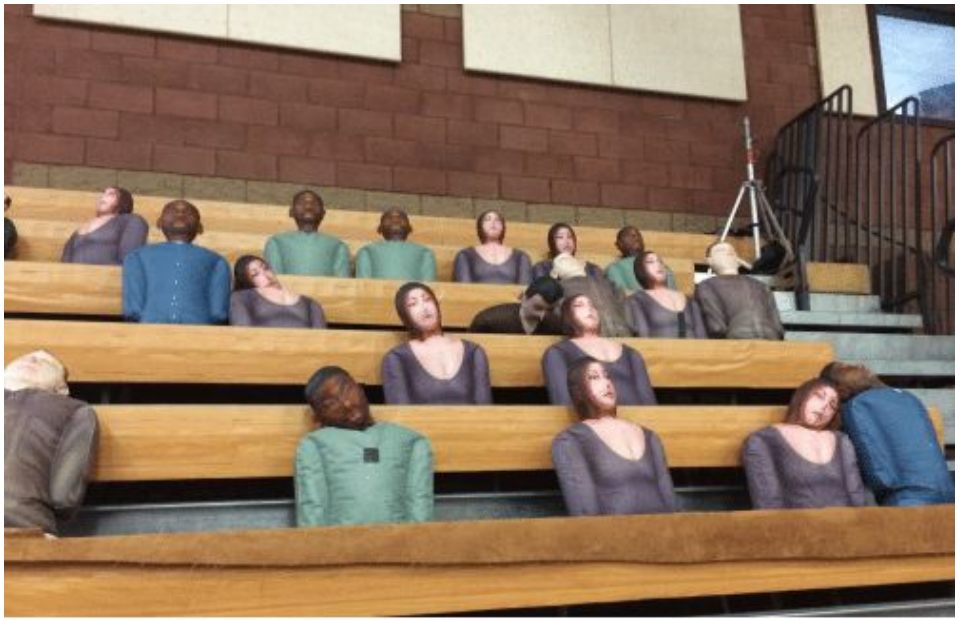
INE5429-07208 Segurança em Computação Criptografia Assimétrica e Integridade

Prof. Jean Everson Martina

O que vimos na aula passada:

- Modos de Operação
- Números Primos
- Teoremas de Fermat e Euler
- Testes de Primalidade
- Geradores de Números Aleatórios
- Logaritmo Discreto
- Propriedades de Criptosistemas de Chave Publica



Requisitos de Chave Pública

- Fácil (computacionalmente) gerar um par (de chaves)
- Fácil para o remetente operar com a chave pública
- Fácil para o destinatário operar com a chave privada
- Impossível determinar Kr a partir de Ku
- Impossível recuperar M conhecendo Ku e C

RSA

- 1977, Rivest, Shamir e Adelman / MIT
- É o algoritmo mais aceito
  - Base para a Web
  - Base para assinatura digital no Brasil
- Texto claro e texto cifrado são inteiros mod n
- n é normalmente 1024 bits (309 dígitos)
- É baseado em exponenciação mod p
- Algoritimo:
  - Blocos do tamanho de n
  - $C = M^e \bmod n$
  - $M = C^d \bmod n = ((M^e)^d) \bmod n = M^{ed} \bmod n$
  - Todos conhecem n, o remetente conhece e, o destinatário conhece d
  - Chave Pública → (n, e)
  - Chave Privada → (n, d)

