



Curso de Extensão  
Tecnologias Microsoft



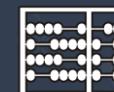
INF-0998  
Programação segura  
(Segurança de software)  
Aula 2

Prof. Dr. Alexandre Braga, CISSP, CSSLP, PMP

[alexbraga@ic.unicamp.br](mailto:alexbraga@ic.unicamp.br) / [ambraga@cpqd.com.br](mailto:ambraga@cpqd.com.br) / [braga.alexandre.m@gmail.com](mailto:braga.alexandre.m@gmail.com)

[br.linkedin.com/in/alexmbraga](https://br.linkedin.com/in/alexmbraga)

19 de Novembro de 2022



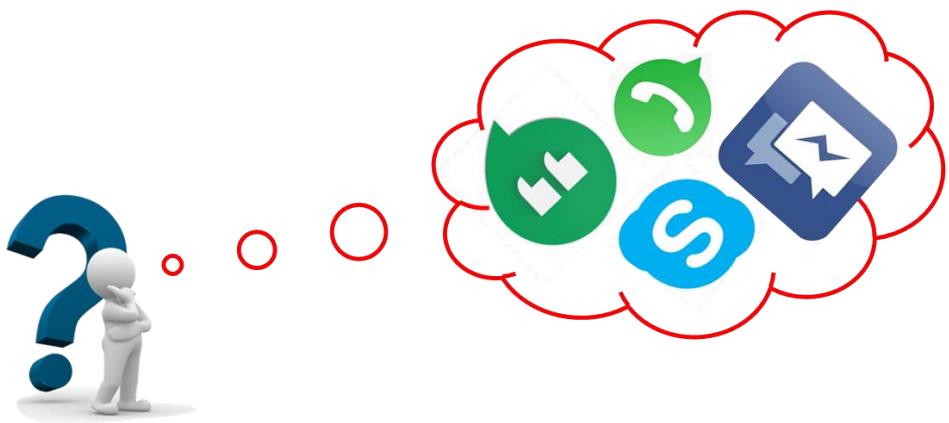
INSTITUTO DE  
COMPUTAÇÃO

# O que é criptografia, conceitos e objetivos

# Você sabia?



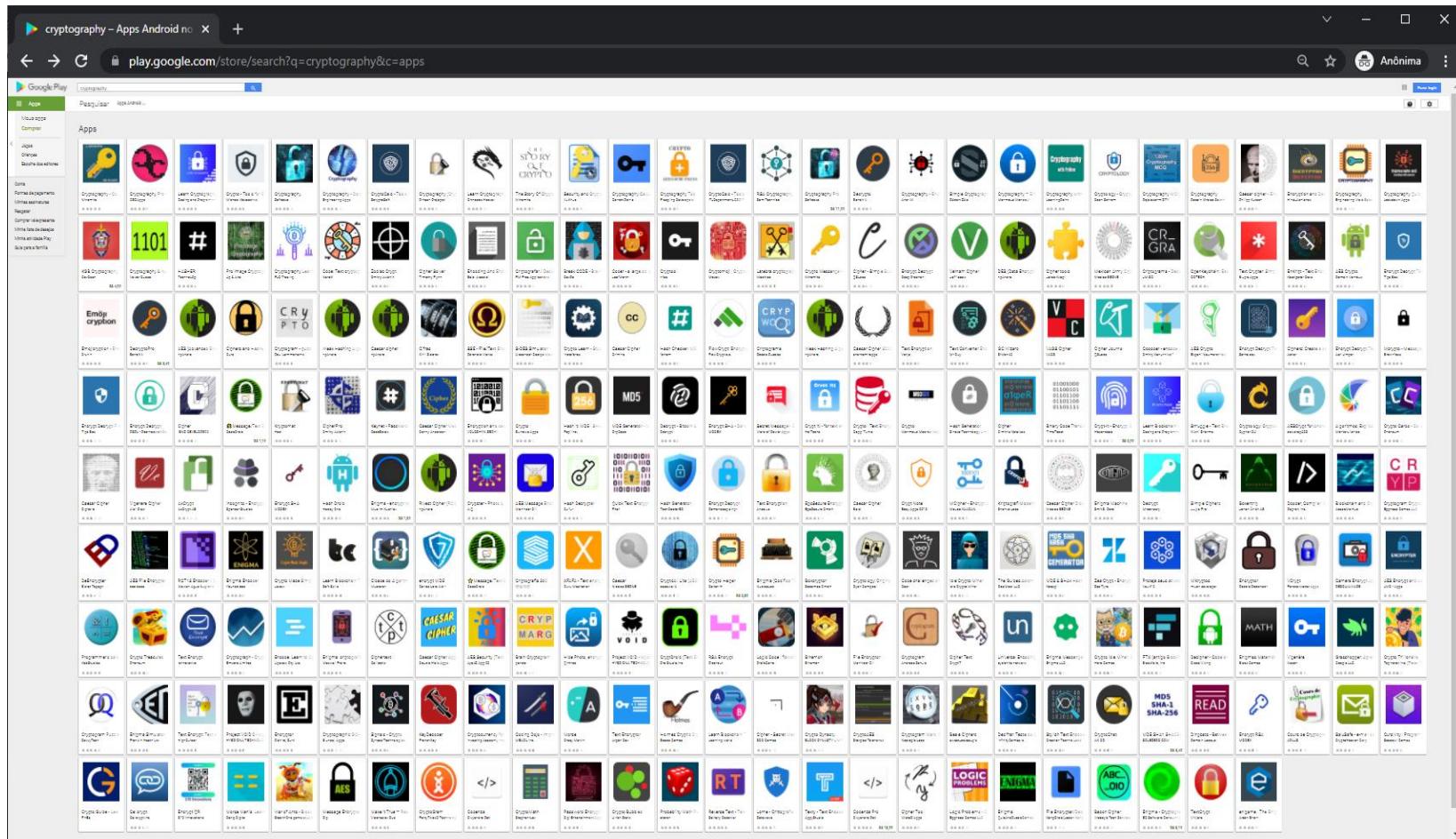
- Os softwares criptográficos mais populares são os aplicativos com **criptografia na lógica de negócios**
- O uso correto da criptografia na aplicação
  - Não é problema de infraestrutura!
  - Não é resolvido apenas com TLS!
  - Não é só https!





# Você tem, usa ou faz!

- Software criptográfico é cada vez mais comum



Fonte: <https://play.google.com/store/search?q=cryptography&c=apps>

# Contexto: por que isto é importante?



- O uso incorreto é a causa mais comum de problemas de segurança criptográfica
- Diversos erros de configuração evitáveis se tornaram recorrentes, resultando em vulnerabilidades exploráveis em ataques reais aos sistemas criptográficos
- The Heartbleed Bug
  - <http://heartbleed.com>
- Apple's goto fail in SSL/TLS bug
  - <http://support.apple.com/kb/HT6147>
- The Logjam Attack & Weak Diffie-Hellman
  - <https://weakdh.org>
- ECDSA Attack no bitcoin
  - <http://blog.bettercrypto.com/?p=916>
- WhatsApp Encryption (2013)
  - <https://blog.thijsalkema.de/blog/2013/0/08/piercing-through-whatsapp-s-encryption>
- The DROWN attack
  - <https://drownattack.com>
- The POODLE attack
  - <https://www.openssl.org/~bodo/ssl-poodle.pdf>
- The FREAK attack
  - [{mitls.org/pages/attacks}](https://mitls.org/pages/attacks)
- The CRIME attack
  - <http://netifera.com>
- The Lucky Thirteen
  - <http://www.isg.rhul.ac.uk/tls/Lucky13.html>
- BEAST, PlayStation 3, etc. etc. etc.

# Objetivos e funções da criptografia



- A criptografia é a única tecnologia capaz de garantir o sigilo, integridade e autenticidade da informação digital em trânsito pelas redes abertas, públicas, sem fio ...
- A Criptografia é a ciência que se dedica ao estudo e ao desenvolvimento das técnicas (matemáticas) utilizadas para tornar uma mensagem secreta
- A criptografia na antiguidade clássica
  - Realizada manualmente
  - Historicamente associada à encriptação para confidencialidade
- A criptografia moderna
  - A partir da segunda guerra mundial
  - Oferece também serviços para autenticidade, integridade e irrefutabilidade
- A confidencialidade é obtida com os serviços de encriptação e decriptação
- Autenticidade e irrefutabilidade são obtidas com os serviços de geração de assinaturas digitais e verificação destas assinaturas
- Integridade é obtida com os serviços de geração de resumos criptográficos (hash seguros).
- Serviços criptográficos são quase sempre combinados. Exemplo:
  - Mensagem eletrônica pode ser encriptada e assinada digitalmente
  - Arquivo encriptado pode estar acompanhado do hash do criptograma

# O que é criptografia



- A criptografia pode ser usada de muitas maneiras
  - Ela é primeira linha de defesa contra ameaças conhecidas, tais como o *snooping* (“grampo” eletrônico) e *spoofing* (personificação falsa) em ambientes computacionais públicos
- Origem da palavra criptografia
  - Criptografia = kryptós + gráphein
  - Escrita em segredo
  - Kryptós, do grego “escondido” ou “secreto”
  - Gráphein que significa “escrita”
- Conceito: criptografia
  - A **ciência/arte** da transformação da informação em código ilegível, para proteger seu conteúdo, tornando a informação secreta
    - Usa seus próprios métodos e técnicas

# Exemplo de criptografia clássica: Vigenère



Cada letra da chave define um deslocamento diferente.

