/13 rounds nos quais executa:
  - Substituição de bytes (um S-box para todos os bytes)
  - Deslocamento de linhas
  - Mistura de colunas
  - Adição (XOR) da subchave do round
- Possui um XOR inicial e o último round é incompleto
- Todas as operações podem ser combinadas em operações XOR e buscas em tabela
  - Bastante rápido e eficiente

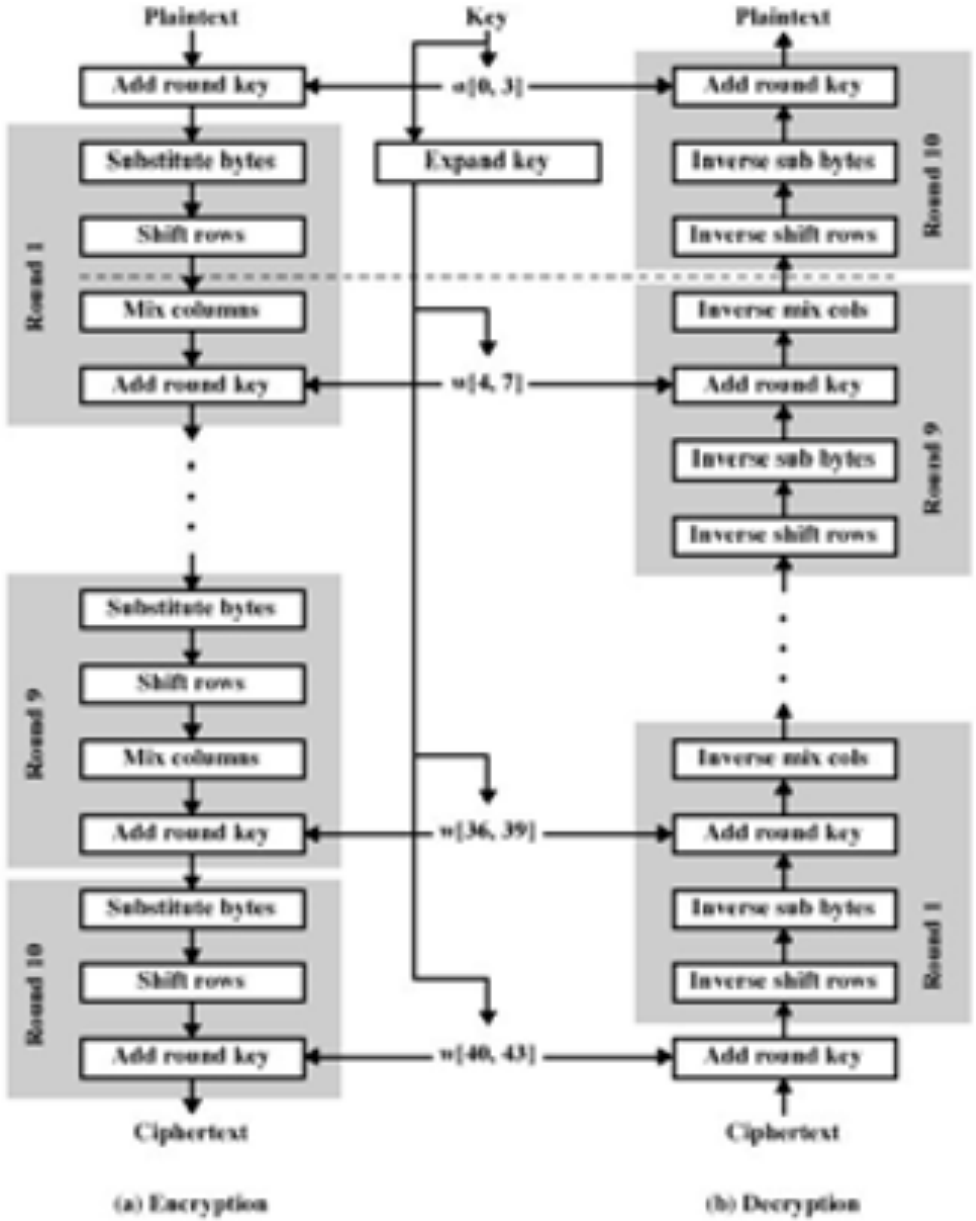
Aspectos de implementação

- Implementação eficiente em CPU 8 bits
  - Substituição de bytes usando tabela com 256 entradas
  - Deslocamento de linha é deslocamento de byte
  - Adição de chave é byte XOR
  - Mistura de colunas pode ser simplificada com busca em coluna

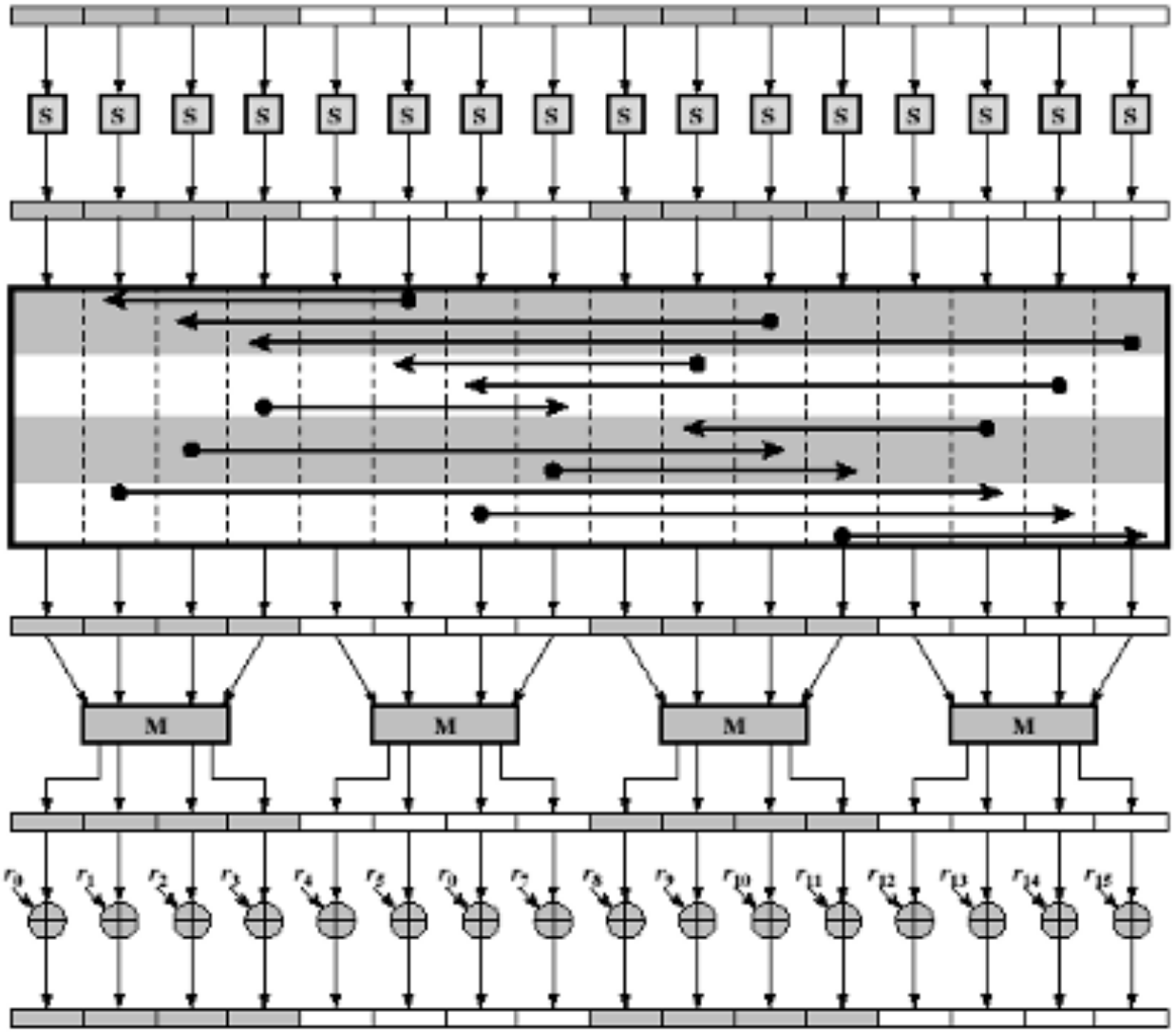
Aspectos de Implementação

- Implementação eficiente em CPU 32 bits
  - Redefina passos para usar palavras de 32 bits
  - precompute 4 tabelas de 256 palavras
  - Cada coluna em cada round pode ser precomputada usando 4 buscas em tabela + 4 operações XOR
  - Custo de 16Kb para armazenar tabelas
- Projetistas crêem que esta implementação eficiente foi decisiva na escolha do Rijndael