RSA - Requisitos

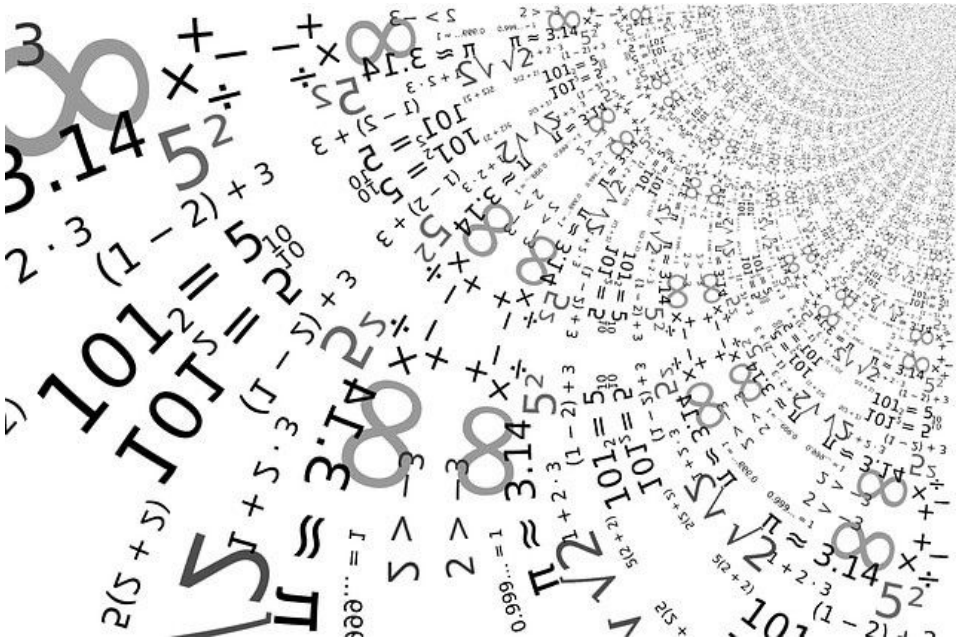
- e, d, n são escolhidos para satisfazer M^{ed} mod n = M para todo M < n
- Para isso “e” e “d” devem ser multiplicativas inversas mod ϕ(n) → e.d mod ϕ(n) = 1
  - e.d ≡ 1 mod ϕ(n) → d ≡ e-1 mod ϕ(n)
  - gcd(ϕ(n),d)=1 e gcd(ϕ(n),e)=1
- p, q primos: privados e escolhidos
- n = p.q: publico e calculado
- e | gcd(ϕ(n),e) = 1 ^ 1 < e < ϕ(n): publico e calculado
- $d \equiv e^{-1} (\text{mod } \phi(n))$
- Chave pública (e, n)
- Chave privada (d, n)

RSA na Prática

- Geração de Chaves
  - p = 17 e q = 11
  - n = porque = 17 x 11 = 187
  - $\phi(n) = (p - 1)(q - 1) = 16 \times 10 = 160$
  - e = 7, gcd(160, 7) = 1 ˆ 1 < 7 < 160
  - d | de ≡ 1(mod 160) ˆ d < 160 → d = 23
  - 23 x 7 = 161
  - Ku = {7, 187} , Kr = {23, 187}

- Cifragem
  - Texto Claro = 88
  - 887 mod 187 = 11
  - Texto cifrado = 11
  - 1123 mod 187 = 88
  - Computacionalmente intensivo de fazer com números grande
  - Teorema chinês do resto torna possível

RSA - Considerações Computacionais



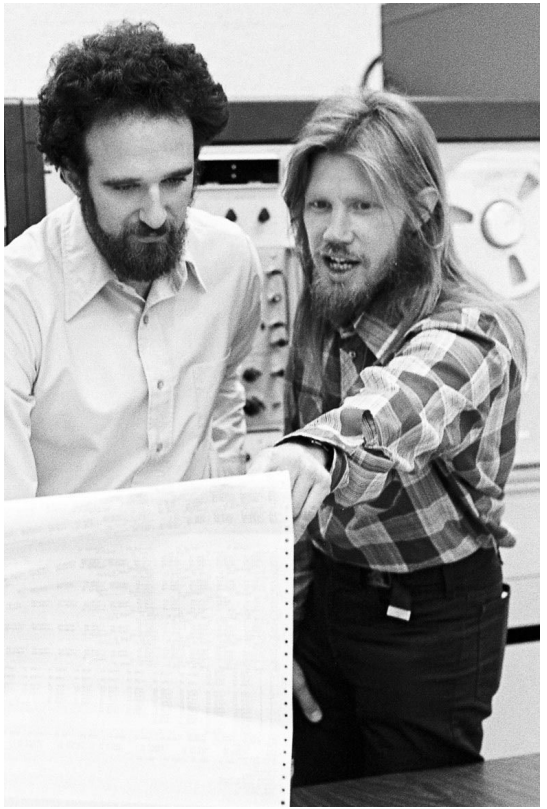
- Exponenciação mod p requer truques matemáticos
- O e acaba sendo fixo em 65537 e 17. O Número 3 sofre ataques se utilizado muitas vezes
- d tem que ser grande para evitar força bruta
- Gerar chaves pode ser demorado pois precisamos do teste de primalidade várias vezes em um número muito grande