Mensagem: P L **A** I N T E X T  
Chave: f r **e** e f r e e f  
Criptograma: U C **E** M S L I B Y

Alfabeto do texto claro

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
a	<b>A</b>	B	C	D	<b>E</b>	F	G	H	I	J	K	L	M	N	O	P	<b>Q</b>	R	S	T	U	V	W	X	Y	Z
b	B	<b>C</b>	D	E	F	G	H	I	J	K	L	M	N	O	P	<b>Q</b>	R	S	T	U	V	W	X	Y	Z	
c	C	D	<b>E</b>	F	G	H	I	J	K	L	M	N	O	P	<b>Q</b>	R	S	T	U	V	W	X	Y	Z		
d	D	E	F	<b>G</b>	H	I	J	K	L	M	N	O	P	<b>Q</b>	R	S	T	U	V	W	X	Y	Z	A	B	
e	<b>E</b>	F	G	H	I	J	K	L	M	N	O	P	<b>Q</b>	R	S	T	<b>U</b>	V	W	X	Y	Z	A	B	C	
f	F	G	H	I	J	K	L	M	N	O	P	<b>Q</b>	R	S	T	U	V	W	W	X	Y	Z	A	B	C	
g	G	H	I	J	K	L	M	N	O	P	<b>Q</b>	R	S	T	U	V	W	W	X	Y	Z	A	B	C	D	
h	H	I	J	K	L	M	N	O	P	<b>Q</b>	R	S	T	U	V	W	W	X	Y	Z	A	B	C	D		
i	I	J	K	L	M	N	O	P	<b>Q</b>	R	S	T	U	V	W	W	X	Y	Z	A	B	C	D	E		
j	J	K	L	M	N	O	P	<b>Q</b>	R	S	T	U	V	W	W	X	Y	Z	A	B	C	D	E	F		
k	K	L	M	N	O	P	<b>Q</b>	R	S	T	U	V	W	W	X	Y	Z	A	B	C	D	E	F	G	H	
l	L	M	N	O	P	<b>Q</b>	R	S	T	U	V	W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	
m	M	N	O	P	<b>Q</b>	R	S	T	U	V	W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	
n	N	O	P	<b>Q</b>	R	S	T	U	V	W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	
o	O	P	<b>Q</b>	R	S	T	U	V	W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	
p	P	<b>Q</b>	R	S	T	U	V	W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	
q	Q	R	S	T	U	V	W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	
r	R	S	T	U	V	W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
s	S	T	U	V	W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	
t	T	U	V	W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	
u	U	V	W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	
v	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	
w	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	
x	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	
y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	
z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	

Alfabeto da chave

# Objetivos e funções da criptografia



## Criptografia para confidencialidade

- Manter a informação secreta
- A informação só é revelada pelo uso da chave

### Exemplos

- Enviar e-mail encriptado
- Armazenar arquivo encriptado

## Criptografia para autenticidade

- Validar a identidade de uma entidade
- Validar a origem/destino da informação

### Exemplo

- Verificação de autoria de documentos
- Validação de remetente/destinatário da mensagem eletrônica

## Criptografia para Integridade

- Garantir que a informação não foi maliciosamente modificada desde sua criação
- Garantir que os dados não foram adulterados desde a última modificação legítima

### Exemplo

- Valores *hash* associados a arquivos para download

## Criptografia para Irrefutabilidade

- Também chamada de não-repudião
- Incapacidade de negar a autoria de uma mensagem
- O autor é incapaz de negar que produziu a mensagem

### Exemplos

- Assinaturas autenticadas não podem ser negadas em documentos assinados

# O que não é criptografia



## Esteganografia não é criptografia

- Derivada do grego para “escrita escondida”
- Ocultação de mensagens dentro de outras mensagens
  - Camufla ou oculta a presença da mensagem
  - Uma mensagem serve de meio de transporte para outra mensagem

## Codificação base64 não é criptografia

- “Teste de codificação base 64”
- Codificação
  - VGVzdGUgZGUgY29kaWZpY2HDp8OjbyBiYXNlIDY0
- <https://www.base64encode.org>

Criptografia	Esteganografia	Codificação (Base64)
Impede a leitura de mensagem	Oculta a existência da mensagem	Muda a representação da informação na mensagem
Não oculta a existência da mensagem	A mensagem poderá ser lida se for descoberta	Não impede a leitura e nem oculta a existência da mensagem

# O feijão com arroz ...



Evitar a criptografia fraca ou obsoleta  
DES, 3DES, RC4, MD5, MD2, SHA-1



[https://commons.wikimedia.org/wiki/File:Rice\\_and\\_beans,\\_Hot\\_in\\_Itatiaia.jpeg](https://commons.wikimedia.org/wiki/File:Rice_and_beans,_Hot_in_Itatiaia.jpeg)

Evitar tamanhos de chave inseguros

< 2048 para RSA

< 2048 para DH

< 128 para encriptação simétrica

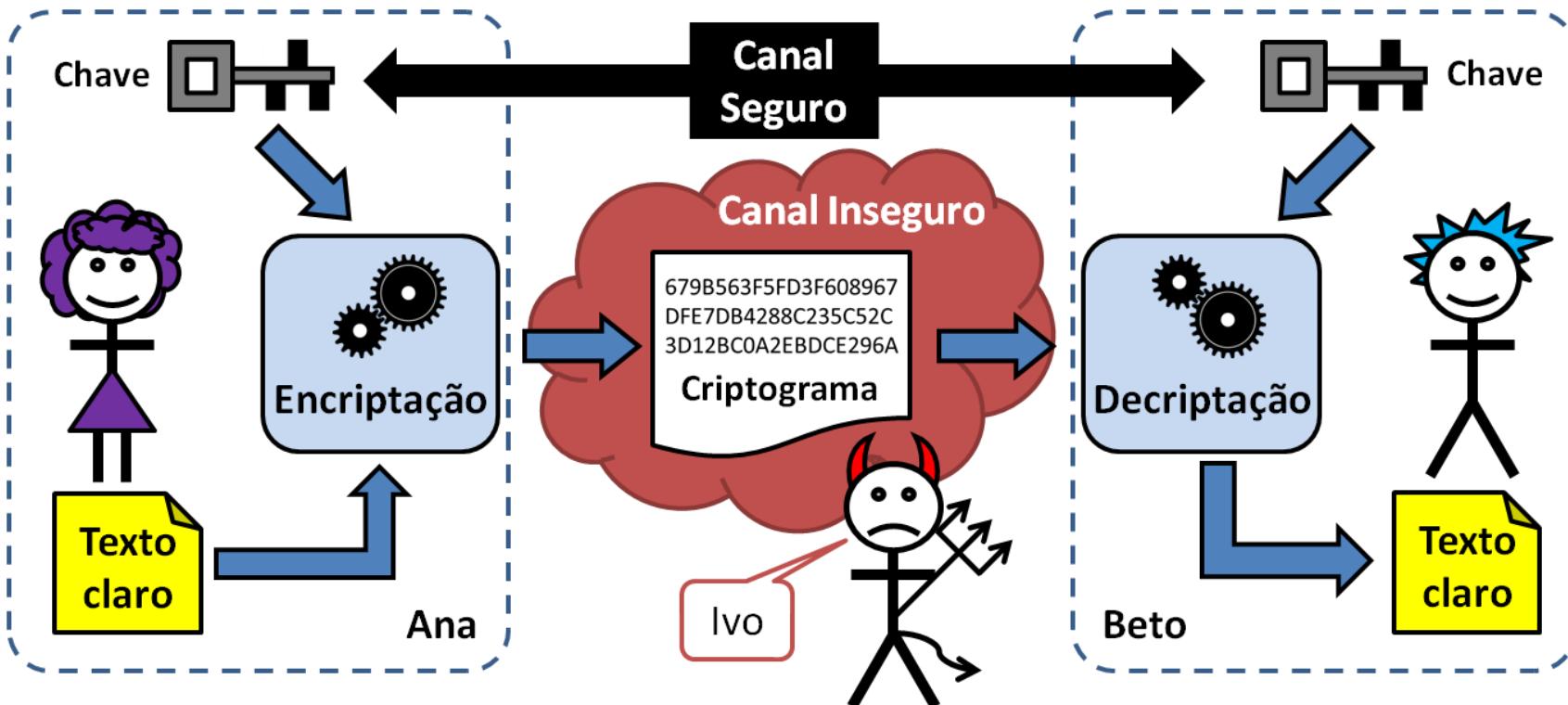
< 256 para *hashes (output)*

< 256 para ECC

AES	RSA	DH (k)	DH (p)	ECC	hash
80	1024	160	1024	160–163	SHA1 (160)
112	2048	224	2048	224–233	SHA-224/512 SHA3-224
128	3072	256	3072	256–283	SHA-256/512 SHA3-256
192	7680	384	7680	384–409	SHA-384 SHA3-384
256	15360	512	15360	512–571	SHA-512 SHA3-512

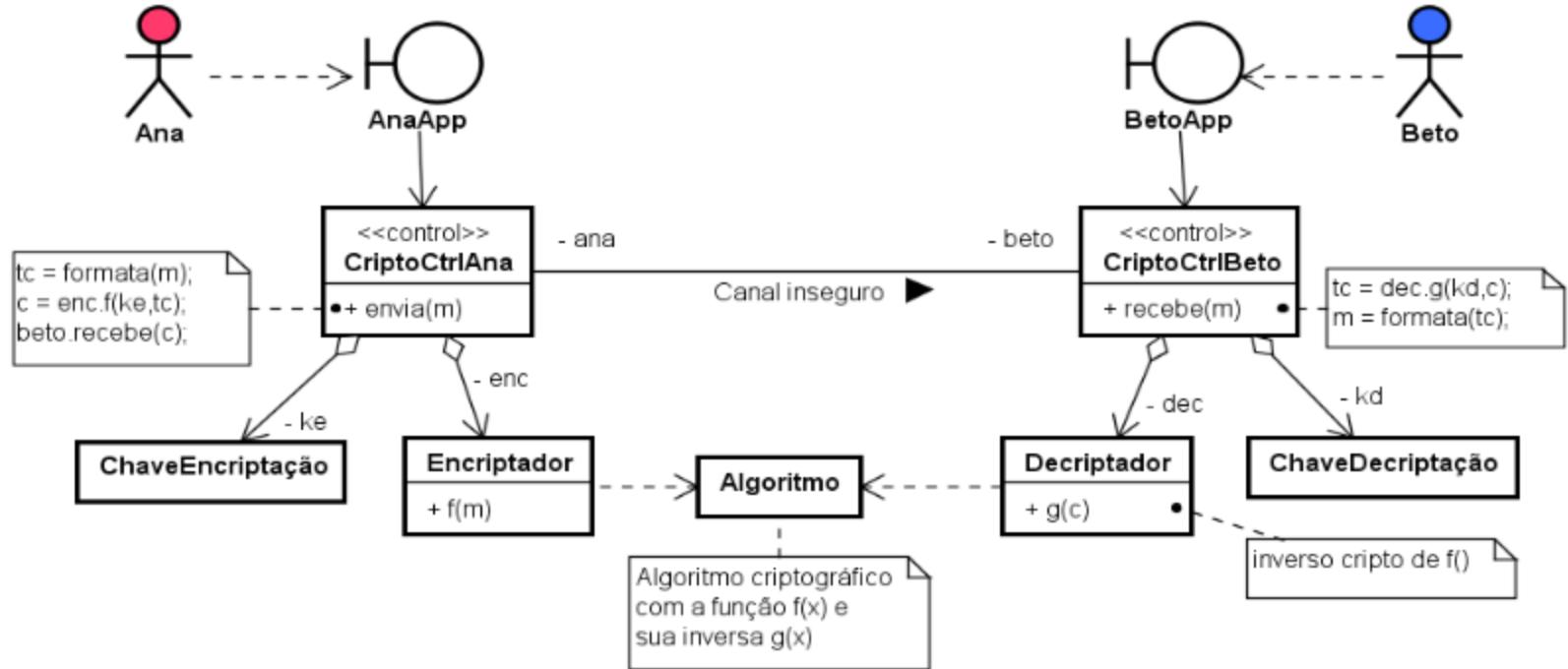
# Casos de uso, Tecnologias e Padrões

# Sistema criptográfico



- Três personagens ilustram a figura: Ana, a remetente das mensagens; Beto, o destinatário das mensagens; e Ivo, o adversário com desejo de conhecer os segredos de Ana e Beto. As mensagens passam por um canal de comunicação inseguro e controlado por Ivo.
- O algoritmo criptográfico é usado para transformar o texto em claro (legível por qualquer um) em texto encriptado (o criptograma legível apenas por Ana e Beto) e vice-versa.