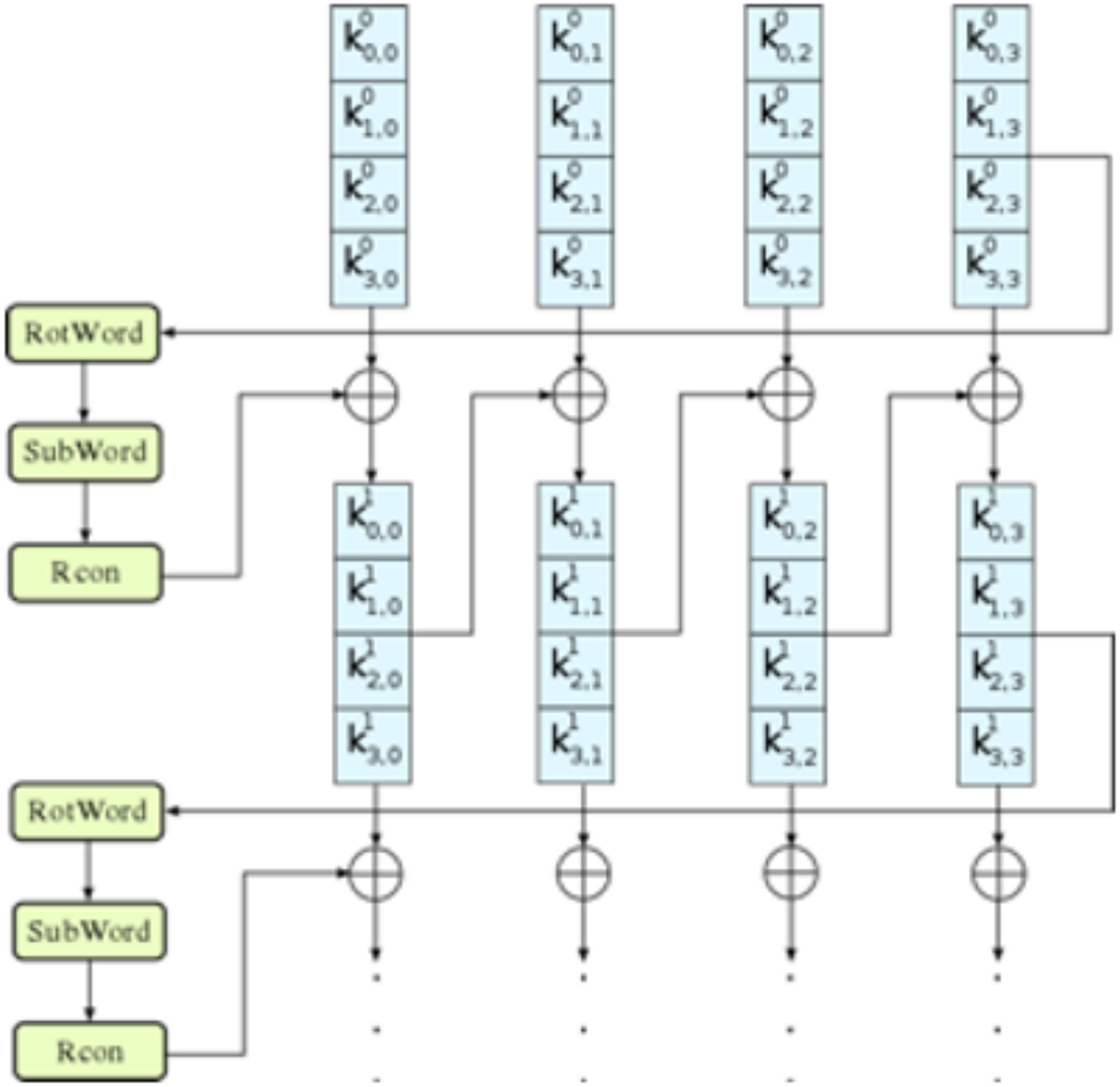
Estrurura do AES



Round AES



Key Schedule



Modos de operação de cifras

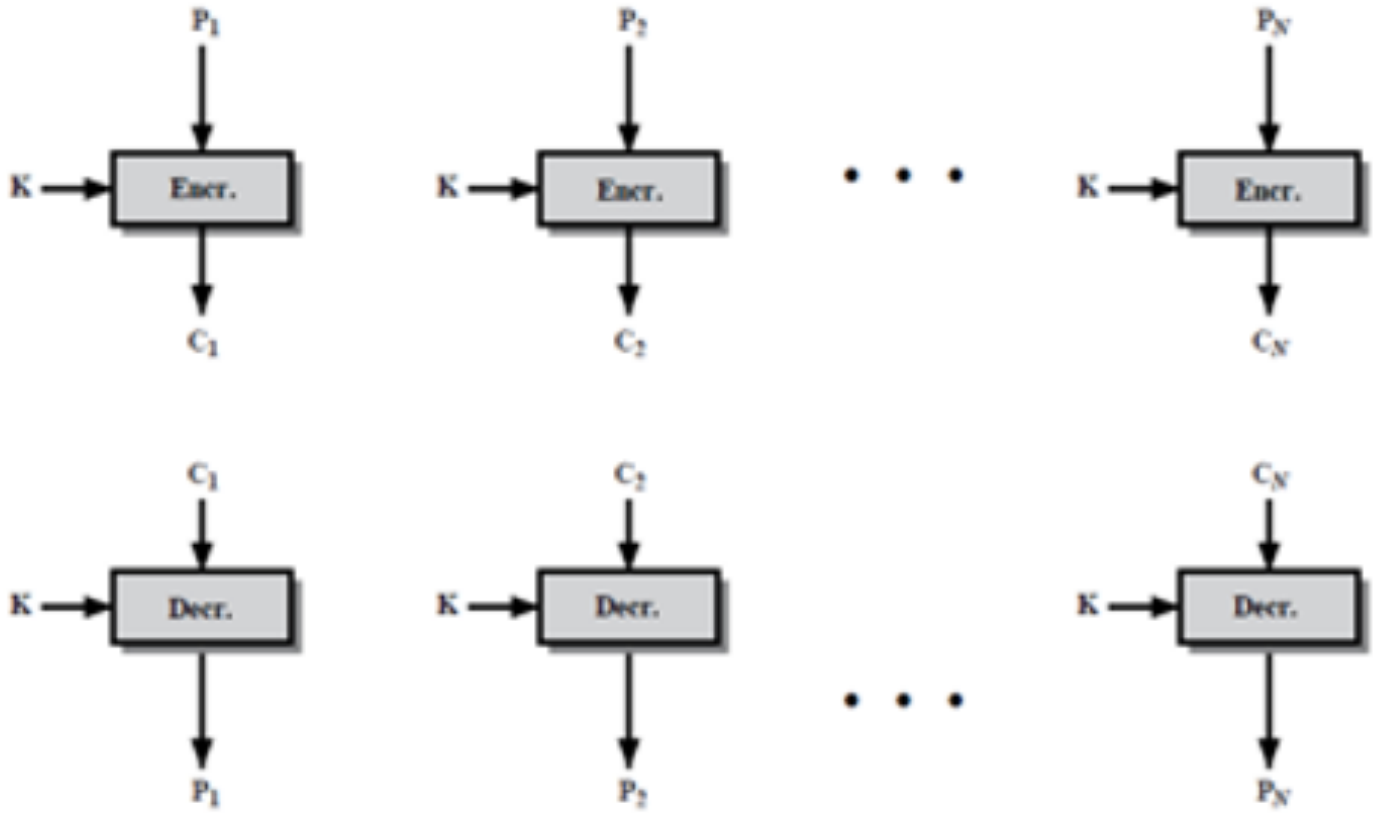
Modos de Operação

- 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

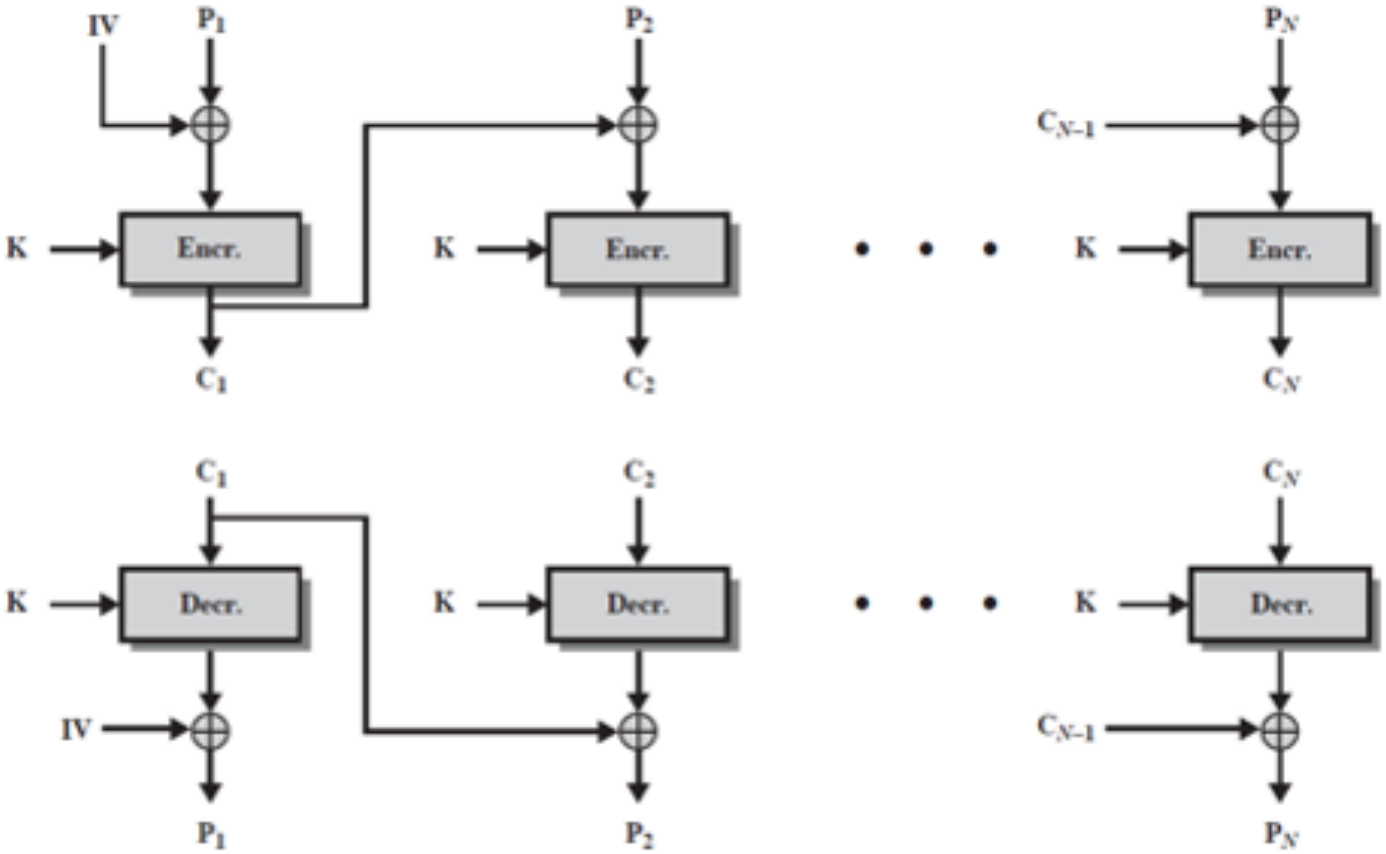
Modo ECB



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

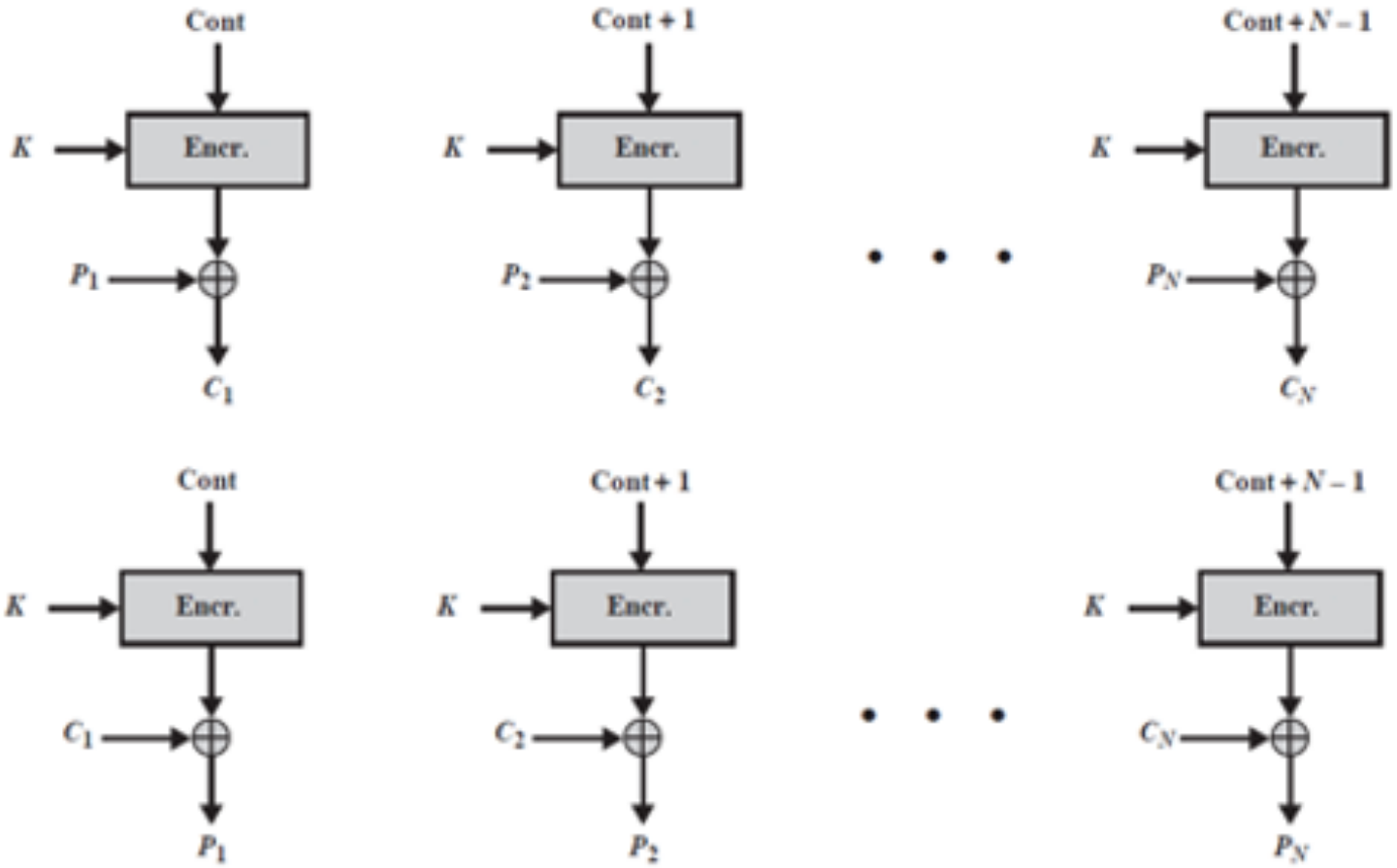
Modos stream: CTR, CFB e OFB

- Cifra é usada para gerar keystream que é combinada com a mensagem

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

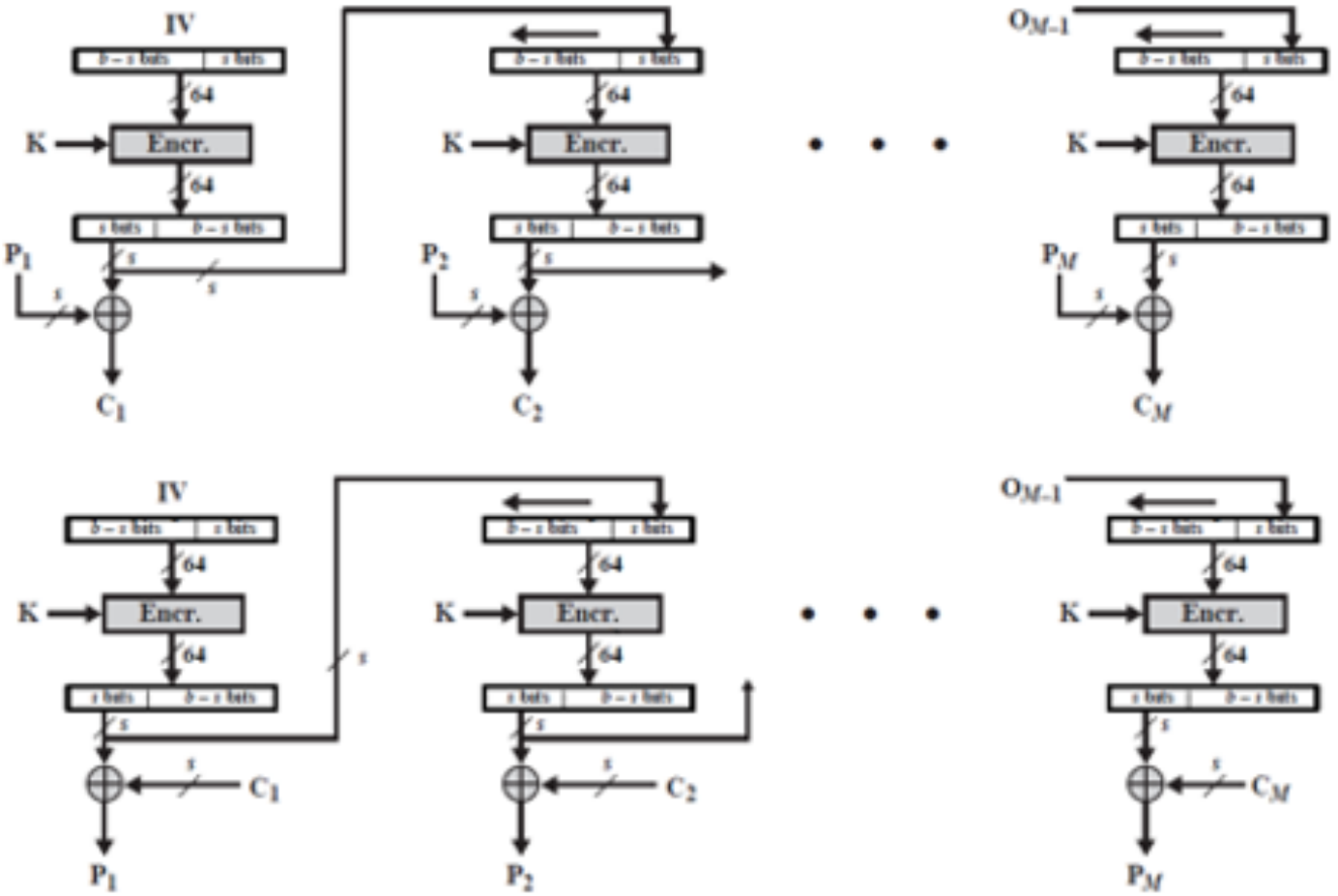