RSA Factoring Challenge

RSA number	Decimal digits	Binary digits	Cash prize offered	Factored on	Factored by
RSA-100	100	330	US$1,000[4]	April 1, 1991[5]	Arjen K. Lenstra
RSA-110	110	364	US$4,429[4]	April 14, 1992[5]	Arjen K. Lenstra and M.S. Manasse
RSA-120	120	397	$5,898[4]	July 9, 1993[6]	T. Denny *et al.*
RSA-129 [**]	129	426	US$100	April 26, 1994[5]	Arjen K. Lenstra *et al.*
RSA-130	130	430	US$14,527[4]	April 10, 1996	Arjen K. Lenstra *et al.*
RSA-140	140	463	US$17,226	February 2, 1999	Herman te Riele *et al.*
RSA-150	150	496		April 16, 2004	Kazumaro Aoki *et al.*
RSA-155	155	512	$9,383[4]	August 22, 1999	Herman te Riele *et al.*
RSA-160	160	530		April 1, 2003	Jens Franke *et al.*, University of Bonn
RSA-170 [*]	170	563		December 29, 2009	D. Bonenberger and M. Krone [***]
RSA-576	174	576	US$10,000	December 3, 2003	Jens Franke *et al.*, University of Bonn
RSA-180 [*]	180	596		May 8, 2010	S. A. Danilov and I. A. Popovyan, Moscow State University[7]
RSA-190 [*]	190	629		November 8, 2010	A. Timofeev and I. A. Popovyan
RSA-640	193	640	US$20,000	November 2, 2005	Jens Franke *et al.*, University of Bonn
RSA-200 [*] ?	200	663		May 9, 2005	Jens Franke *et al.*, University of Bonn
RSA-210 [*]	210	696		September 26, 2013[8]	Ryan Propper
RSA-704 [*]	212	704	US$30,000	July 2, 2012	Shi Bai, Emmanuel Thomé and Paul Zimmermann
RSA-220 [*]	220	729		May 13, 2016	S. Bai, P. Gaudry, A. Kruppa, E. Thomé and P. Zimmermann
RSA-230 [*]	230	762		August 15, 2018	Samuel S. Gross, Noblis, Inc.
RSA-232	232	768			
RSA-768 [*]	232	768	US$50,000	December 12, 2009	Thorsten Kleinjung *et al.*
RSA-240	240	795			
RSA-250	250	829			
RSA-260	260	862			
RSA-270	270	895			
RSA-896	270	896	US$75,000		
RSA-280	280	928			
RSA-290	290	962			
RSA-300	300	995			
RSA-309	309	1024			
RSA-1024	309	1024	US$100,000		

Troca de Chaves Diffie-Hellman

- Primeiro algoritmo publicado de chave pública
- Sozinho é suscetível a ataque MITM
- Objetivo: Troca segura de parâmetros para estabelecer uma chave de sessão
- O algoritmo depende da dificuldade de calcular logaritmos discretos
- Raiz primitiva → $a \bmod p \ldots a^{p-1} \bmod p$
- $b \equiv a^i \pmod p$ onde $0 \leq i \leq p \rightarrow \text{dlog}_{a,p}(b)$

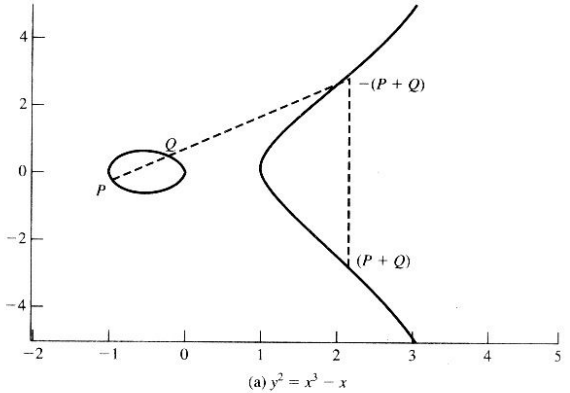


Diffie-Hellman - Algoritmo

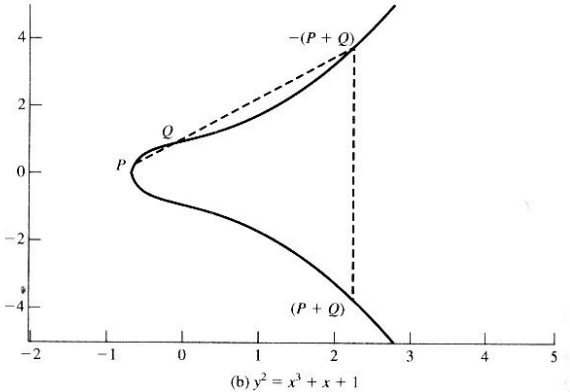
- Parâmetros:
  - q numero primo, α raiz primitiva de q → públicos
  - Xa e Xb números aleatórios < q
  - Geração de chave:
  - $Ya = \alpha^{Xa} \bmod q$ e $Yb = \alpha^{Xb} \bmod q$
- Segredo:
  - $K = (Yb)^{Xa} \bmod q$
  - $K = (Ya)^{Xb} \bmod q$
- O adversários só sabe q, α ,Ya e Yb