# O Padrão de projeto de software criptográfico



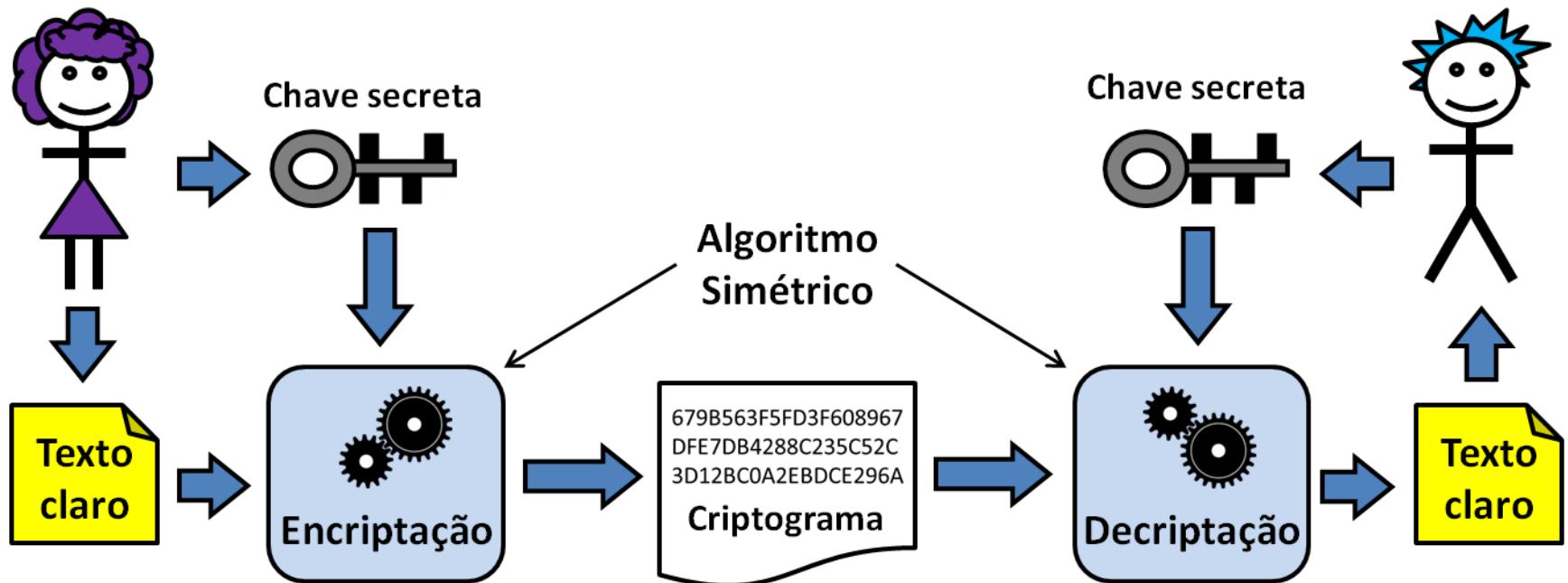
- A estrutura de um sistema criptográfico segue um padrão recorrente de arquitetura que foi documentado pelos autores e ficou conhecido como o padrão de projeto de software criptográfico.
- A Figura mostra o diagrama de classes, em UML, de um sistema criptográfico simétrico para sigilo, em que Ana encripta e Beto descripta.
- O leitor deve ser capaz de abstrair a arquitetura de software da Figura e visualizar estruturas semelhantes nos trechos de programas criptográficos mostrados a seguir.

# Nomenclatura: palavras de uso específico



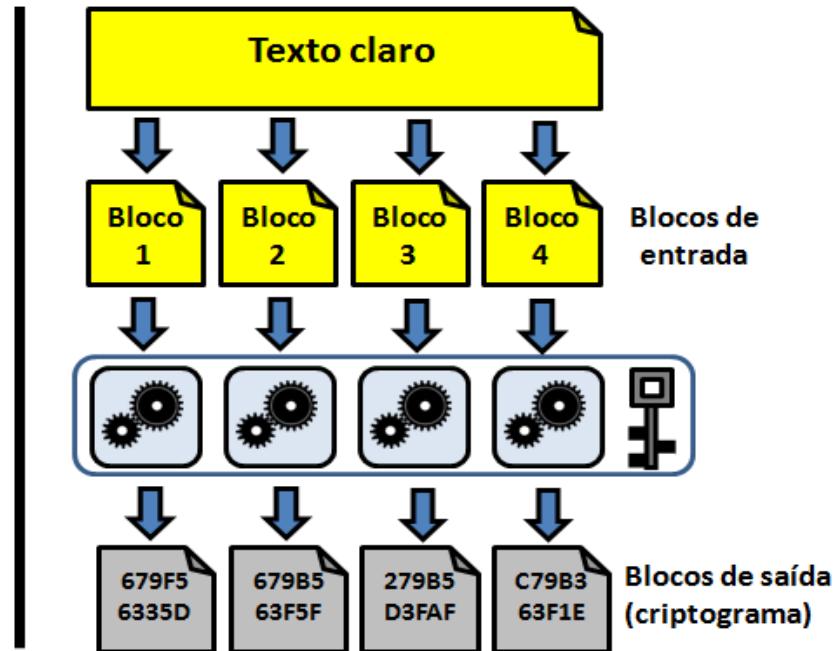
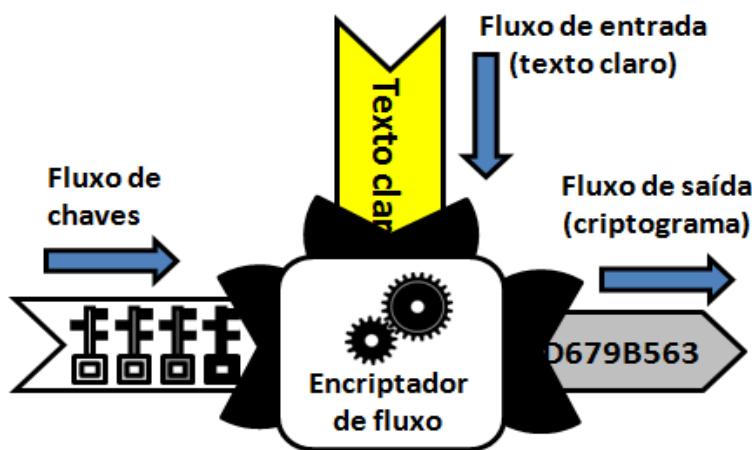
- **Texto claro** = texto original, legível, em claro, não encriptado
- **Criptograma** = Texto encriptado, texto ilegível, não comprehensível
- **Encriptar** = Transformar texto claro em criptograma por meio de operação criptográfica e uma chave
- **Decriptar** = Transformar criptograma em texto claro por meio da operação inversa e a chave
- **Chave criptográfica** = Informação sensível que dirige as operações criptográficas e controla a segurança da transformação
- **Algoritmo criptográfico** = Sequência de passos da transformação criptográfica
- **Esquema criptográfico** = Combinação de algoritmos criptográficos e outros componentes
- **Sistema criptográfico**
- Sistema formato por todos os elementos acima
- Com o ciclo completo de operações criptográficas diretas e inversas
- Por exemplo, operações de encriptação e de decriptação

# Sistema criptográfico simétrico



- A Figura ilustra os passos da encriptação com chave secreta:
  - Ana configura o algoritmo para o modo de encriptação com a chave secreta;
  - Ana alimenta o algoritmo com a mensagem original, o texto claro;
  - Ana encripta a mensagem e obtém o criptograma (mensagem encriptada)
- A Figura também ilustra os passos da decriptação com chave secreta:
  - Beto configura o algoritmo para o modo de decriptação com a chave secreta;
  - Beto alimenta o algoritmo com a mensagem encriptada (criptograma);
  - Beto decripta a mensagem encriptada e obtém o texto claro original.

# Encriptadores de fluxo e de bloco



- Nos encriptadores de bloco, o texto claro é quebrado em blocos de bits de tamanho fixo. O encriptador trabalha sobre blocos e produz saídas em blocos também. O tamanho da chave criptográfica é geralmente um múltiplo do tamanho do bloco.
- Os encriptadores de fluxo trabalham sobre sequências (de bits). A sequência ou fluxo de entrada é transformado continuamente na sequência ou fluxo de saída, bit a bit. É importante que a chave criptográfica seja uma sequência de bits pelo menos do mesmo tamanho do fluxo de entrada.

# Alguns algoritmos criptográficos simétricos



Os algoritmos de encriptação seguros:

- AES (*Advanced Encryption Standard*) é um padrão de encriptação do governo norte-americano e é a principal opção de escolha para um algoritmo de encriptação de blocos.
- Camelia é um algoritmo de encriptação de blocos utilizado nas novas versões do TLS.

Os algoritmos de encriptação obsoletos:

- 3DES, TripleDES ou DES-EDE é obtido pela realização de três encriptações sucessivas com o algoritmo DES.
- Kasumi é um algoritmo de encriptação de blocos proposto pelo 3GPP
- Blowfish é um algoritmo de encriptação de bloco proposto pelo criptólogo Bruce Schneier.
- O DES (*Data Encryption Standard*) não deve mais ser utilizado, pois é suscetível aos ataques de força bruta.

# Fluxo: One-Time Pad (cifra de Vernam) com XOR



- Uma mensagem M binária é transformada por uma chave K binária.
  - K e M tem o mesmo comprimento.
  - A transformação criptográfica é a operação binária XOR (ou exclusivo)

$$c_i = m_i \oplus k_i, \quad 1 \leq i \leq t.$$

- Se K é aleatória e descartável (só é usada uma única vez) esta cifra pode ser chamada de *one-time-pad*.