Modo CTR


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

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

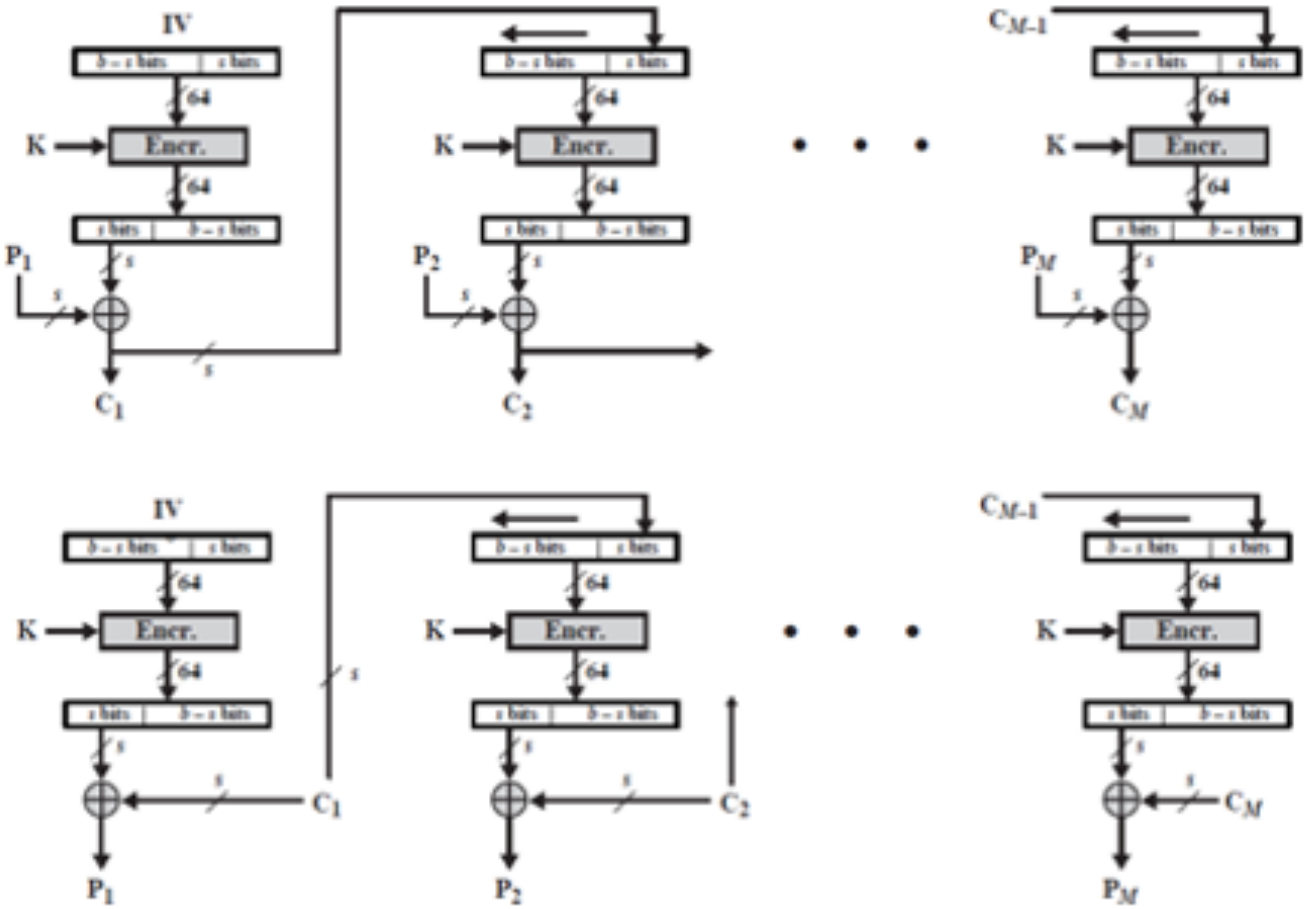
Modo OFB



Modo 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


IV
b – s bits
s bits
64
K
Encr.
64
s bits
b – s bits
P1
s
C1
P2
C2
CM-1
PM
CM
C1
C2
CM

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

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...

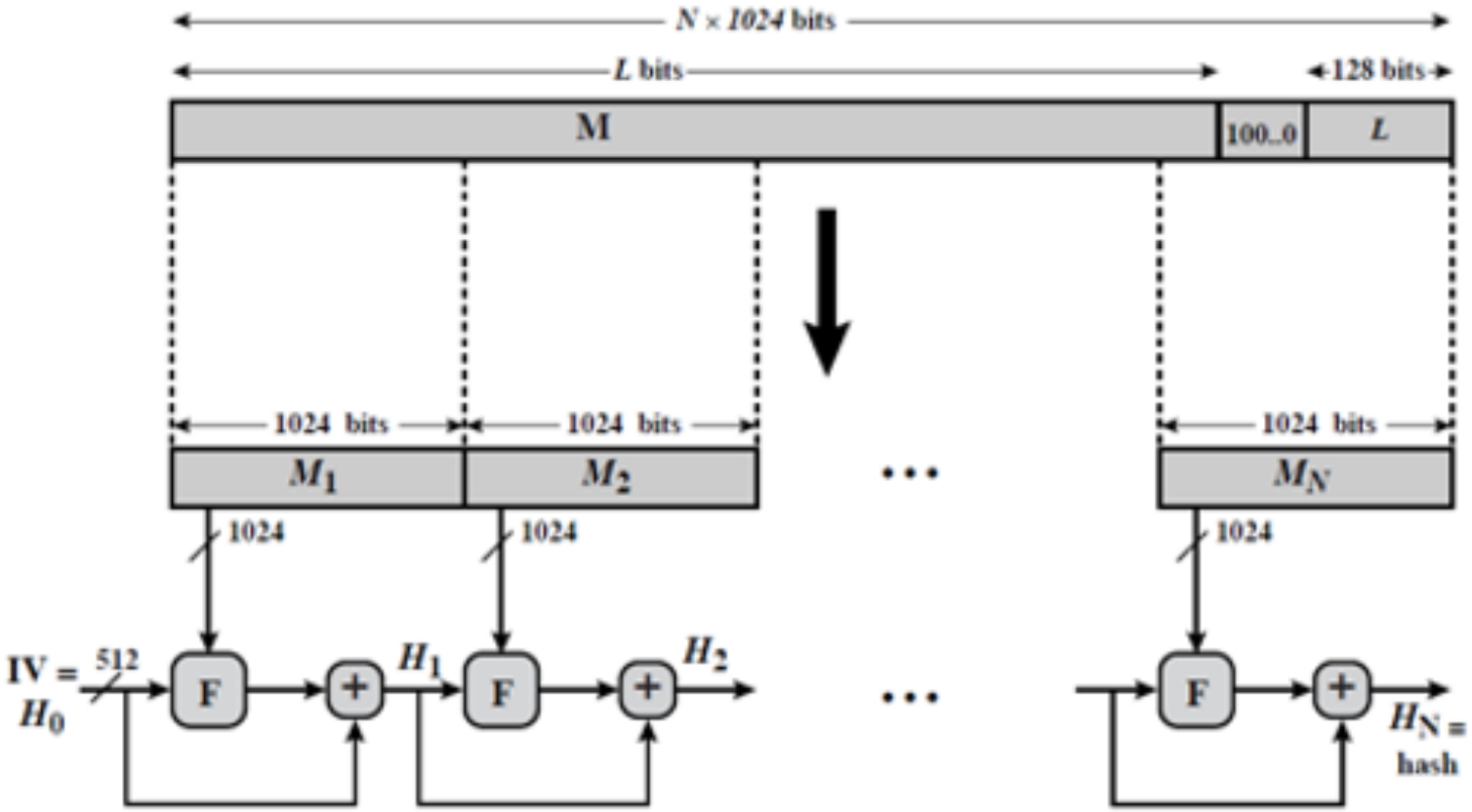
Padrões SHA

- 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.