- q = 353, α = 3, Xa = 97 e Xb = 233
- A computa:
  - $Ya = 3^{97} \bmod 353 = 40$
- B computa:
  - $Yb = 3^{233} \bmod 353 = 248$
- A deriva:
  - $K = 248^{97} \bmod 353 = 160$
- B deriva:
  - $K = 40^{233} \bmod 353 = 160$

Curvas Elípticas



(a) $y^2 = x^3 - x$



(b) $y^2 = x^3 + x + 1$

- Resposta ao tamanho de chaves RSA
  - Custo computacional crescente
- Mesma segurança com chaves menores
- Menos processamento
- Teoria antiga, pratica nova
- Pouca criptoanálise → Menos confiança
- Muito mais difícil de entender e explicar
- Cifradores de chave pública são baseados em grupos abelianos (ex. Diffie-Hellman)
  - CE tem adição e multiplicação
- Multiplicação é repetição de adição
- É uma equação com duas variáveis e coeficientes
- Restritos a elementos de um corpo finito
- Resulta em grupos abelianos finitos

Criptografia em Curvas Elípticas

- ECDH – Elliptic Curve Diffie-Hellman
- ECIES – Elliptic Curve Integrated Encryption Scheme
- ECDSA – Elliptic Curve Digital Signature Algorithm
- É fácil de converter algoritmos que usem logaritmo discreto como problema base
    - Pois o requisito é o grupo abeliano

Elliptic-Curve Digital Signature Algorithm (ECDSA)

NIST Guidelines for Public Key Sizes for AES			
ECC key size (bits)	RSA key size (bits)	Key size ratio	AES key size (bits)
163	1,024	1:6	
256	3,072	1:12	128
384	7,680	1:20	192
512	15,360	1:30	256

Supplied by NIST to ANSIX9F1

Table 1

Autenticação de Mensagens

- Garantia de que a mensagem esta integra e que foi enviada por alguém válido
- Cifragem garante autenticação
- Assinatura eletrônica garante autenticação de mensagens
- Ataques:
  - Mascaramento:
    - Origem fraudulenta
  - Modificação de conteúdo:
    - Alteração da carga da mensagem
  - Modificação de seqüência:
    - Reordenamento de mensagens
  - Modificação de tempo:
    - Replay e prevenção de entrega

Funções de Autenticação

- Autenticação acontece em dois níveis
  - Autenticador
  - Alto nível
- Autenticadores:
  - Cifragem
  - Message Authentication Codes
  - Funções HASH
- Alto nível → Protocolos criptográficos

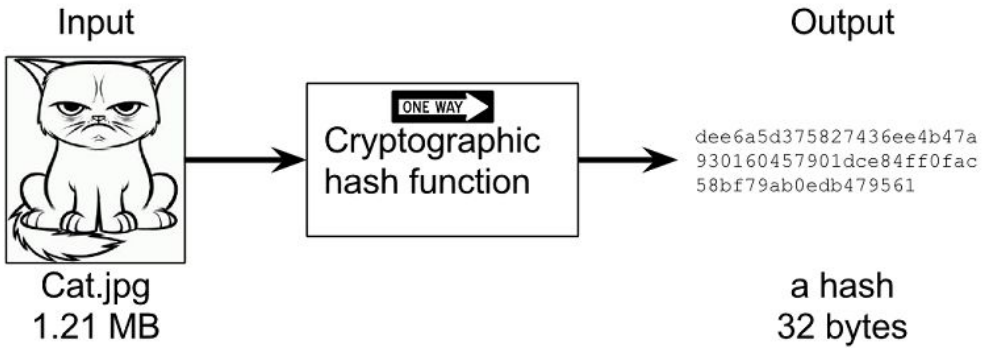
Autenticação - Cifragem

- Provê autenticação usando algoritmos criptográficos
- Autenticação por cifragem pode ser dividida em:
  - Simétrica
    - Somente A e B compartilha a chave K
  - Assimétrica
    - Autenticação pelo uso da chave privada
- A autenticação é baseada na manutenção dos segredos da chaves que devem ser protegidas