- É demonstrada ser inquebrável na teoria.
- Além disso, para chaves muito grandes, a quantidade de 0s e 1s deve ser aproximadamente a mesma.
- A chave K precisa ser trocada de modo seguro.
- Tabela verdade do XOR
  - 0011
  - 1010
  - ----- (XOR)
  - 1001

# Exemplo de Cifra de fluxo OTP com XOR



```
private static byte[] crypt(byte M[], byte K[]) {  
    byte C[] = null;  
    C = new byte[M.length];  
    for (int i = 0; i < M.length; i++) {  
        C[i] = (byte) (M[i] ^ K[i % K.length]);  
    }  
    return C;  
}
```

- M = "alex" = 61 6c 65 78
- K = "abcd" = 61 63 64 65
- M: 01100001 01101100 01100101 01111000
- XOR K: 01100001 01100010 01100011 01100100
- = C: 00000000 00001110 00000110 00011100
- XOR K: 01100001 01100010 01100011 01100100
- = M: 01100001 01101100 01100101 01111000

1	0	0	1	1
1	1	0	0	1
$1 \oplus 1$	$0 \oplus 1$	$0 \oplus 0$	$1 \oplus 0$	$1 \oplus 1$
0	1	0	1	0

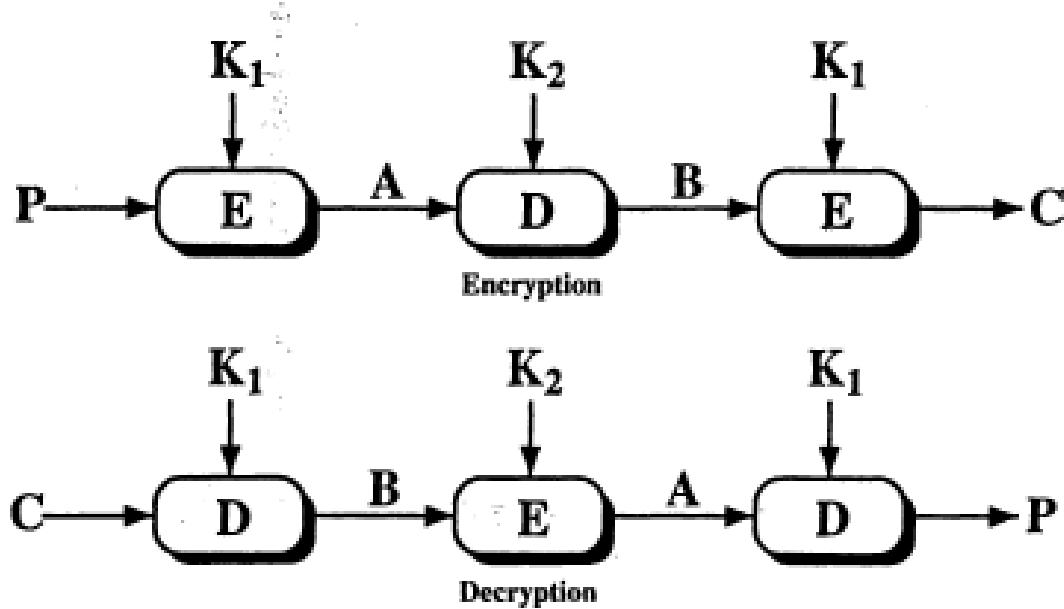
Isto está  
errado!  
Por que?

# DES – Data Encryption Standard



- D.E.S (Data Encryption Standard), foi o método oficial do governo dos E.U.A. para ciframento de informação não “sensível”, até muito pouco tempo.
- Ainda é muito usado em toda a Internet, em sua forma mais fortalecida, o 3DES.
- Critérios de projeto do D.E.S. não foram totalmente revelados pelos desenvolvedores (originalmente um projeto da IBM).
- Isso causou, por anos, desconfiança de que o algoritmo contivesse backdoors que permitiriam acesso a mensagens cifradas dos usuários.
- 56 bits de chave
- mais 8 bits paridade !?
- Processa blocos de texto de 64 bits cada vez, sob a ação de uma chave de 56 bits, produzindo 64 bits de texto cifrado.
- É composto de 16 rodadas de produtos de transposições e substituições, cada uma usando uma porção diferente da chave.
- O efeito avalanche garante que qualquer mudança no texto claro ou chave cause mudança de metade do texto cifrado.

# Triple DES, 3DES, DES-EDE



O DES não é um grupo algébrico. Então NÃO EXISTE uma  $k_3$  tal que:

$$E_{K_2}[E_{K_1}[P]] = E_{K_3}[P]$$

Comumente, 3 chaves são usadas, de modo que:

$$C = DES_{K_1}(DES_{K_2}^{-1}(DES_{K_1}(M)));$$

$$DES_{K_1}^{-1}(DES_{K_2}(DES_{K_1}^{-1}(C))) = M.$$

- **$k_1 = k_2 = k_3$ :** equivalente ao DES original. (compatibilidade com DES legado)
- **$k_1 = k_3 \neq k_2$ :** duas chaves diferentes (o tamanho da chave final é 112 bits)
- **$k_1 \neq k_2 \neq k_3$ :** 3 chaves independentes (o tamanho da chave final é 168 bits);

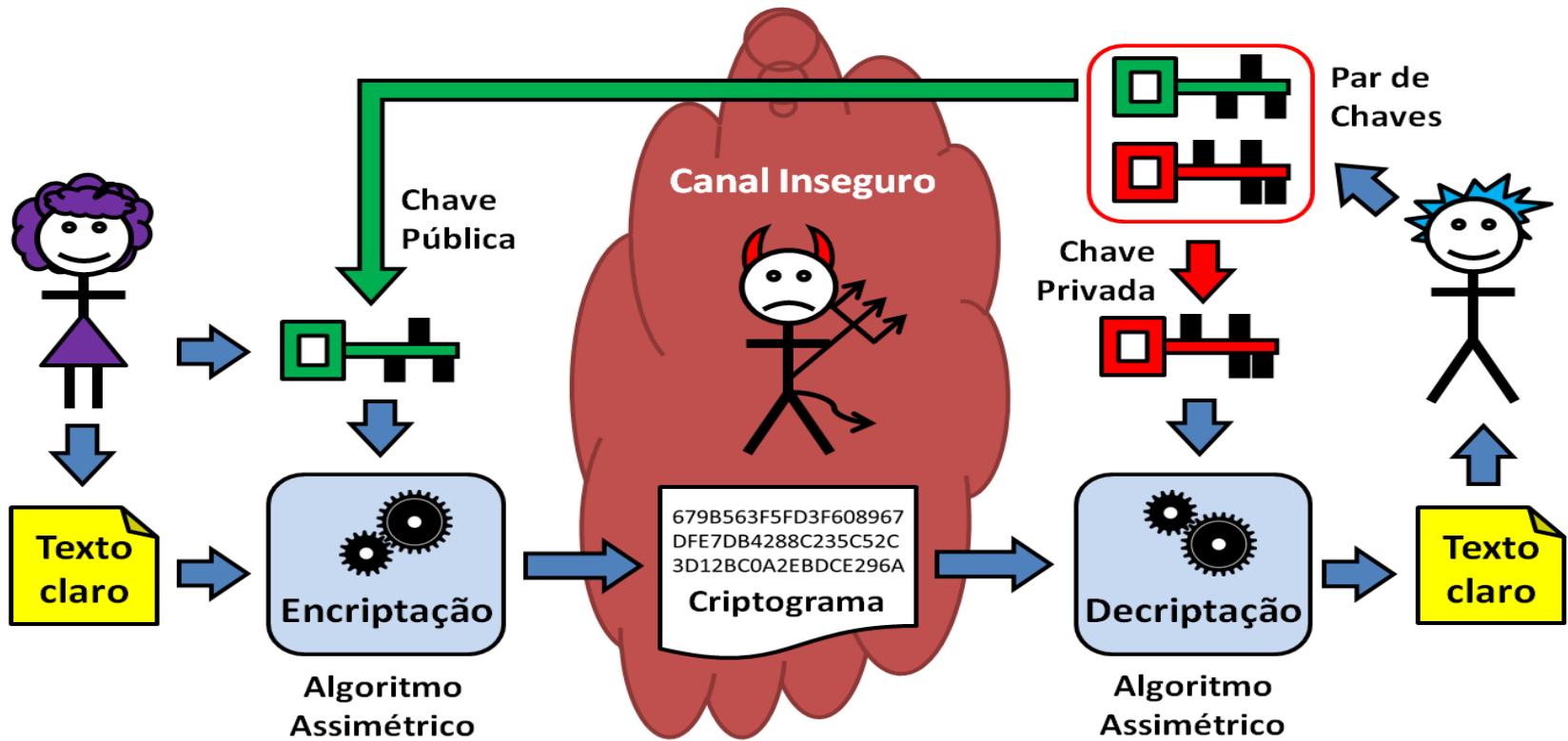
# AES – Advanced Encryption Standard



- O DES já não resistia a ataques de força bruta. Foi preciso substituí-lo. O padrão avançado de criptografia de dados do governo americano foi definido em um concurso internacional, que foi vencido pelo algoritmo Rijndael (de dois criptólogos Belgas).
- O Rijndael poderia processar outros tamanhos de bloco, mas o tamanho de 128 bits foi fixado para o AES. O número de rounds depende do tamanho da chave. Vide tabela.