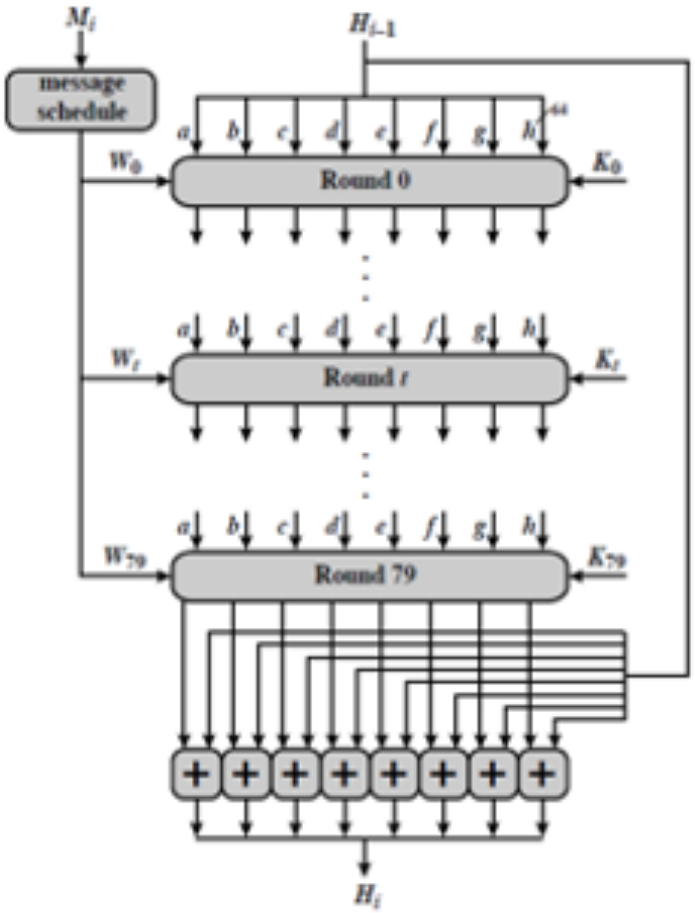
Comparação de funções SHA

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	

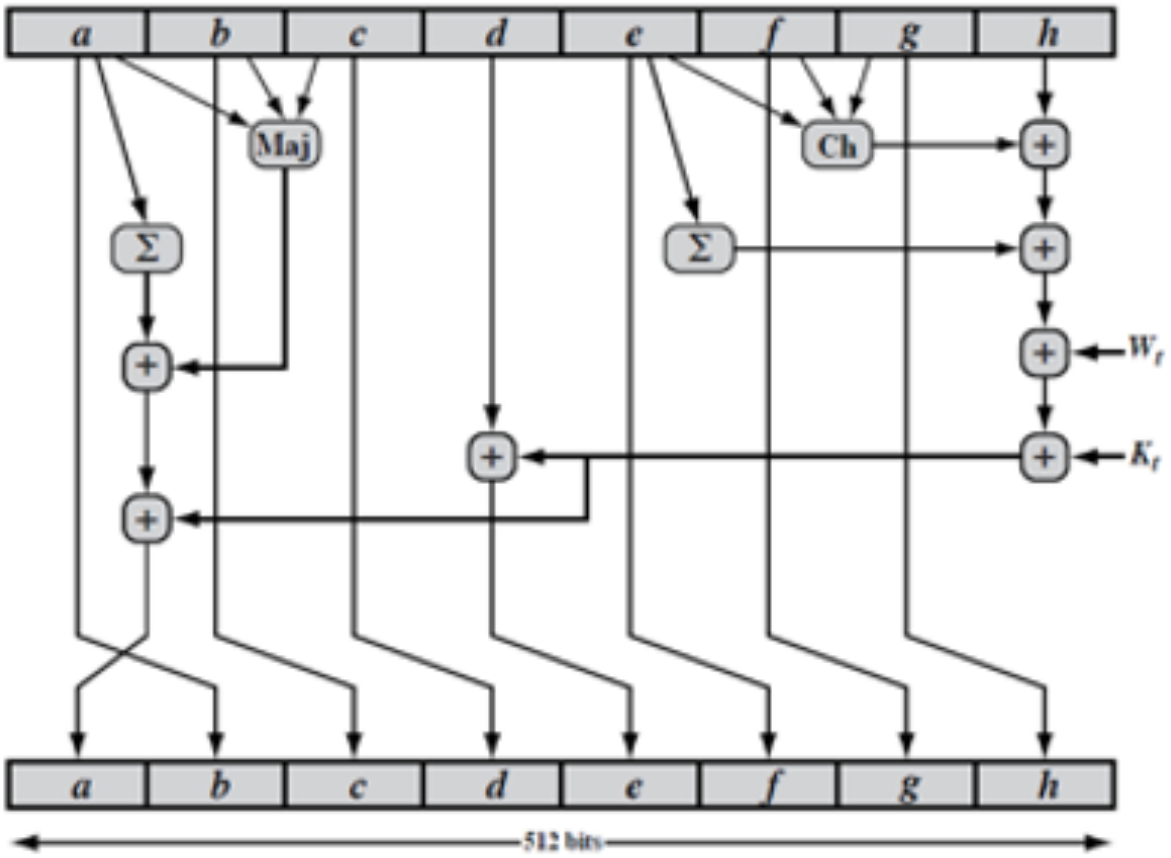
Estrutura do SHA-512



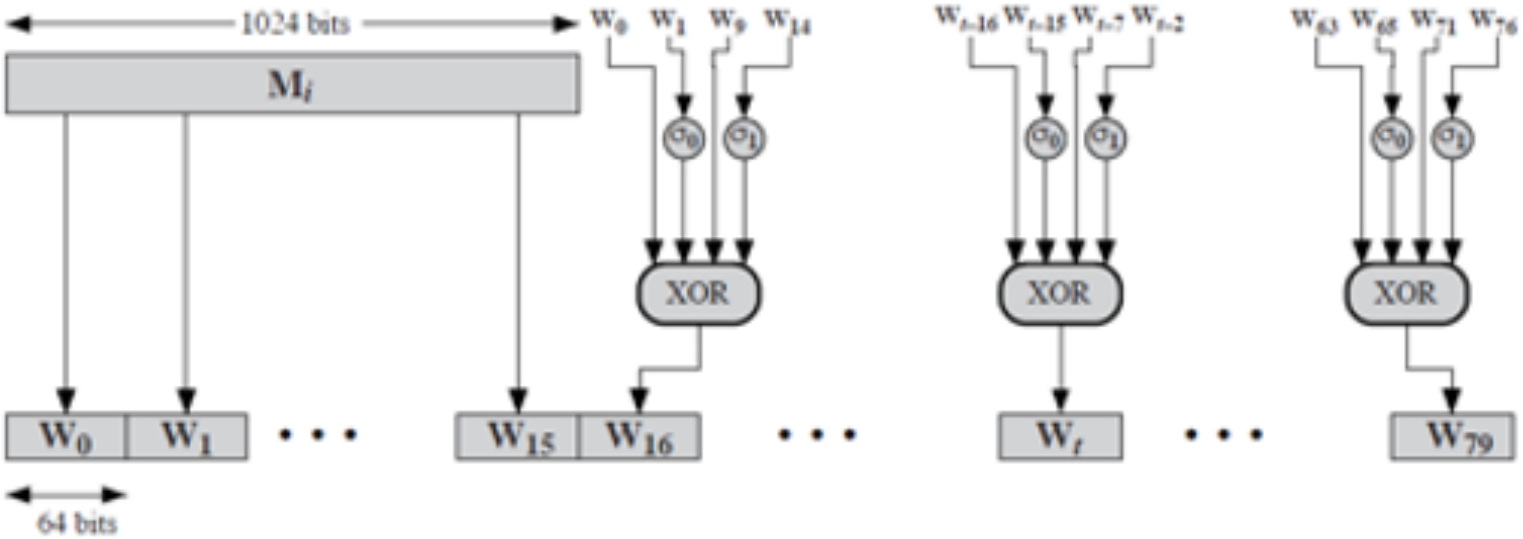
Estrutura do SHA-512



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

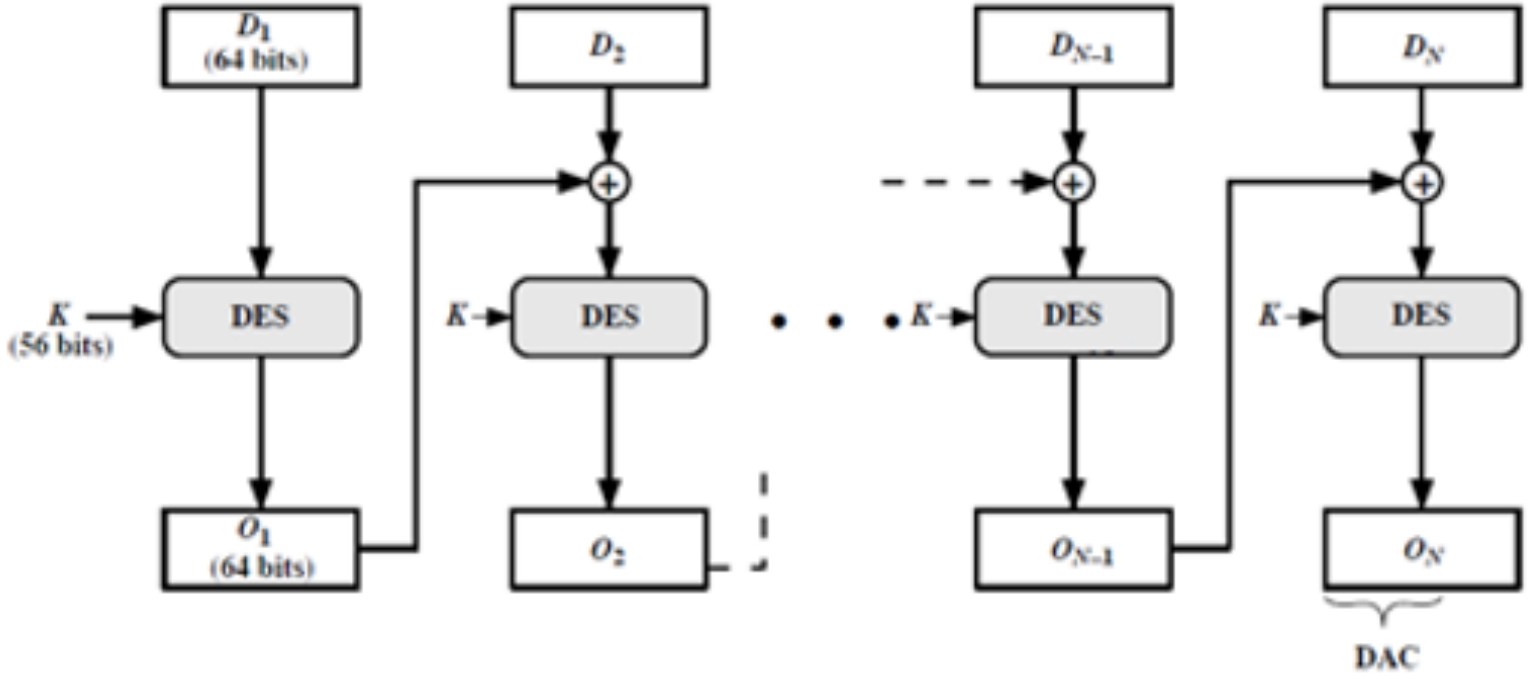
Requisitos de MAC

- 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
  - $Pr[MAC_k(m)=MAC_k(m')]=2^{-n}$
- Não-correlação
  - $Pr[MAC_k(m)=MAC_k(f(m))]= 2^{-n}$
  - 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: $D_1,...,D_n$
    - $O_1 = DES_k(D_1)$
    - $O_2 = DES_k(D_2)$
    - ...
    - $O_n = DES_k(D_{n-1}) \equiv MAC$

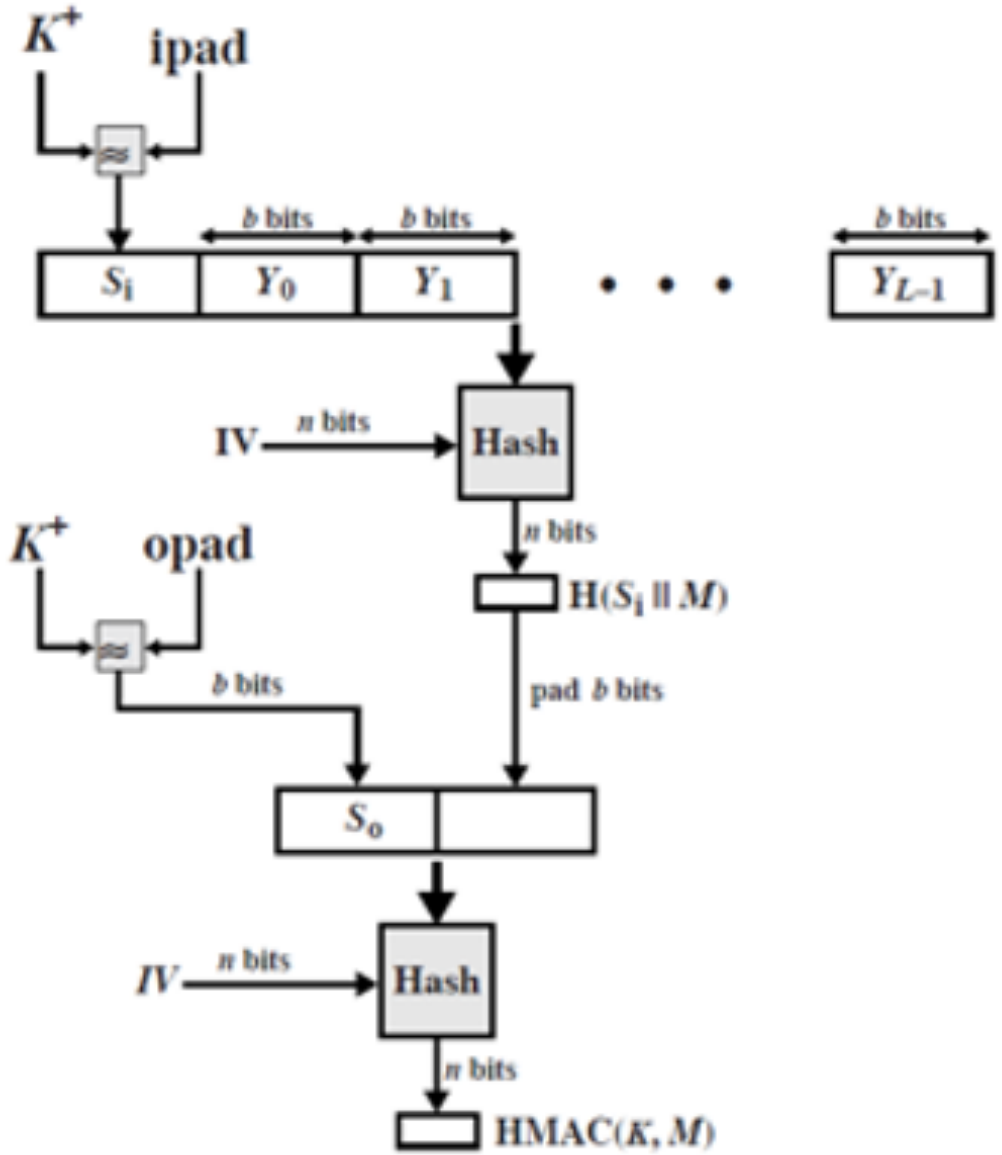
MAC baseado em uma cifra



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

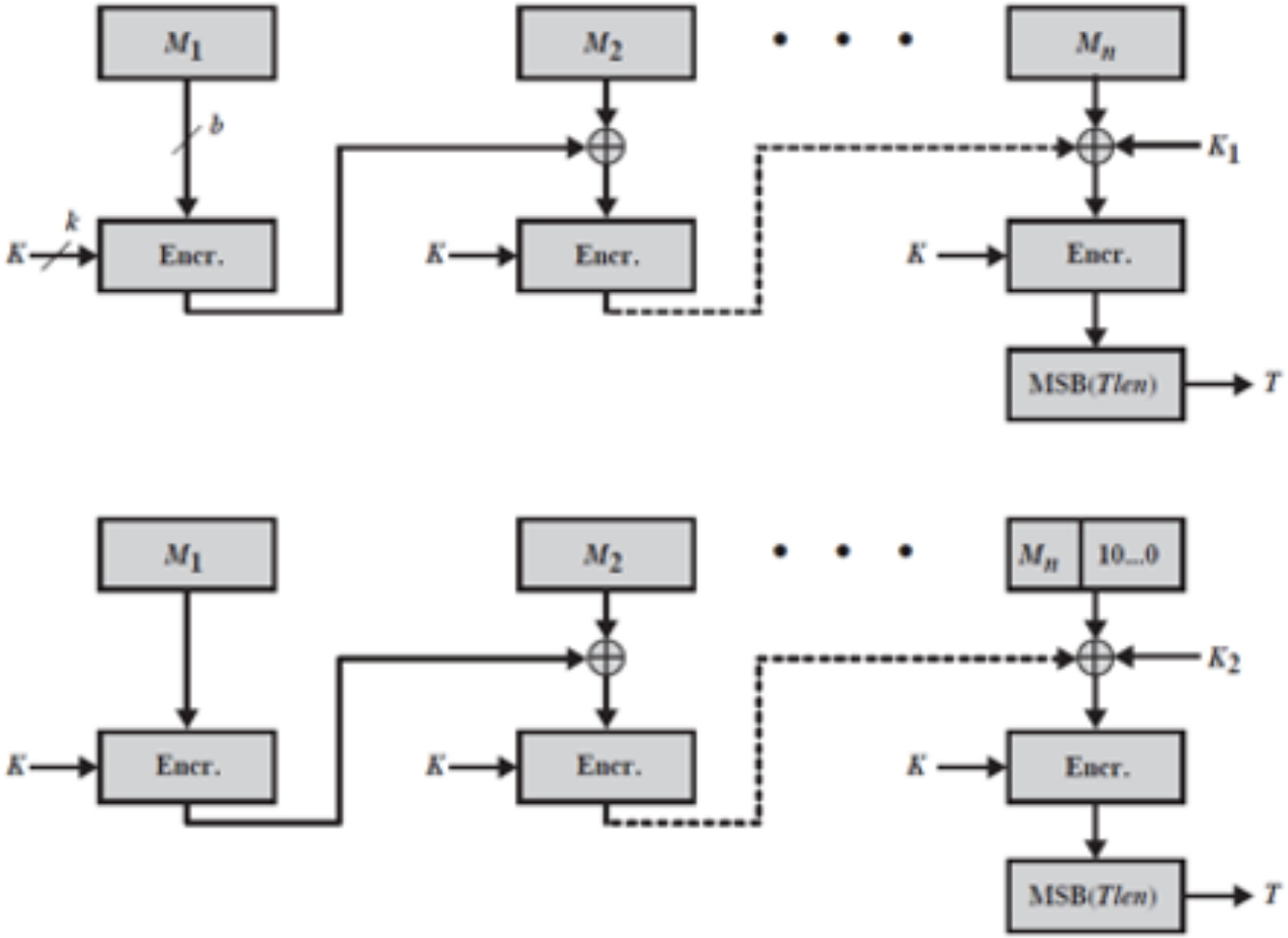
HMAC


K+
ipad
b bits
b bits
b bits
Si
Y0
Y1
YL−1
IV
n bits
Hash
n bits
K+
opad
H(Si || M)
b bits
pad b bits
So
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