Autenticação – Códigos de Autenticação de Mensagens

- Resumo da mensagem baseado em chave simétrica
- MAC = C(K,M)
- Calcula-se dos dois lados usando os mesmos parâmetros para confirmar
- É similar a cifragem mas não tem reversão
- Quando Usar:
  - Mensagem enviada a vários destinatários
  - Não é possível decifrar tudo então usa-se checagem MAC seletiva
  - Verificação de integridade de programas
  - Não é necessário sigilo

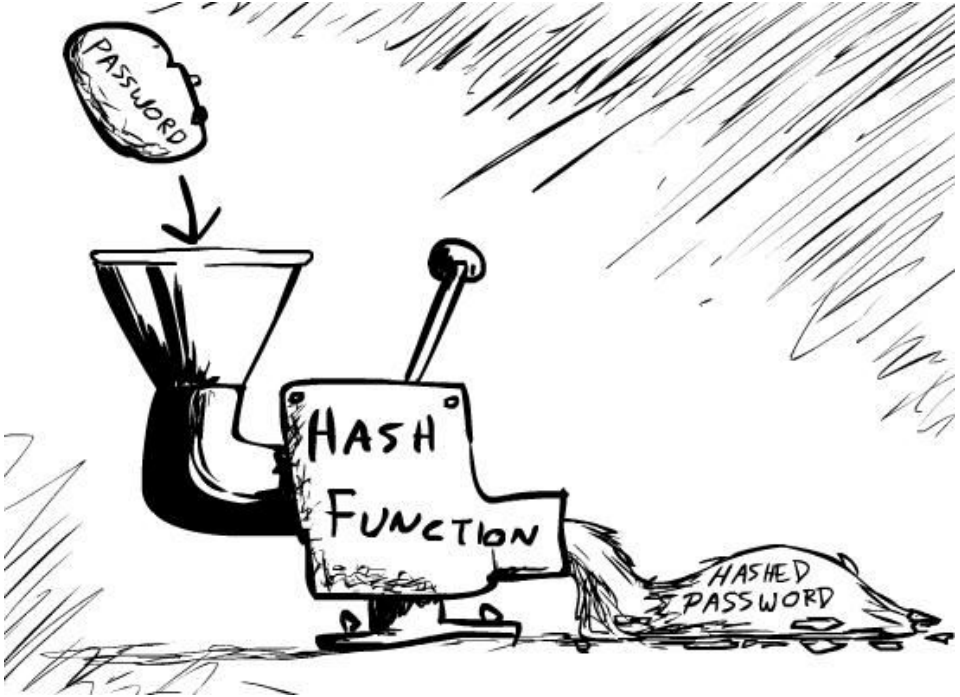
Integridade – Funções HASH



- São similares a MAC mas não tem chaves
- Prove propriedades como efeito avalanche
- Garante que o homomorfismo da mensagem não afeta a assinatura
- Provê uma camada de integridade diferente da autenticação
- Quando Usar:
  - Cifragem é lerdo, então HASH é mais eficiente só para integridade
  - Cifragem pode ser computacionalmente caro em software e em hardware
  - Quando o algoritmo criptográfico tem problemas com homomorfismo
    - ex. RSA

HASH – Descrição/Requisitos

- Função de caminho único, M variável, H(M) Fixo
- Produz uma impressão digital de um arquivo
- Requisitos:
  - Fácil de computar para qualquer M
  - É impossível achar M tendo H(M) → caminho unico
  - H(y)=H(x) ˆ x≠y impossível → 1a. Pre-imagem
  - Achar (x,y) | H(x)=H(y) impossível → 2a. Pre-imagem



Paradoxo do Aniversário

- Um grupo maior que 23 pessoas têm probabilidade maior que 50% de terem a mesma data de aniversário.
- $2^{m/2}$ variações da mensagem com o mesmo significado → 2m/2 variações fraudulentas
- A probabilidade de sucesso é maior que 50%
- Se oferece a versão fraudulenta e se usa a versão variada.