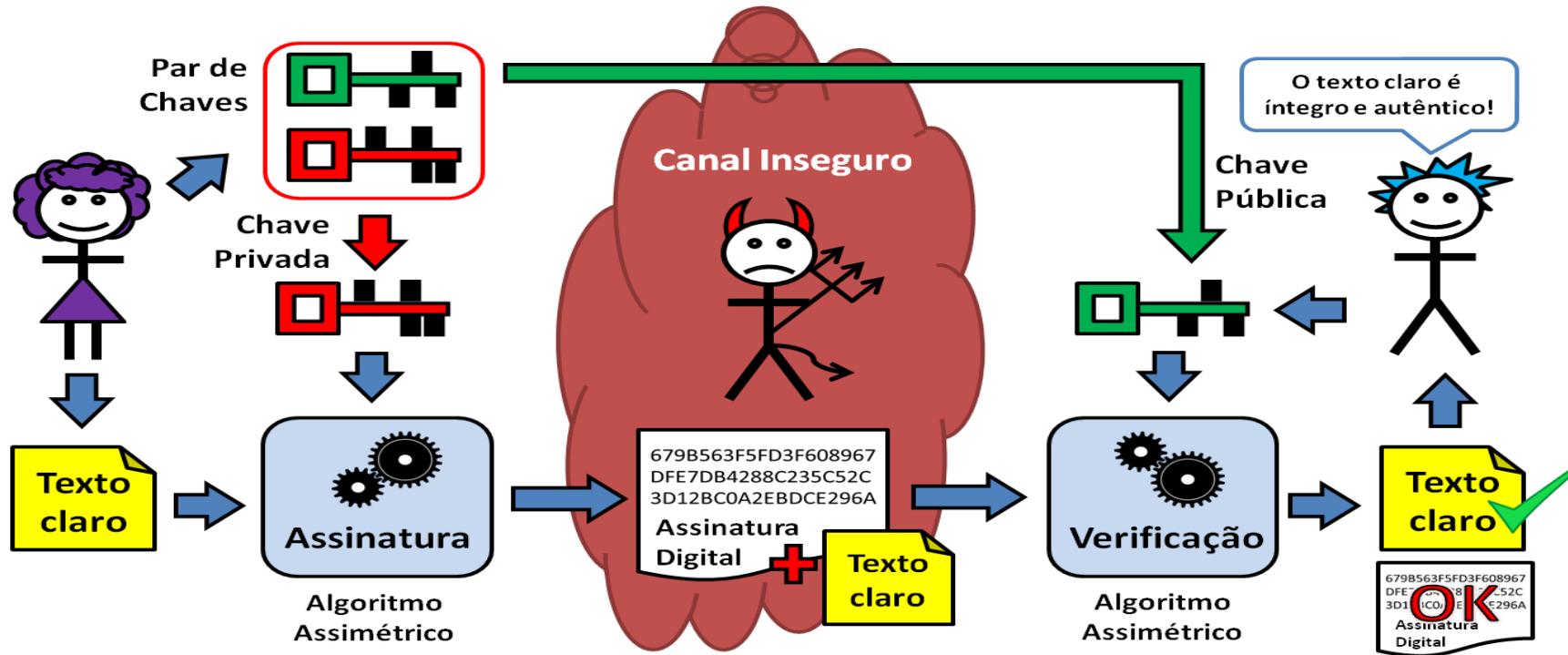
Chave bits (words)	Bloco	ROUNDS
128 (4 w)	128 bits (4 w)	10
192 bits (6 w)	128 bits (4 w)	12
256 bits (8 w)	128 bits (4 w)	14

# Sistema criptográfico assimétrico para sigilo



- Mais uma vez, Ana, Beto e Ivo são os personagens. As mensagens de Ana para Beto são transmitidas por um canal inseguro, controlado por Ivo. Beto possui um par de chaves, uma chave pública e outra privada. Ana conhece a chave pública de Beto, mas somente o dono do par de chaves (Beto) conhece a chave privada (não há segredo compartilhado).
- Observa-se que Ana envia uma mensagem privada para Beto. Para fazer isso, Ana encripta a mensagem usando a chave pública de Beto. Ana conhece a chave pública de Beto por que ela foi divulgada por Beto. Porém, o criptograma só pode ser decrito pela chave privada de Beto; nem Ana pode fazê-lo.

# Sistema cripto. assimétrico para autenticidade



- Assinatura digital é o resultado de uma certa operação criptográfica com a chave privada sobre o texto claro. O dono da chave privada pode gerar mensagens assinadas, que podem ser verificadas por qualquer um que conheça a chave pública correspondente, portanto, sendo capaz de verificar a autenticidade da assinatura digital. Nem sempre a operação de assinatura é uma encriptação e a sua verificação é uma decriptação.
- Ana usa sua chave privada para assinar digitalmente uma mensagem para Beto. O texto claro e a assinatura digital são enviados por um canal inseguro e podem ser lidos por todos, por isso a mensagem não é secreta. Qualquer um que conheça a chave pública de Ana (todo mundo, inclusive Beto), pode verificar a assinatura digital. Ivo pode ler a mensagem, mas não pode falsificá-la, pois não conhece a chave privada de Ana.

# Funções de resumo (hash) criptográfico



Fonte da imagem: [https://images-submarino.b2w.io/produtos/01/00/images/41915/2/41915277\\_1GG.jpg](https://images-submarino.b2w.io/produtos/01/00/images/41915/2/41915277_1GG.jpg)

Seja  $H(x) = h$ , onde  $x$  é o texto claro,  $H()$  é a função de resumo e  $h$  é o valor hash. As seguintes propriedades são verdadeiras:

- $H(x)$  pode ser aplicada a um  $x$  de qualquer tamanho;
- $H(x)$  produz uma saída  $h$  de tamanho fixo;
- É eficiente calcular  $H(x)$  para qualquer  $x$ ;
- Para um  $h$  qualquer, é impossível calcular um  $x$ , tal que  $H(x) = h$ ;
- Para um  $x$ , é impossível achar  $y$ , tal que  $H(x) = H(y)$ ;
- É impossível achar um par qualquer  $(x,y)$ , tal que  $H(x) = H(y)$ .

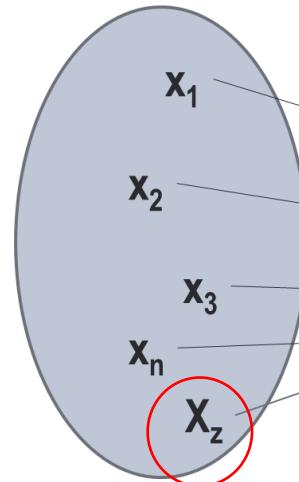
- Normalmente, não é o texto claro inteiro que é assinado digitalmente, mas sim um resumo dele, que o identifique unicamente. Este identificador único é calculado por rotinas matemáticas chamadas de funções de resumo (ou *hash*) criptográfico.
- Estas funções de *hash* geram uma sequência de bits, o valor do *hash*, que é único para o documento de entrada da função. O *hash* é muito menor que o documento original e geralmente tem um tamanho fixo de dezenas (algumas centenas) de bits. A função de *hash* é unidirecional porque não é reversível, isto é, não é possível recuperar o documento original a partir da sequência binária do *hash*. Além disso, idealmente, não existem dois documentos que geram o mesmo valor de *hash*.

# Colisões em hash



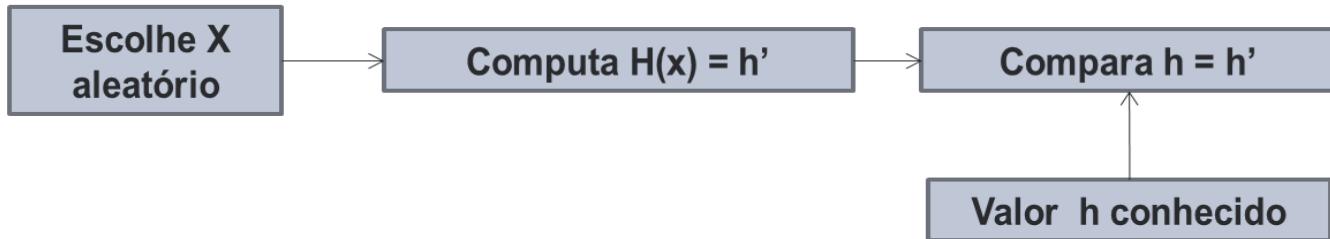
Nas figuras,  $H(x) = h$  é a função de hash com entrada  $x$  e saída  $h$

Espaço combinatório de todas as entradas  $X$  possíveis

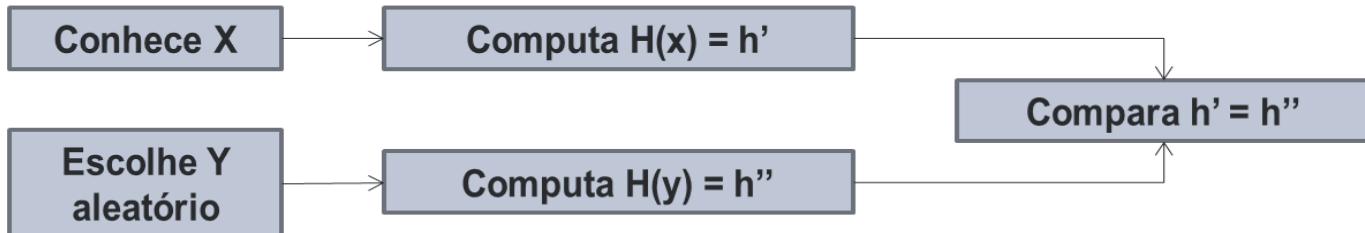


Espaço combinatório de todas as saídas  $h$  possíveis

## Primeira pré-imagem



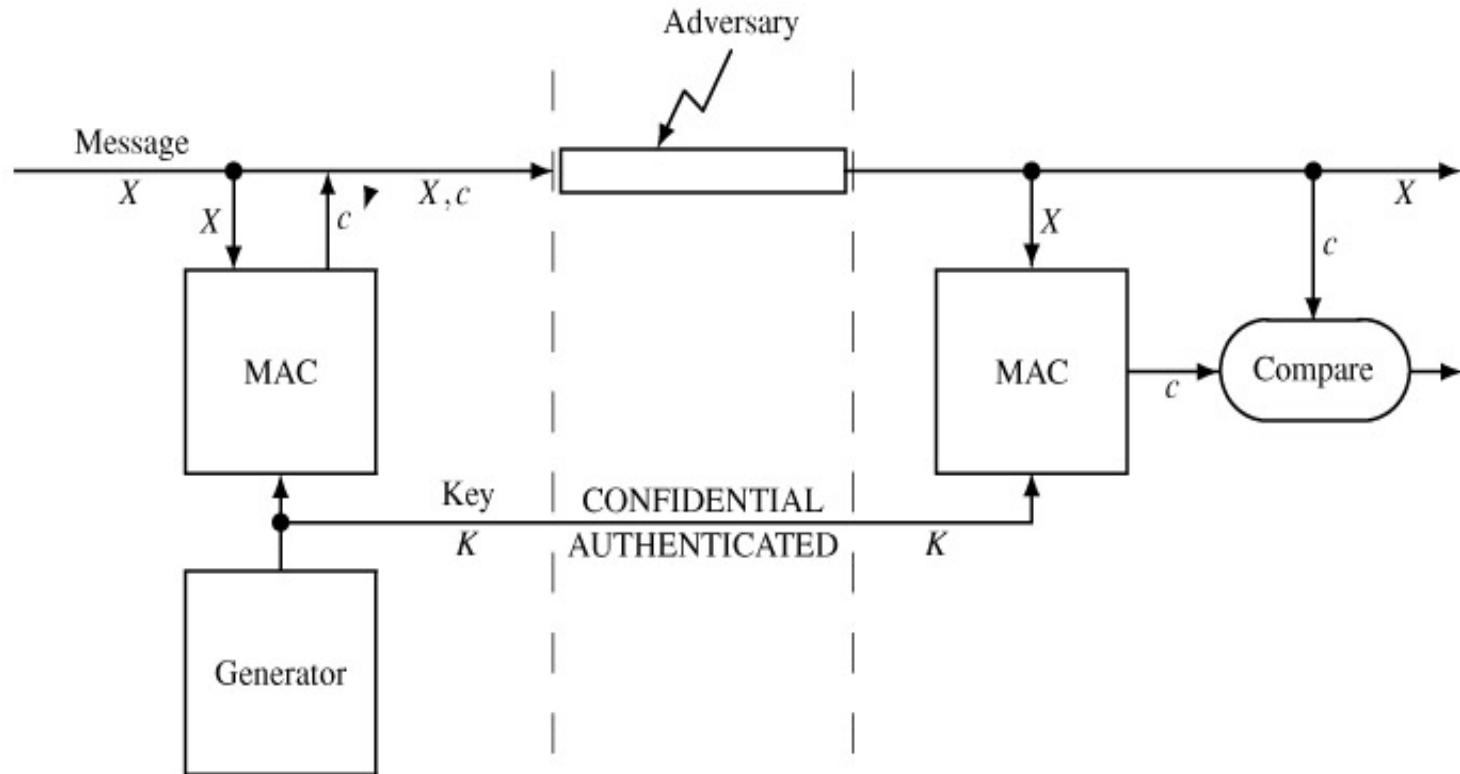
## Segunda pré-imagem



# MAC – Message authentication code



- Duas ou mais partes que compartilham uma chave secreta podem detectar modificação em uma mensagem em trânsito.
- Integridade e autenticidade da mensagem e não da origem de informação.
- HMAC = Funções de hash com chave
  - HMAC em que H é uma função de hash segura