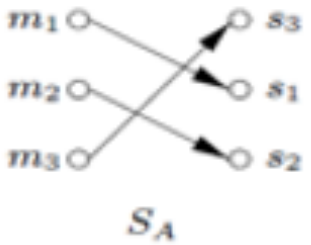
- Nomenclatura
  - $\mathcal{M}$ é o conjunto das mensagens que podem ser assinadas
  - $\mathcal{S}$ é o conjunto dos elementos chamados assinaturas, em geral strings de tamanho fixo
  - S_A é a transformação de assinatura da entidade A
    - leva uma mensagem a uma assinatura (deve ser mantida secreta 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.

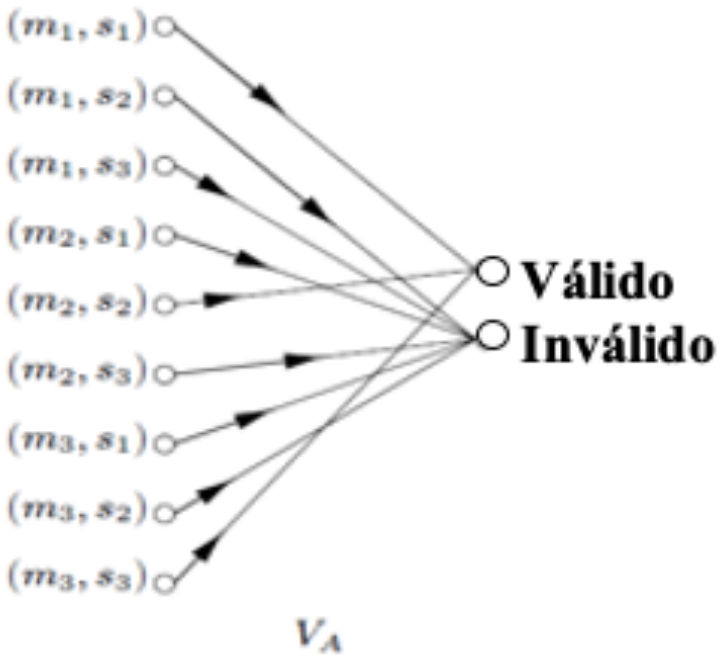
Funcionamento do mecanismo de assinatura

- 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

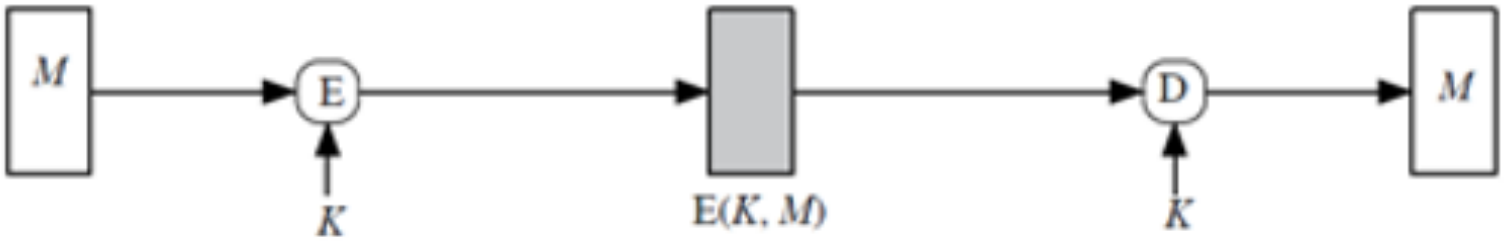
Digital Signature Standard (DSS)

- 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

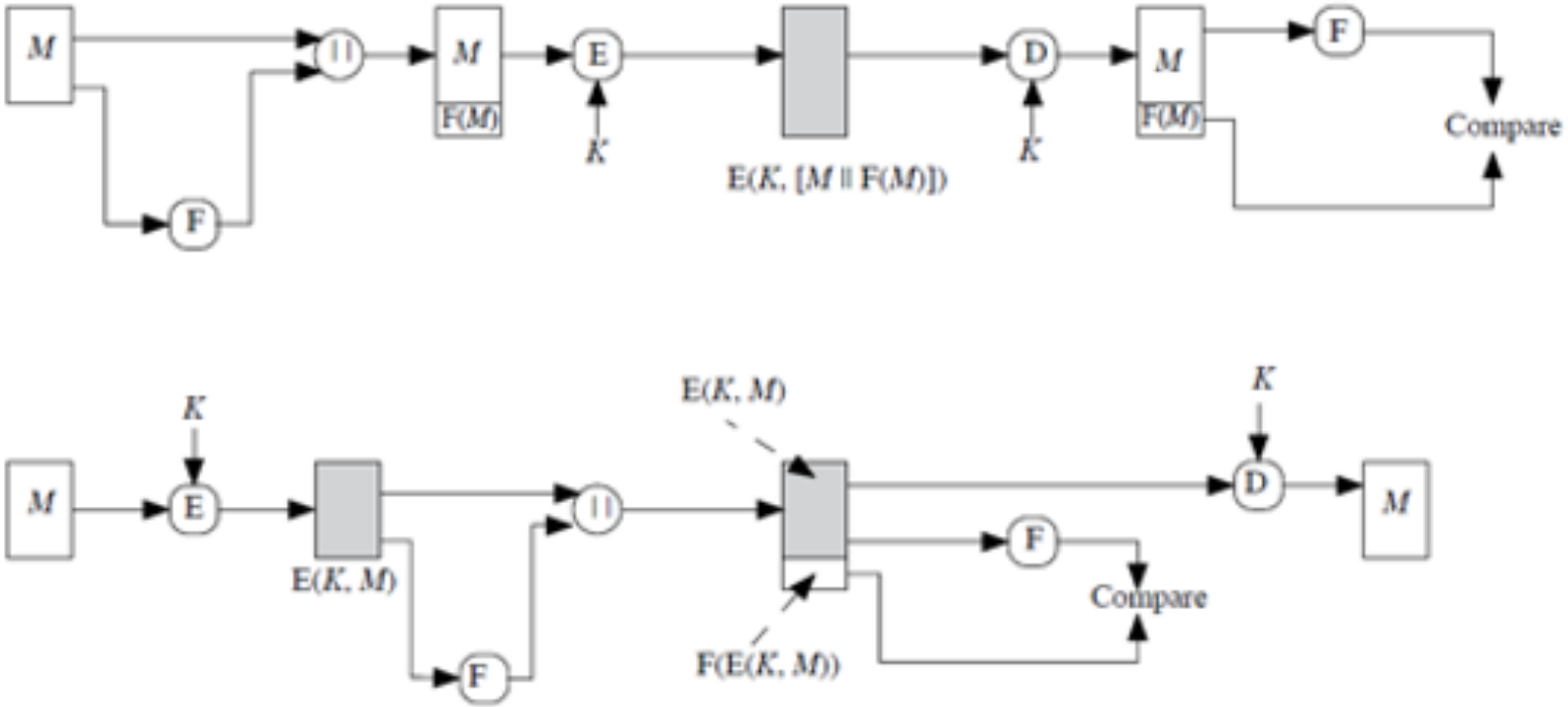
Protocolos básicos

Cifra de chave simétrica

› Confidencialidade (e autenticação)

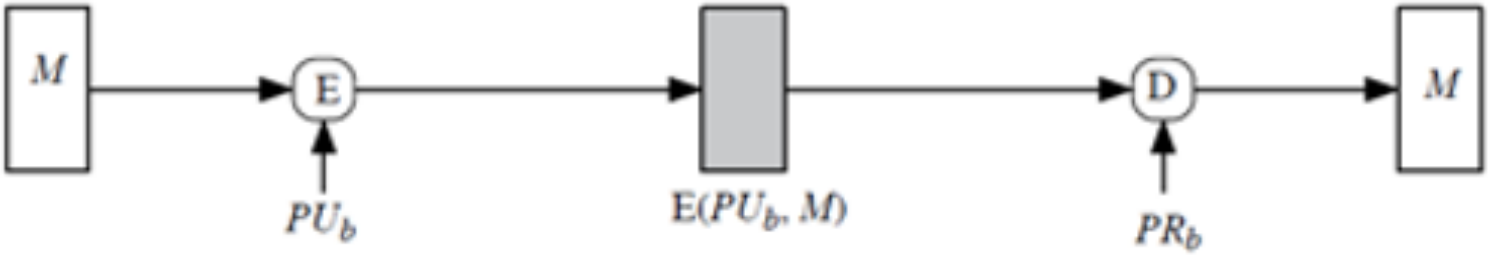


Controle de erros (interno/externo)