Dear Anthony,

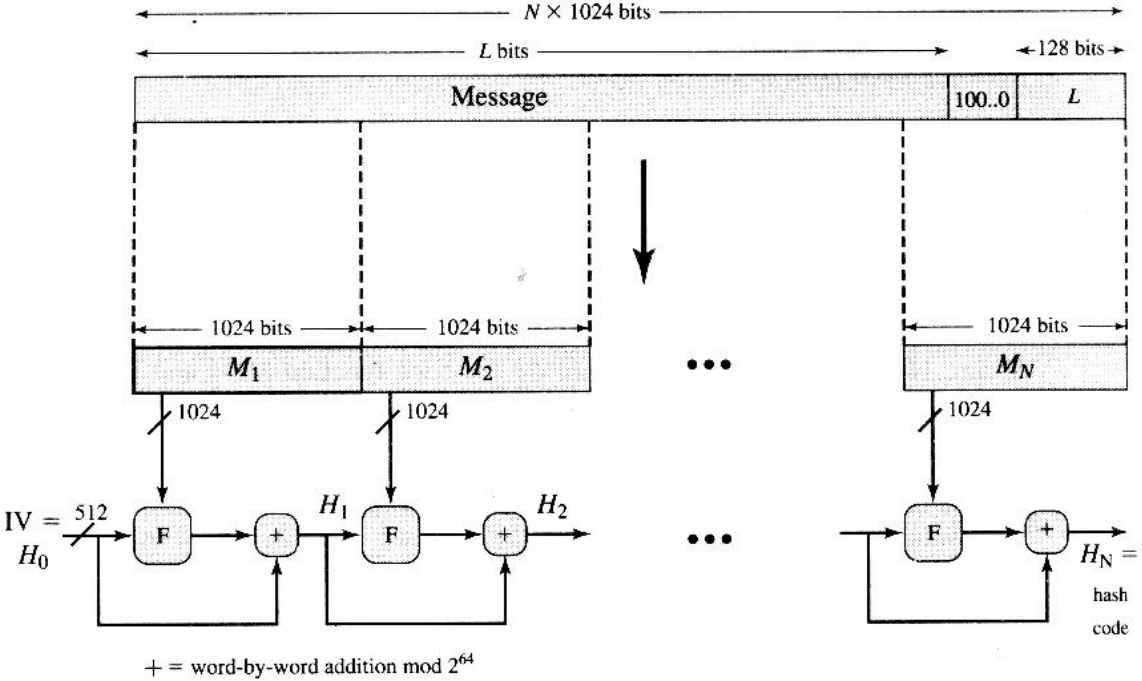
{This letter is / I am writing} to introduce {you to / to you} {Mr. / --} Alfred {P. / --}
Barton, the {new / newly appointed} {chief / senior} jewellery buyer for {our / the}
Northern {European / Europe} {area / division} He {will take / has taken} over {the / --}
responsibility for {all / the whole of} our interests in {watches and jewellery / jewellery and watches}
in the {area / region} Please {afford / give} him {every / all the} help he {may need / needs}
to {seek out / find} the most {modern / up to date} lines for the {top / high} end of the
market. He is {empowered / authorized} to receive on our behalf {samples / specimens} of the
{latest / newest} {watch and jewellery / jewellery and watch} products, {up / subject} to a {limit / maximum}
of ten thousand dollars. He will {carry / hold} a signed copy of this {letter / document}
as proof of identity. An order with his signature, which is {appended / attached}
{authorizes / allows} you to charge the cost to this company at the {above / head office}
address. We {fully / --} expect that our {level / volume} of orders will increase in
the {following / next} year and {trust / hope} that the new appointment will {be / prove}
{advantageous / an advantage} to both our companies.