# Exemplos de funções de hash e MAC

Funções de resumo criptográfico quebradas, obsoletas ou em risco

MD2, MD4 e MD5;

SHA-1

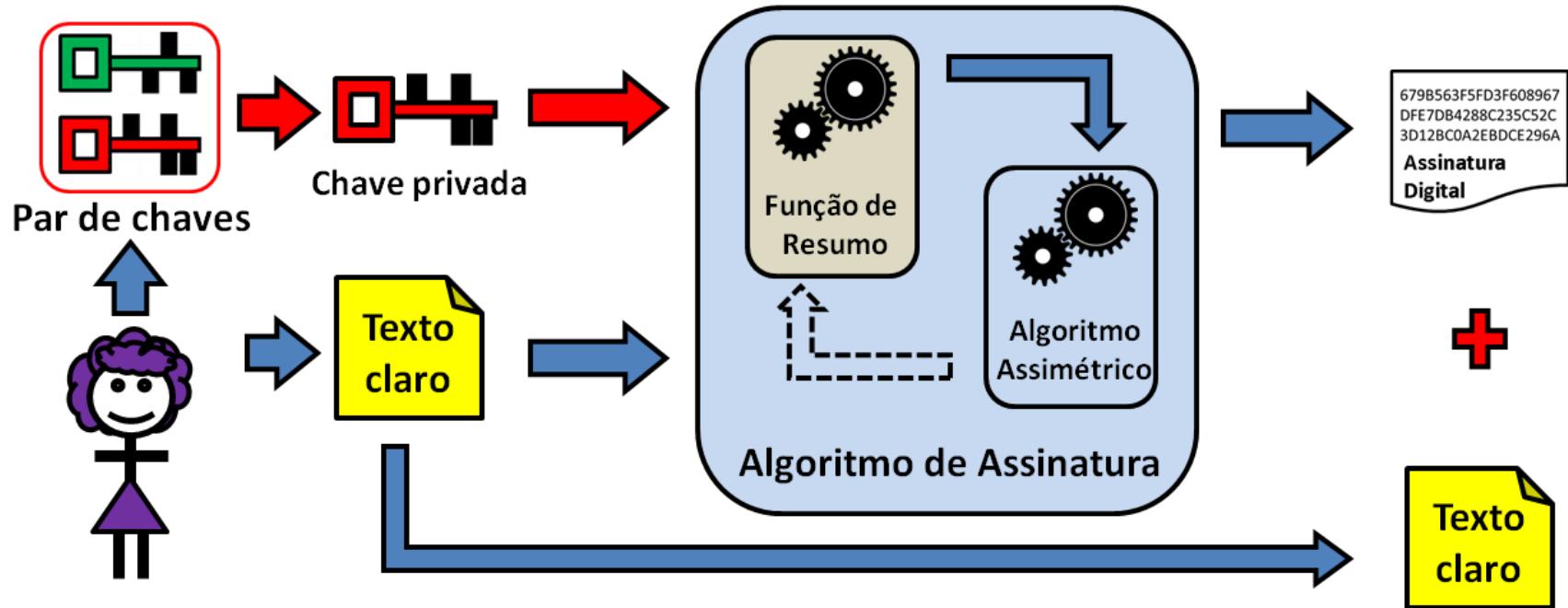
Funções de resumo criptográfico de boa reputação ou seguras para uso

SHA-2 (SHA-224, SHA-256, SHA-384, SHA-512)

SHA-3 (SHA3-224, SHA3-256, SHA3-384, SHA3-512)

Atividade: testar as funções de hash na ferramenta JCrypTool.

# Assinaturas digitais de tamanho fixo



- Ana usa um mecanismo de assinaturas digitais que calcula o *hash* do texto claro usando uma função de resumo criptográfico. O *hash* da mensagem é então assinado com a chave privada de Ana usando um mecanismo de assinatura digital. Esta assinatura digital fixa é enviada junto com o texto claro original e está acessível para quem precisar verificar a autoria do texto claro.
- Ana não pode mais negar (refutar) a autoria do texto claro, pois há uma assinatura digital feita com sua chave privada pessoal. Ninguém precisa da ajuda de Ana para verificar a autoria do documento, desde que a chave pública de Ana esteja amplamente disponível. Por isso, a assinatura digital é irrefutável.

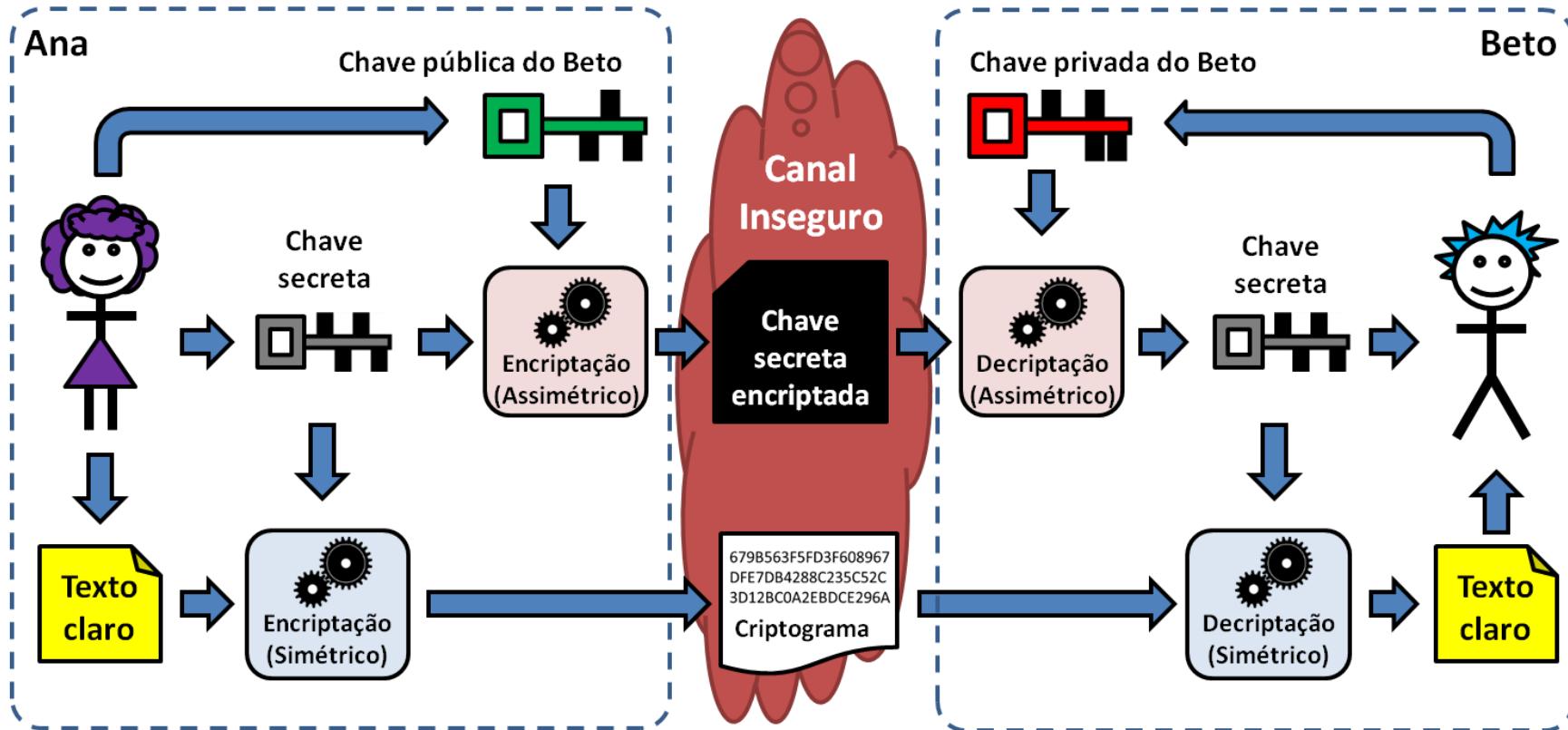
# Sistemas criptográficos híbridos



Criptografia Simétrica	Criptografia Assimétrica
Desempenho superior (mais rápida).	Desempenho inferior (mais lenta).
Não oferece autenticação forte, o segredo é compartilhado.	Autenticação forte com assinaturas digitais.
Não é irrefutável (a autoria pode ser negada).	Assinatura digital é irrefutável.
Distribuição de muitas chaves é trabalhosa.	Distribuição de muitas chaves é simplificada.