Cifra de chave pública

› Confidencialidade

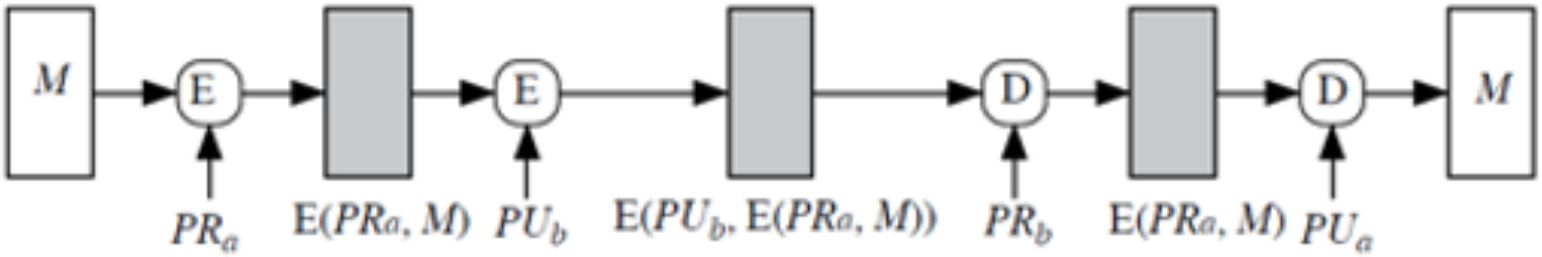


Cifra de chave pública

› Autenticação e assinatura

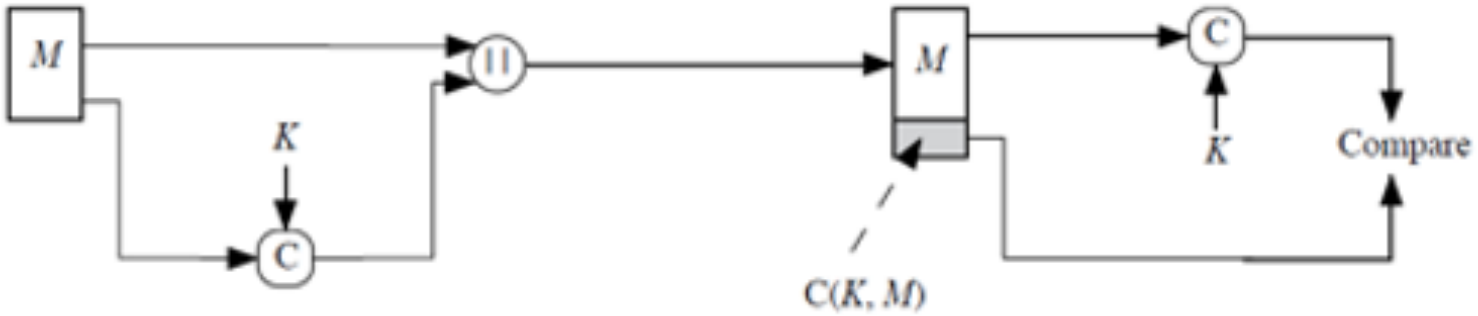
Cifra de chave pública

› Confidencialidade, autenticação e assinatura



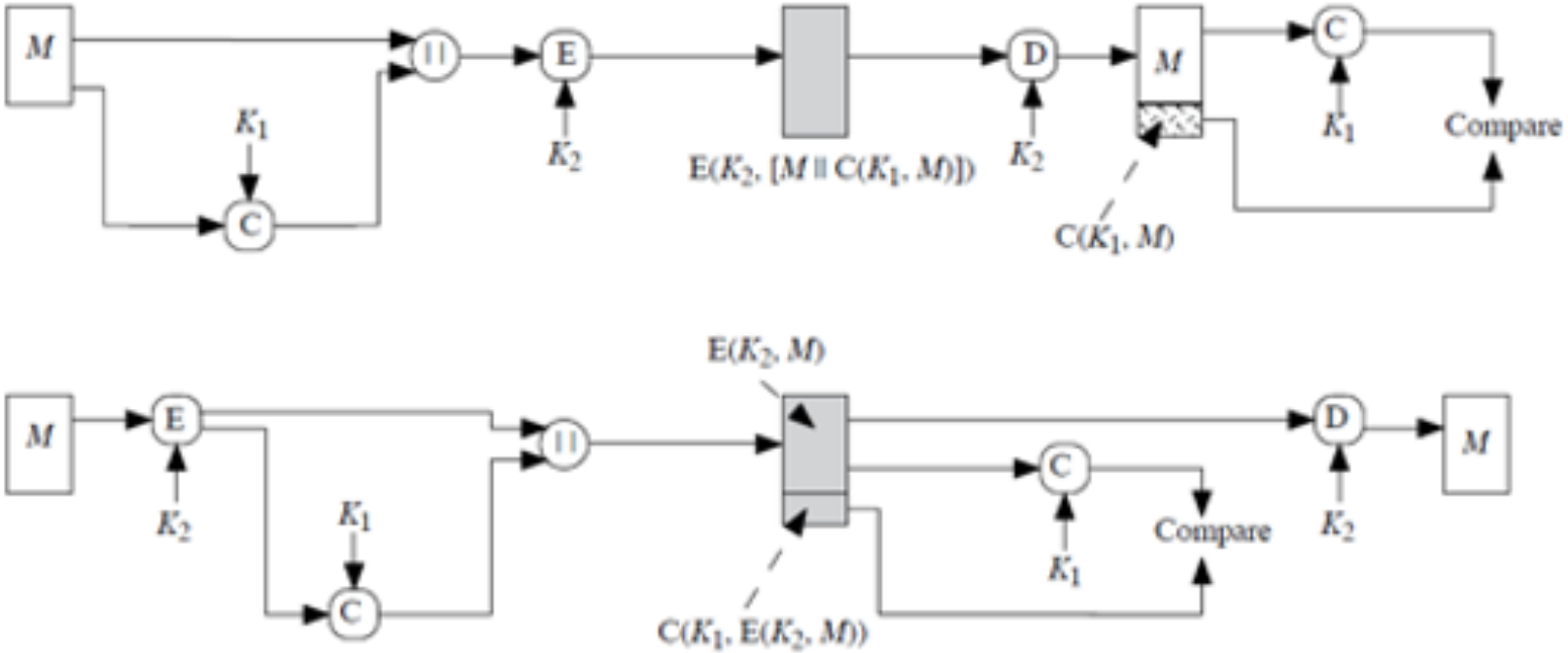
MAC

› Autenticação



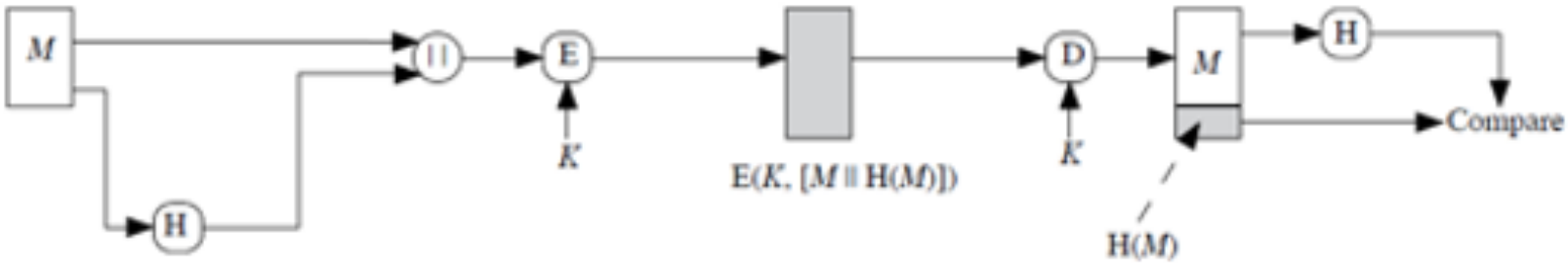
MAC

› Autenticação e confidencialidade



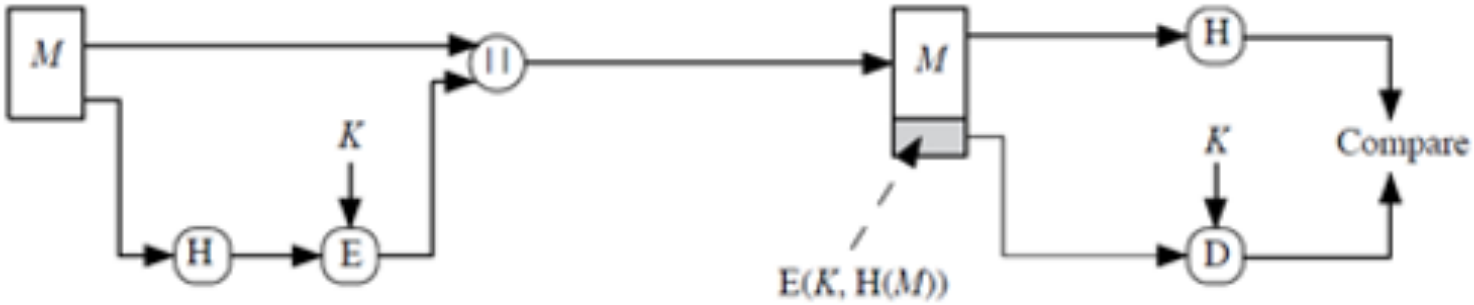
Hash

› Autenticação e confidencialidade



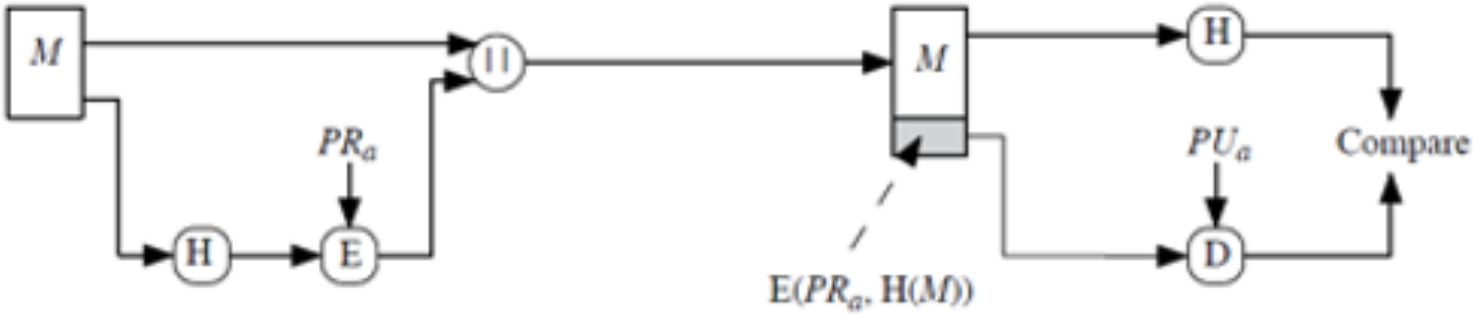
Hash

› Integridade



Hash

› Autenticação e assinatura

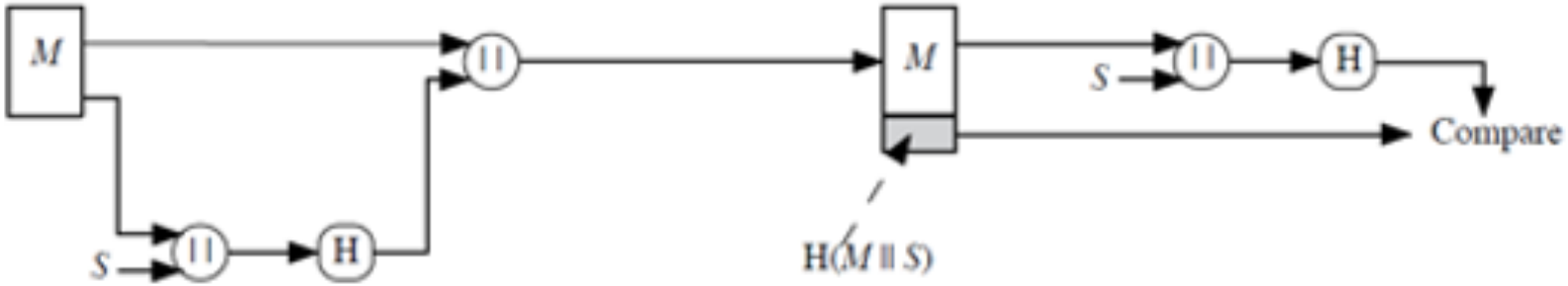


Hash

› Autenticação, assinatura e confidencialidade

Hash

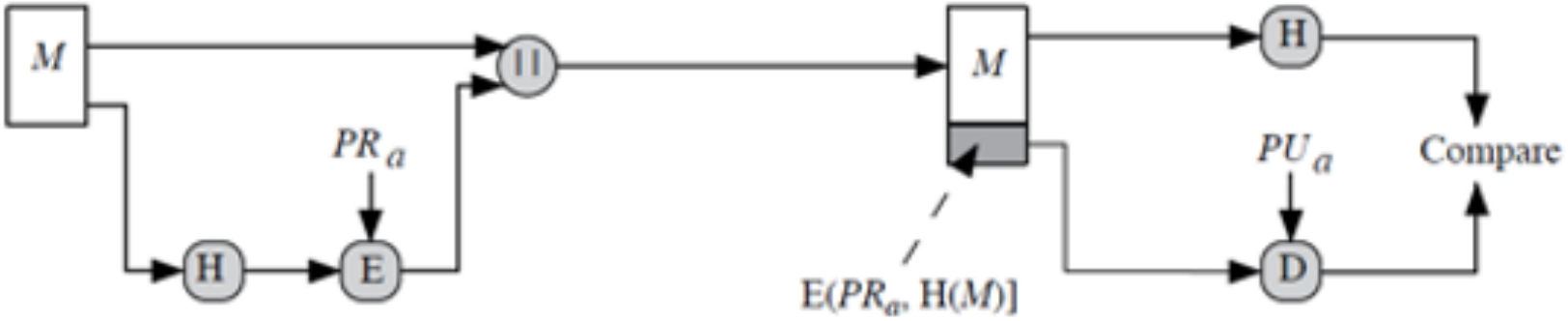
› Autenticação baseada em “segredo”