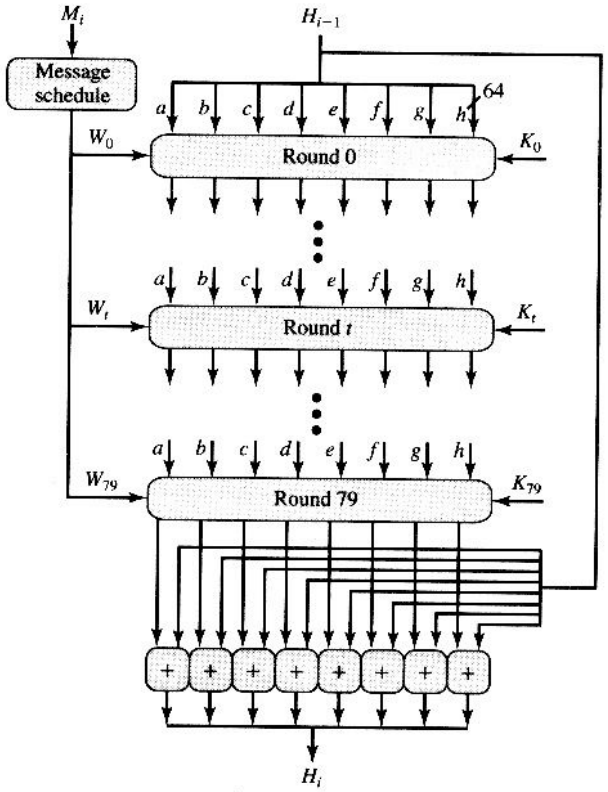
Secure Hash Algorithm

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

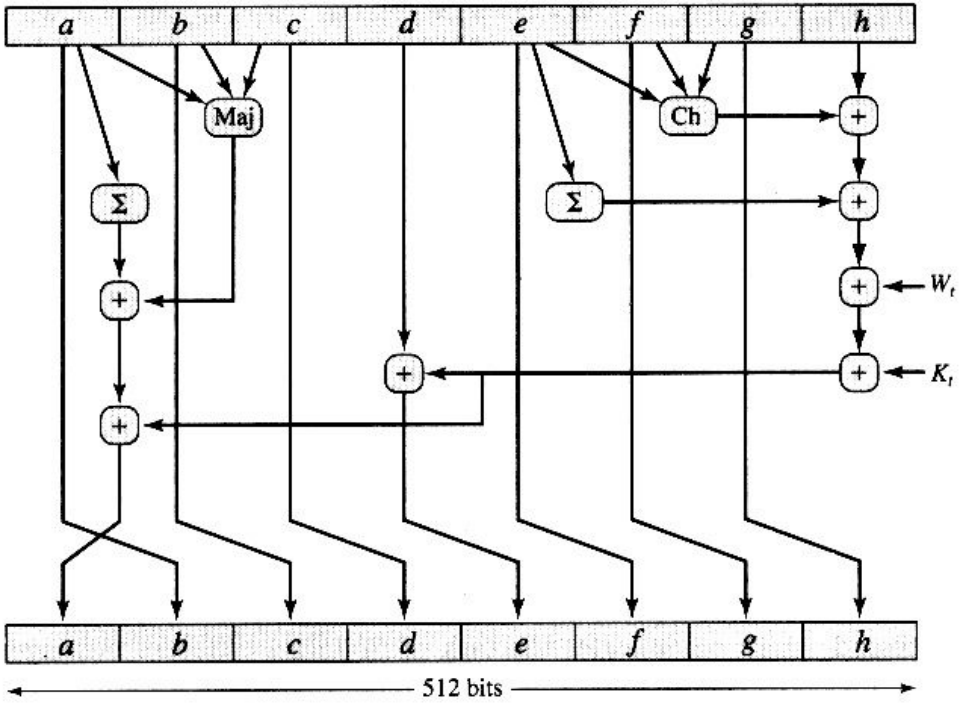
- NIST – FIPS 180/1993 – FIPS 180-1/1995 – FIPS 180-2/2002
- Baseado no MD4
- RFC 3174 – FIPS + Código C de referência
- SHA-1, SHA-256, SHA-384, SHA-512
- SHA-1 não recomendada pois tem colisões em 2^{69}

SHA-512


N × 1024 bits
L bits
128 bits
Message
100..0
L
1024 bits
1024 bits
1024 bits
M1
M2
MN
1024
1024
1024
IV =
H0
512
F
+
H1
F
+
H2
F
+
HN =
hash
code
+ = word-by-word addition mod 2^64


Mi
Hi−1
Message
schedule
64
a b c d e f g h
W0
Round 0
K0
Wt
Round t
Kt
W79
Round 79
K79
+
Hi