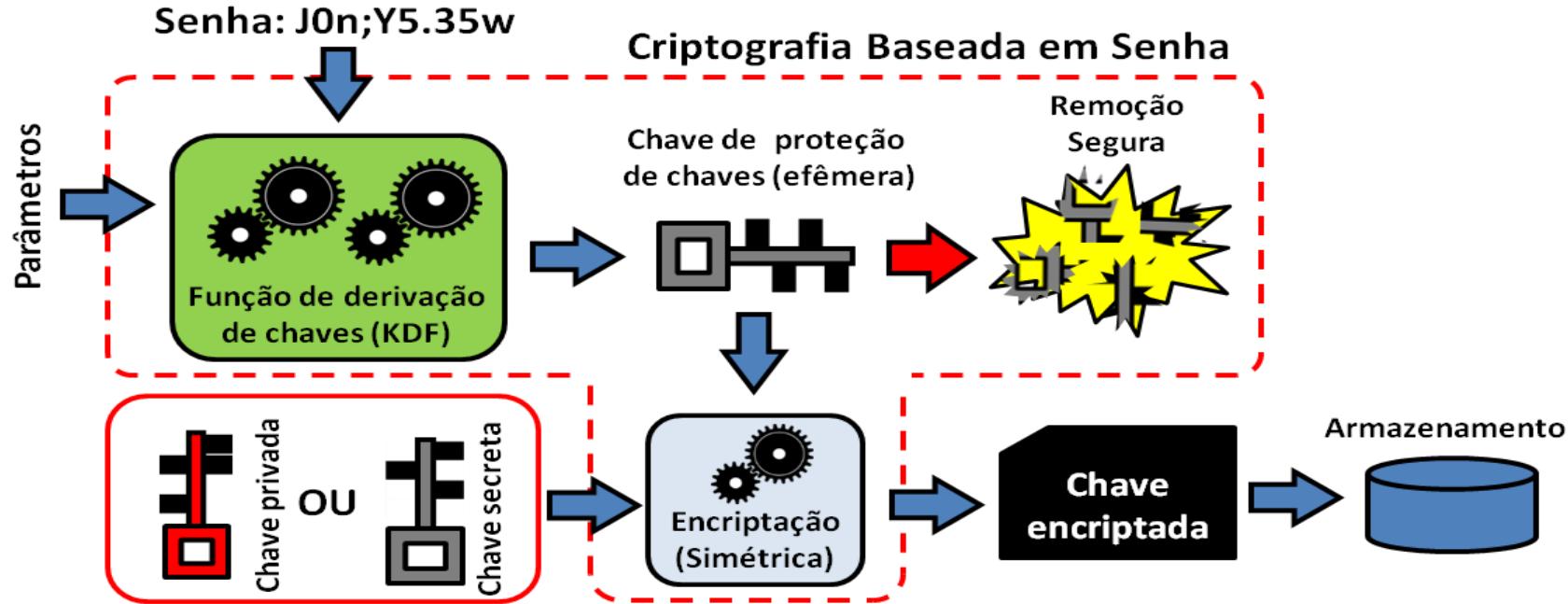
- A criptografia de chave pública é recomendada para grupos grandes e ambientes dinâmicos e públicos. Por outro lado, a criptografia de chave secreta é recomendada para grupos pequenos e estáticos.
- A Tabela compara as tecnologias criptográficas simétricas e assimétricas. Observa-se que as deficiências de uma tecnologia são complementadas pelas vantagens da outra.
- De fato, uma solução amplamente utilizada pelos protocolos de comunicação segura na Internet usa uma combinação das tecnologias simétrica e assimétrica, chamada de sistemas criptográficos híbridos.

# Sistema criptográfico híbrido



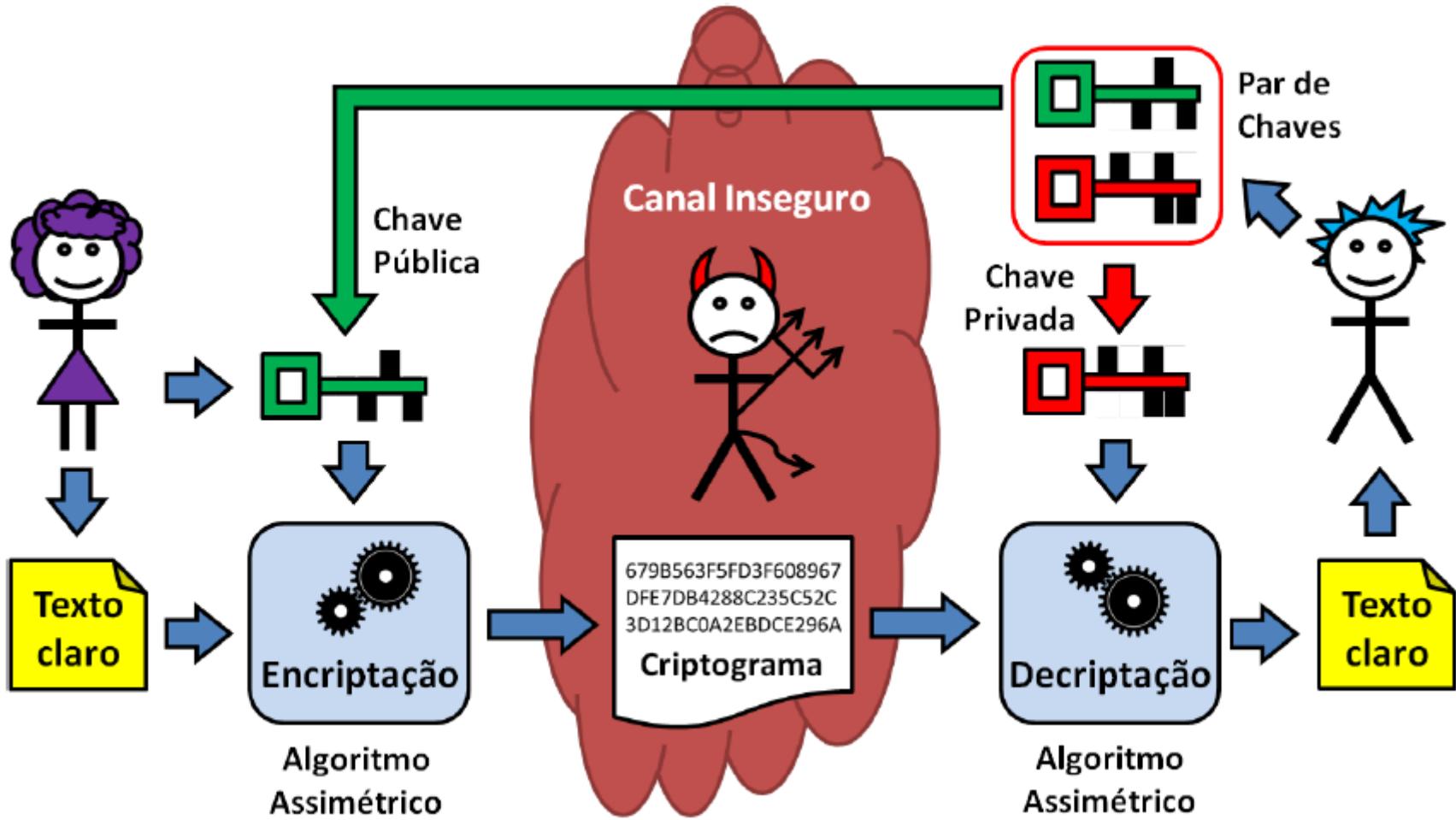
- O sistema híbrido oferece o desempenho superior da criptografia simétrica com a autenticação forte e a facilidade de distribuição de chaves da criptografia assimétrica.
- Esta combinação pode ser implementada pelo uso de uma chave secreta encriptada por uma chave pública para transporte seguro. A criptografia assimétrica é usada como canal seguro para o compartilhamento da chave secreta.

# Armazenamento seguro de chaves em software



- O sigilo das chaves de encriptação e assinatura é da maior importância para a segurança dos sistemas criptográficos. Nas implementações em software, sem auxílio de hardware de segurança, as chaves privadas e secretas devem ser guardadas de forma encriptada.
- A Encriptação Baseada em Senhas (*Password-Based Encryption* - PBE) proporciona os meios adequados para gerar uma chave criptográfica a partir de uma senha.
- O mecanismo de PBE é geralmente utilizado para proteger chaves logo que elas são criadas, para reduzir a exposição destas chaves.

# Encriptação assimétrica para sigilo

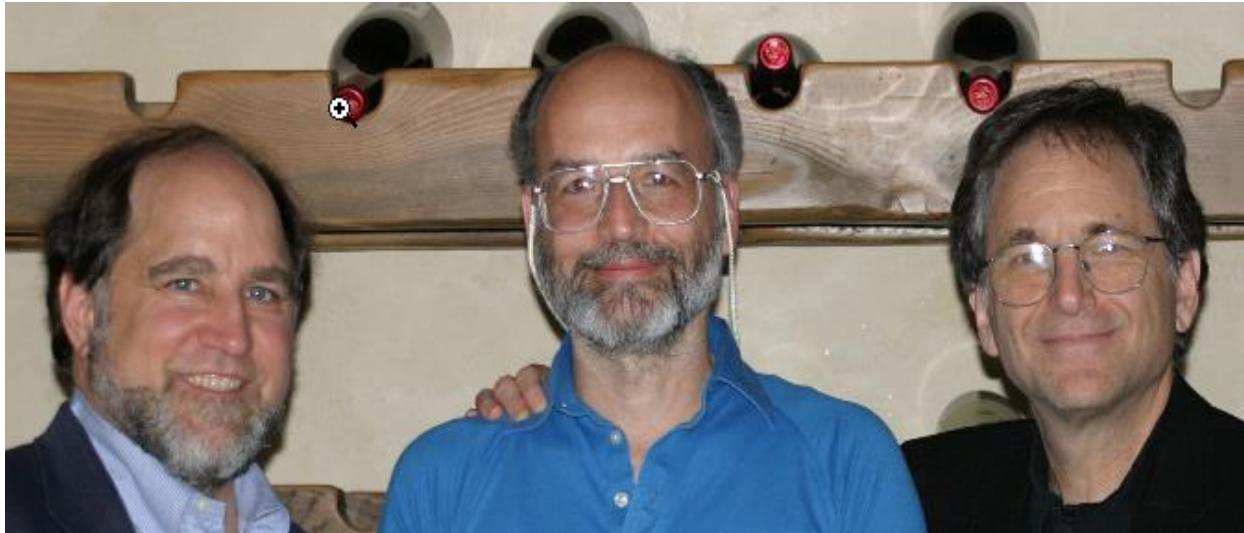


# Exemplos de sist. criptográficos assimétricos



- O algoritmo RSA canônico
- Acordos de chaves com Diffie-Hellman (DH)
- Criptografia de Curvas Elípticas (ECC)
- Outros
  - ElGamal, ECDSA, ECDH, RSA-OAEP, RSA-PSS;