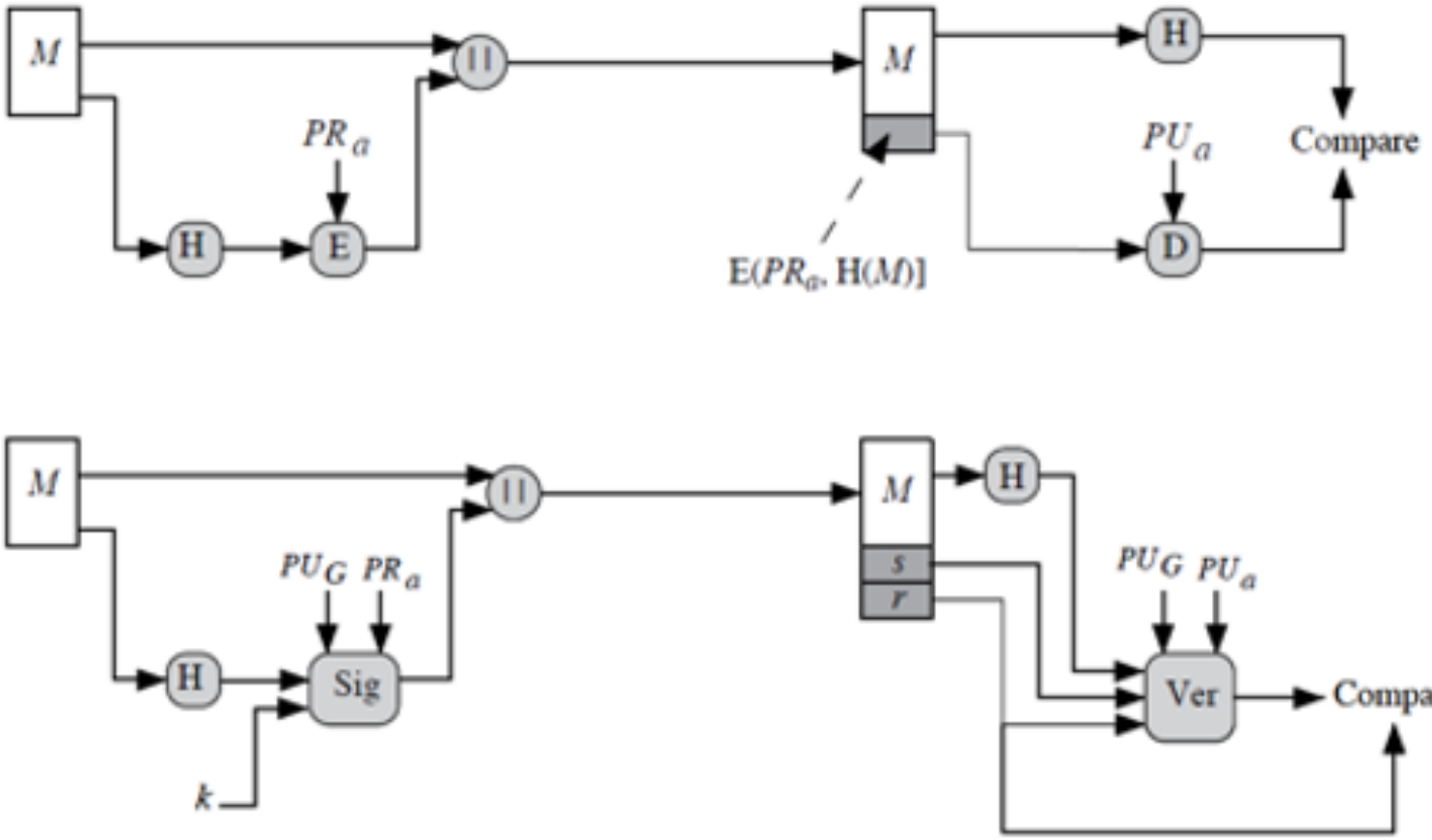
Hash

› Autenticação baseada em “segredo” e confidencialidade

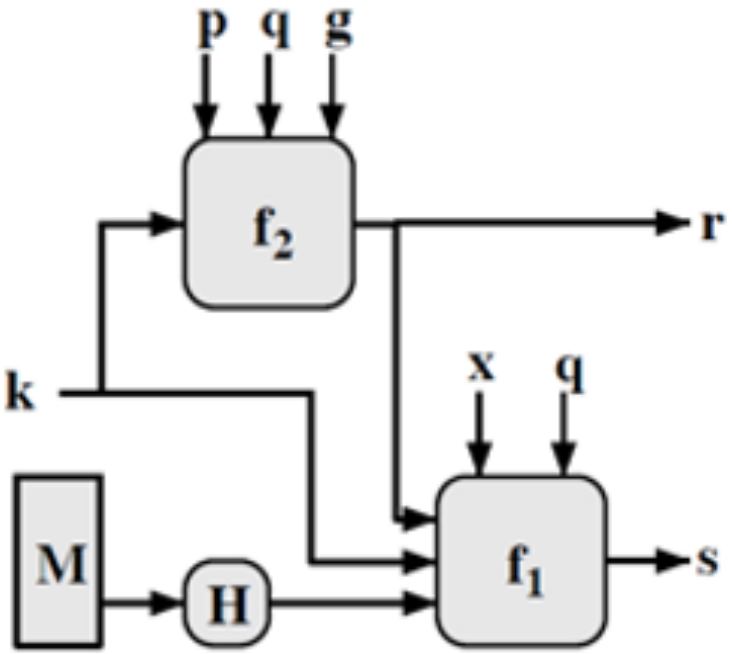
Assinatura baseada em hash + cifra assimétrica

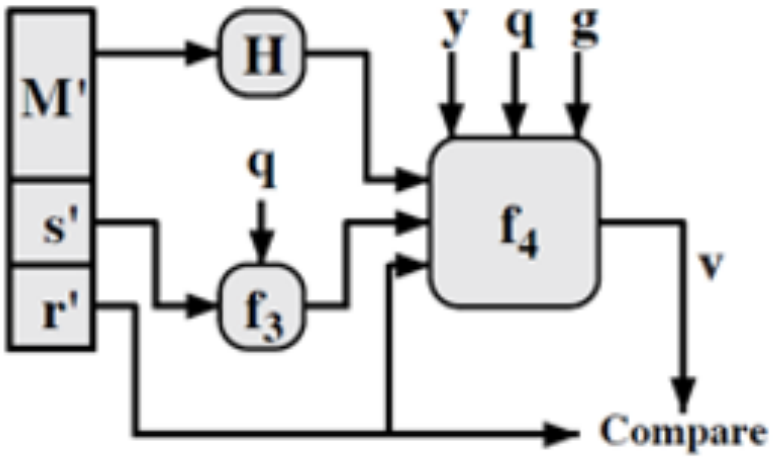


›DSS versus assinatura RSA



›Digital Signature Algorithm





$s = f_1(H(M), k, x, r, q) = (k^{-1} (H(M) + xr)) \bmod q$

$r = f_2(k, p, q, g) = (g^k \bmod p) \bmod q$

$w = f_3(s', q) = (s')^{-1} \bmod q$

$v = f_4(y, q, g, H(M'), w, r')$

$= ((g^{(H(M')w) \bmod q} y^{r'w \bmod q}) \bmod p) \bmod q$

›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.

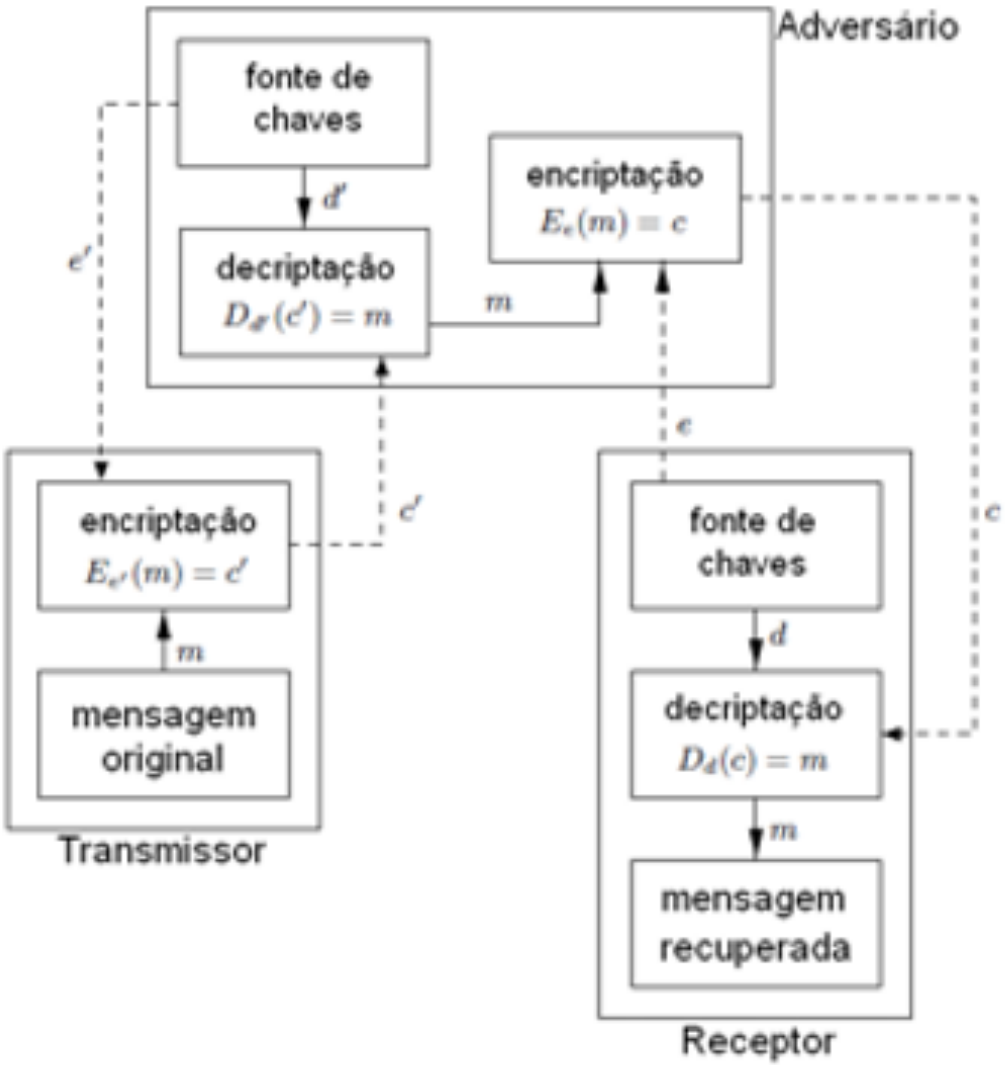
Exemplo de protocolo criptográfico

- 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

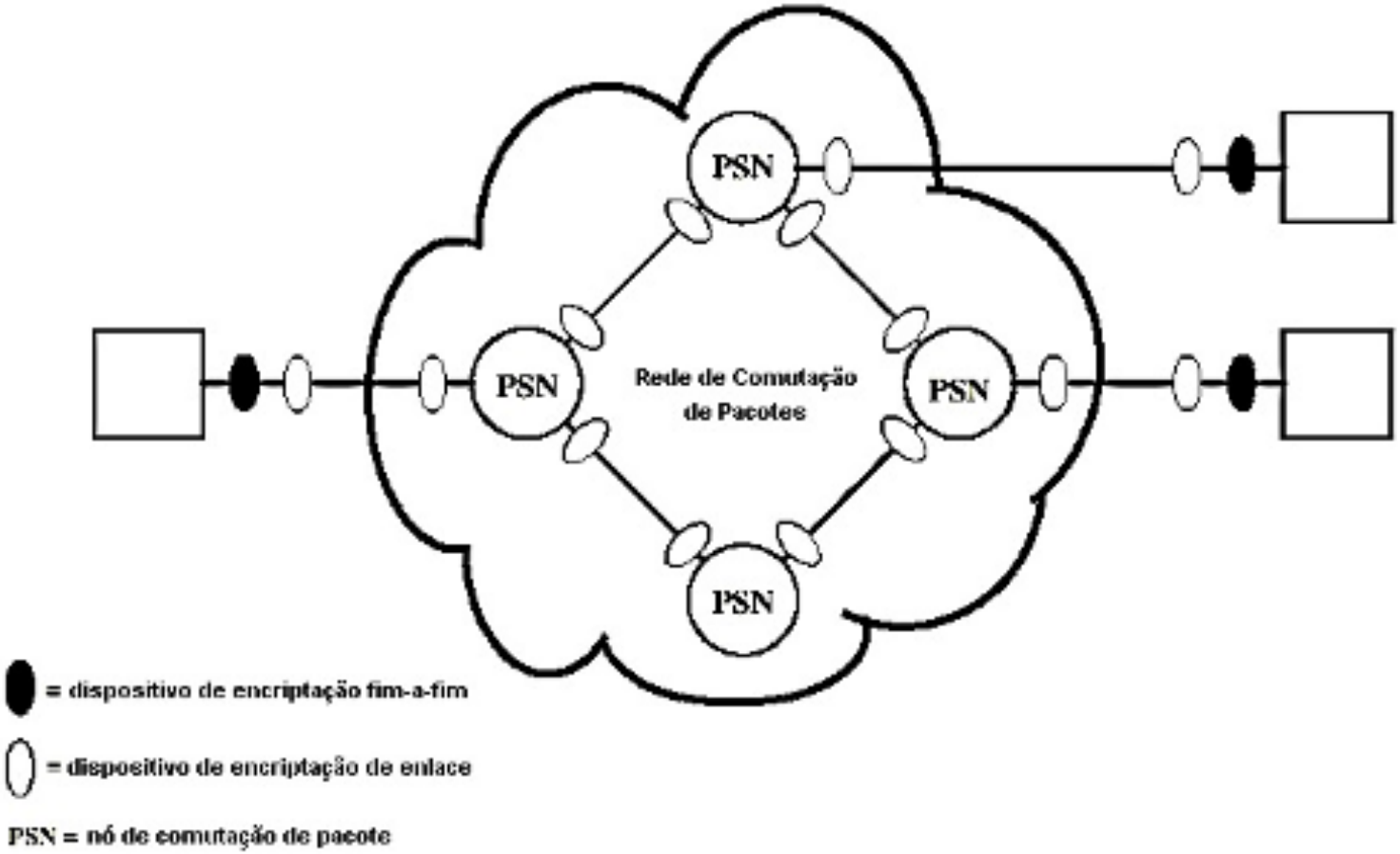
- Fraquezas de determinada primitiva criptográfica podem ser "ampliadas" por um protocolo ou mecanismo inconveniente
- Incorreto entendimento de algum princípio associado a determinada primitiva criptográfica