SHA-512



$Ch = (\wedge)\otimes(\neg\wedge)$

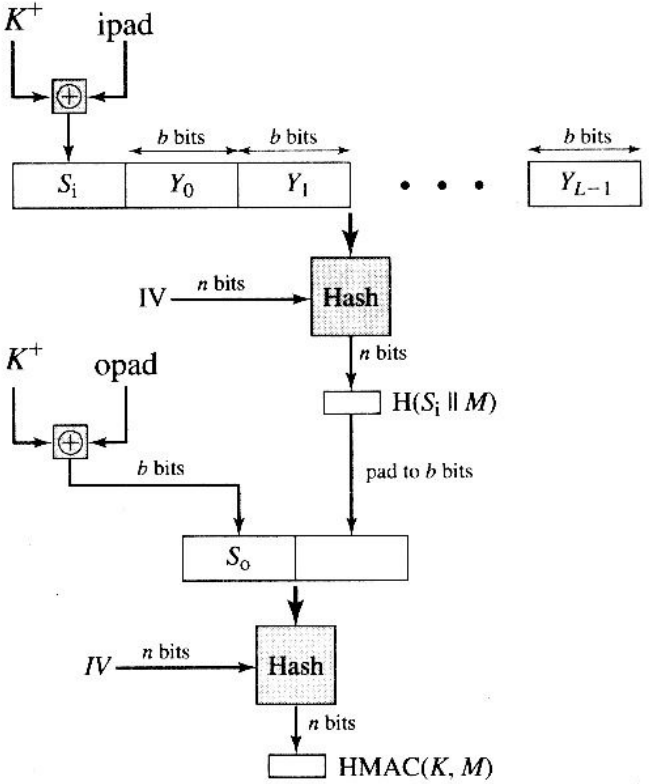
$Maj = (\wedge)\otimes(\wedge)\otimes(\wedge)$

$\sum a = R^{28}\otimes R^{34}\otimes R^{39}$

$\sum e = R^{14}\otimes R^{18}\otimes R^{41}$

HMAC

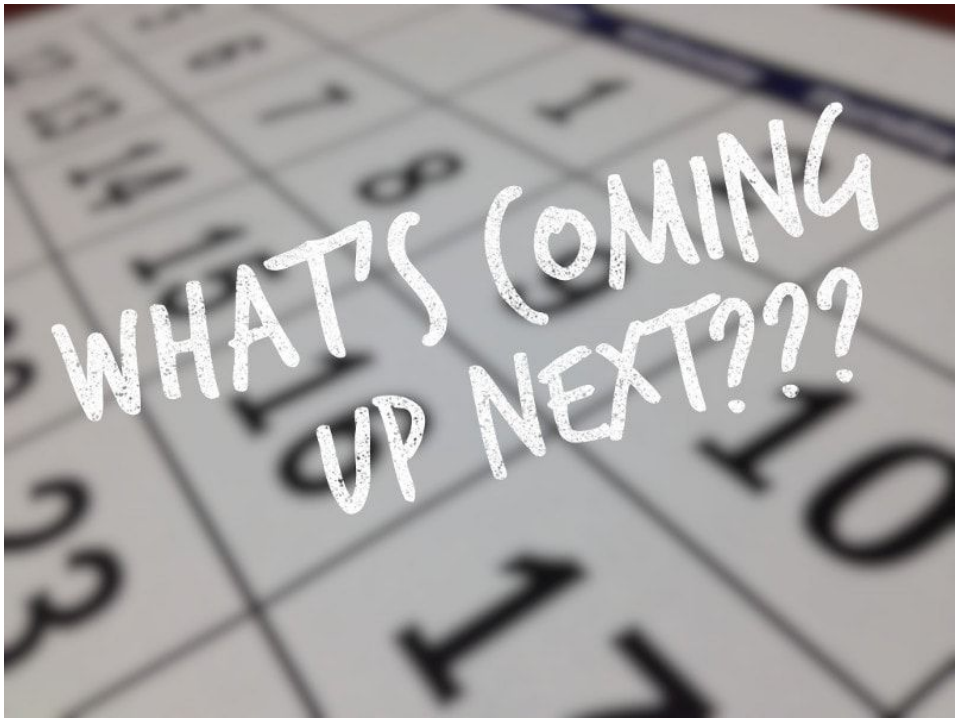
- MAC baseado em função HASH
- Objetivos:
  - Mais rápido que cifragem
  - Funções HASH amplamente disponíveis
- RFC 2104 /FIPS 198 → como adicionar um chave a um HASH
- Usado em SSL e IPSEC
- Objetivos:
  - Usar funções HASH sem modificação
  - Permitir trocar a função HASH
  - Preservar a performance do HASH
  - Usar chave de maneira simples
  - Ter toda análise criptográfica baseada no função HASH



Próximas Aulas

- Prática:
  - Trabalho Individual II

- Teórica:
  - Assinatura Digital e Protocolos Criptográficos



Perguntas?

jean.martina@ufsc.br