- 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

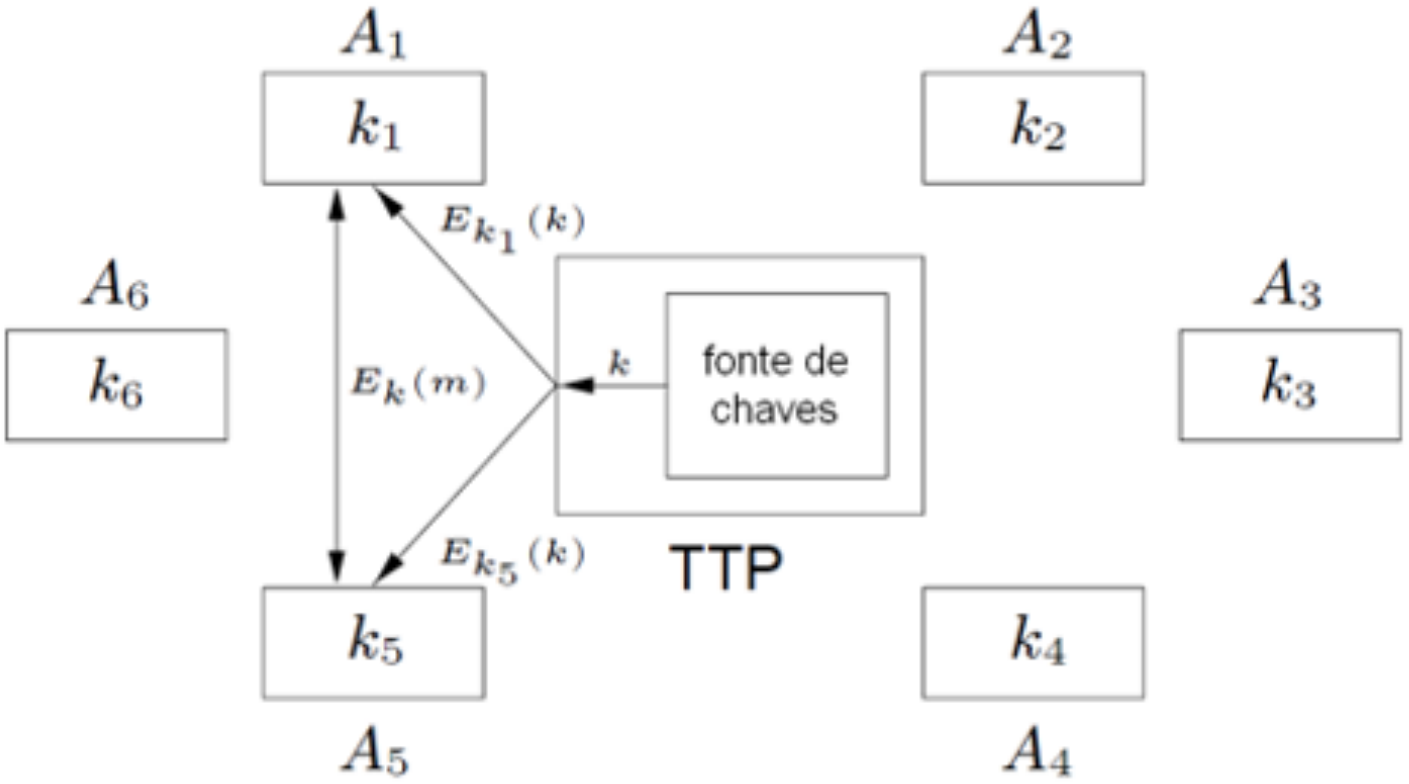
Gerenciamento de Chaves

- 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



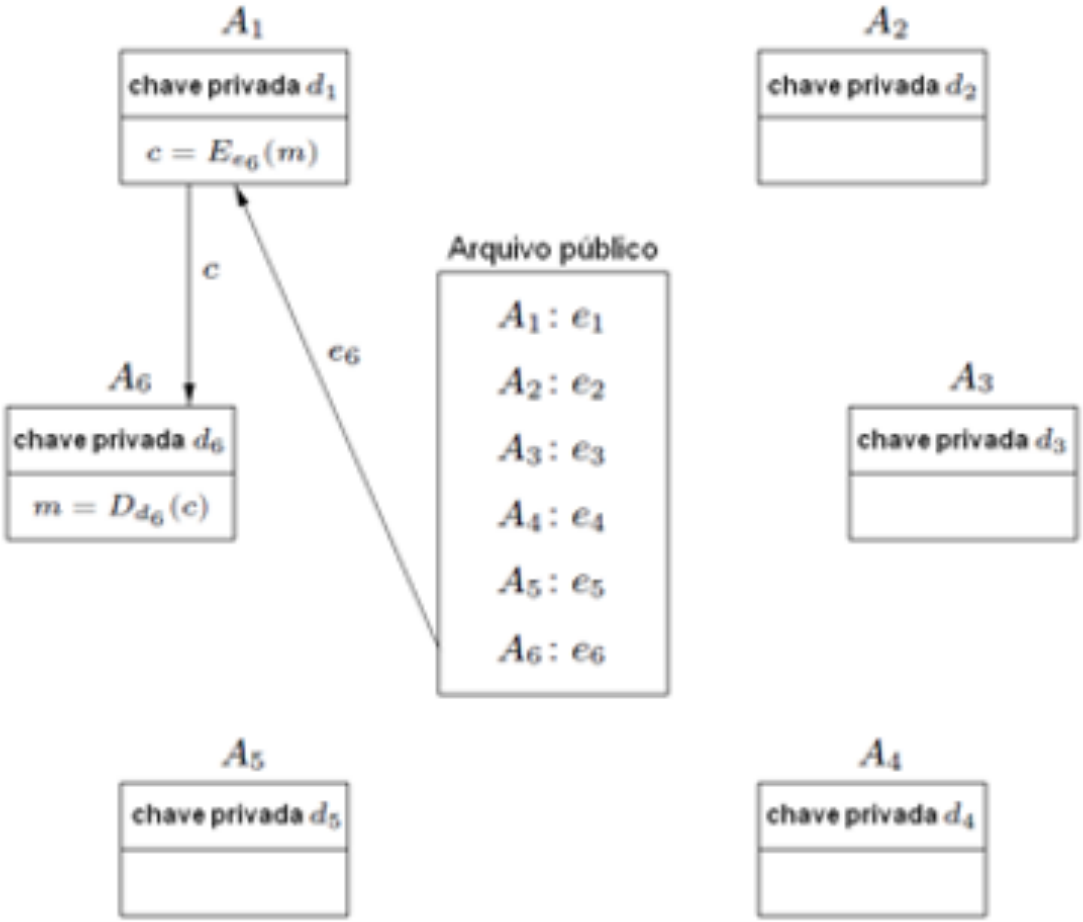
Gerenciamento de chaves com TTP

- 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

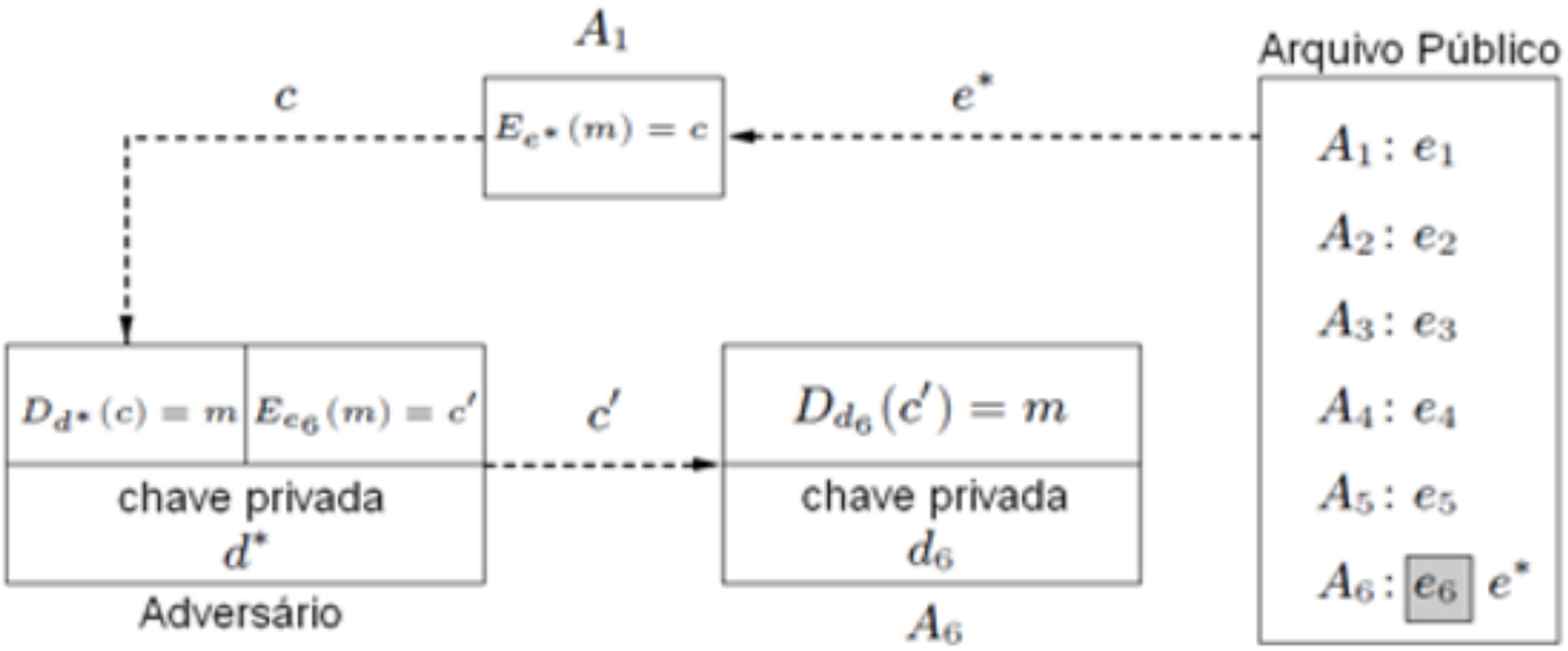


Gerenciamento de chaves em ambientes de chave pública

- 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

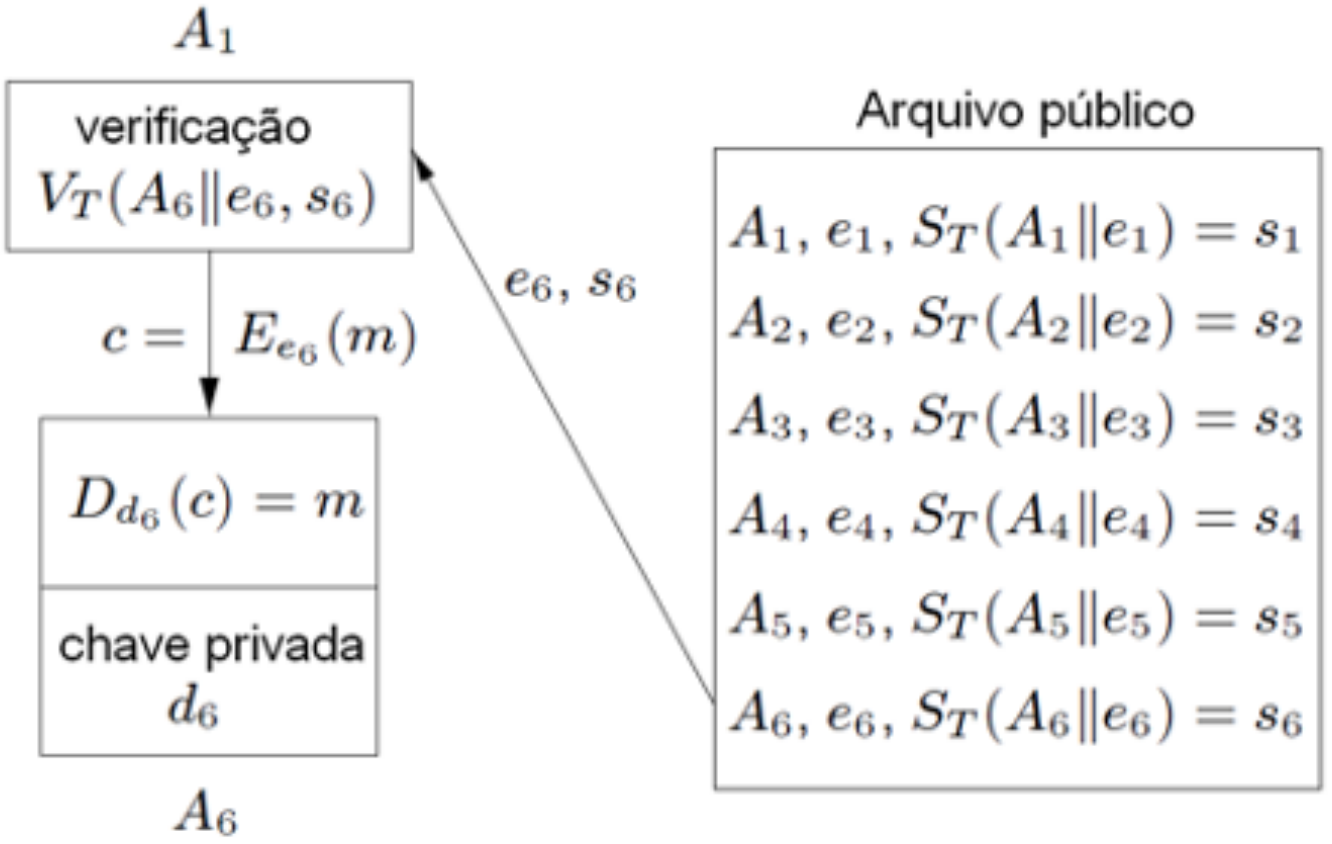
› Adversário pode alterar o arquivo público



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