# Rivest-Shamir-Adleman (RSA)

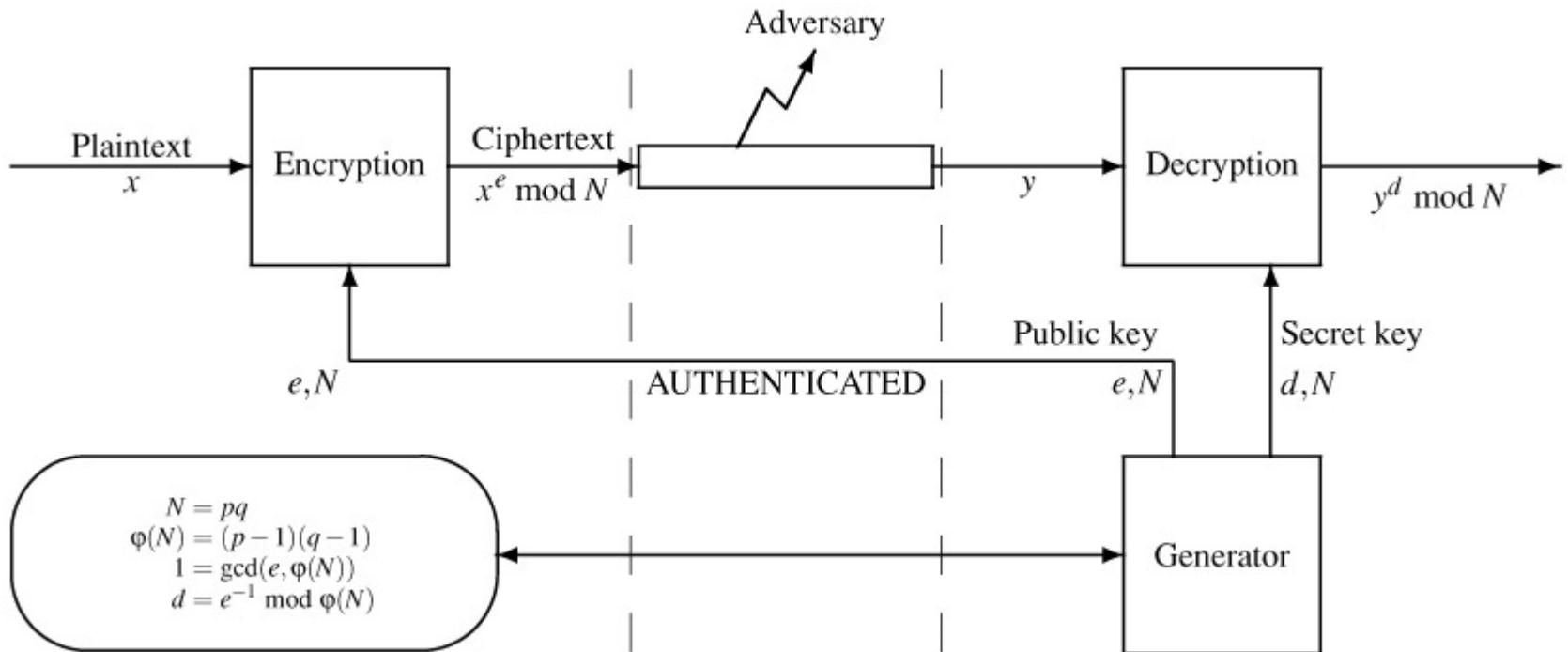


Ronald Rivest

Adi Shamir

Leonard Adleman

# O algoritmo RSA nos livros

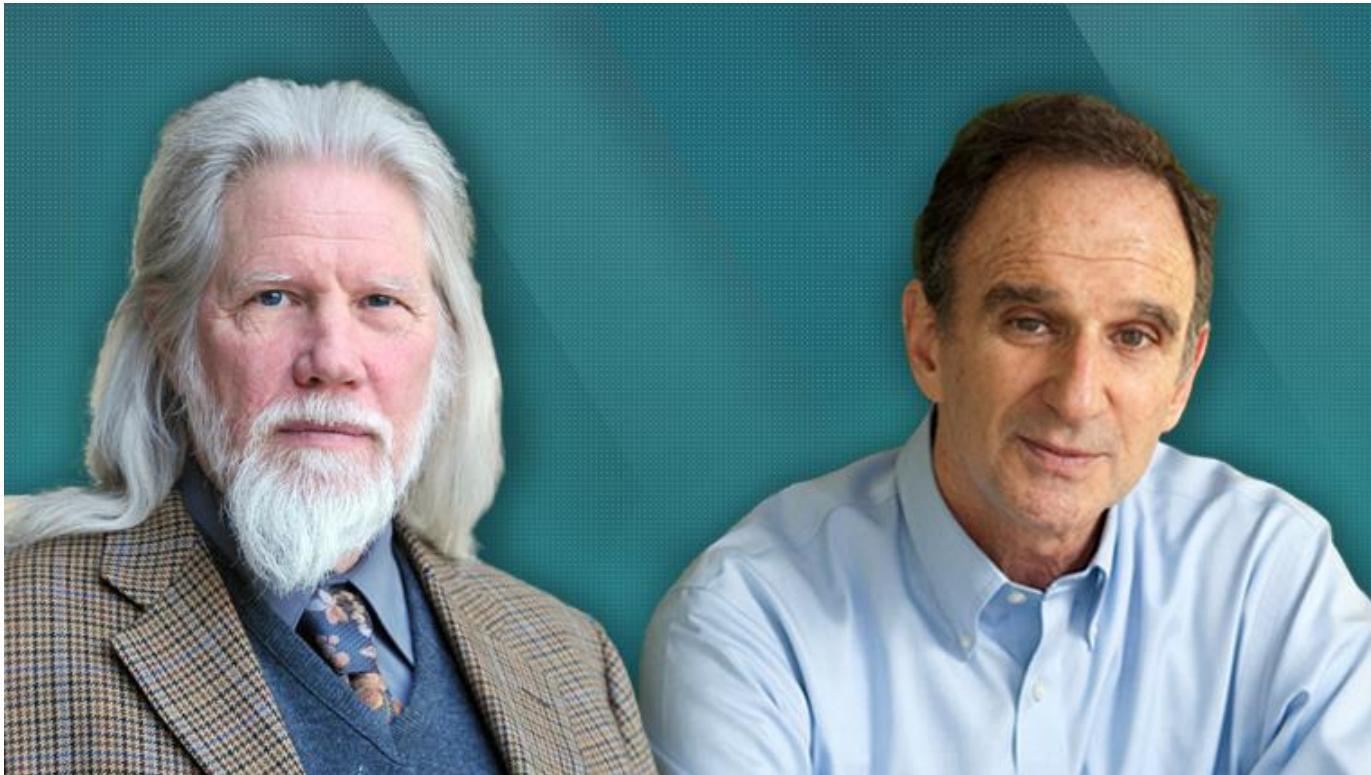


# Acordo de chaves



- Há ocasiões em que entidades que nunca tiveram a oportunidade de compartilhar chaves criptográficas (por exemplo, nunca se encontram ou não se conhecem) precisam se comunicar em sigilo.
- Nestes casos, uma chave efêmera, usada apenas para algumas encriptações e decriptações decorrentes de uma conversa, pode ser gerada momentos antes do início da conversa.
- Os métodos de acordo de chaves são utilizados para combinar ou negociar uma chave secreta entre dois ou mais participantes usando um canal público.
- Uma característica interessante destes métodos é que o segredo compartilhado (a partir do qual a chave será derivada) é combinado pela troca de informações públicas por meio de um canal inseguro.

# Acordo de chaves Diffie-Hellman (DH)

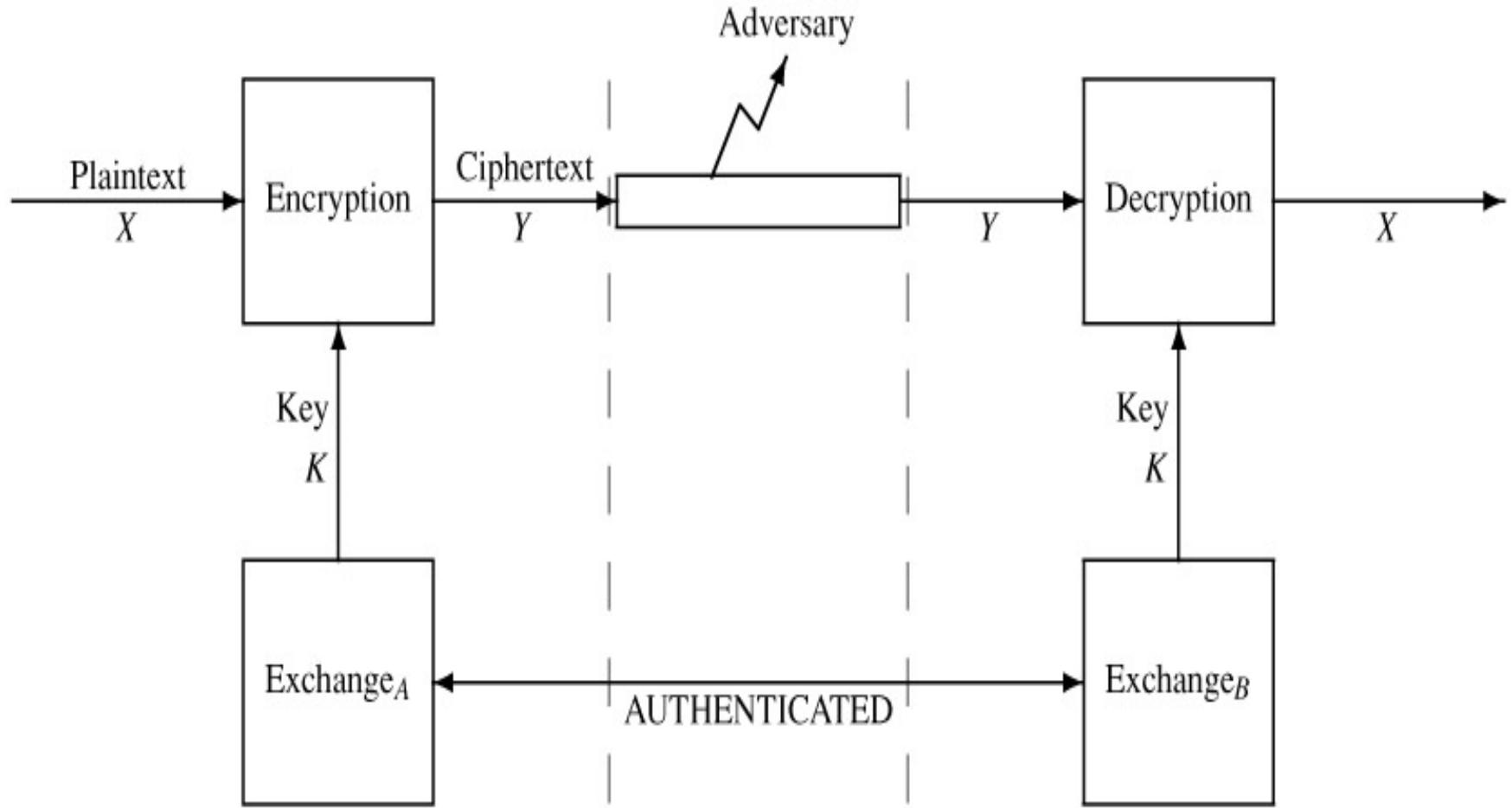


Whitfield Diffie

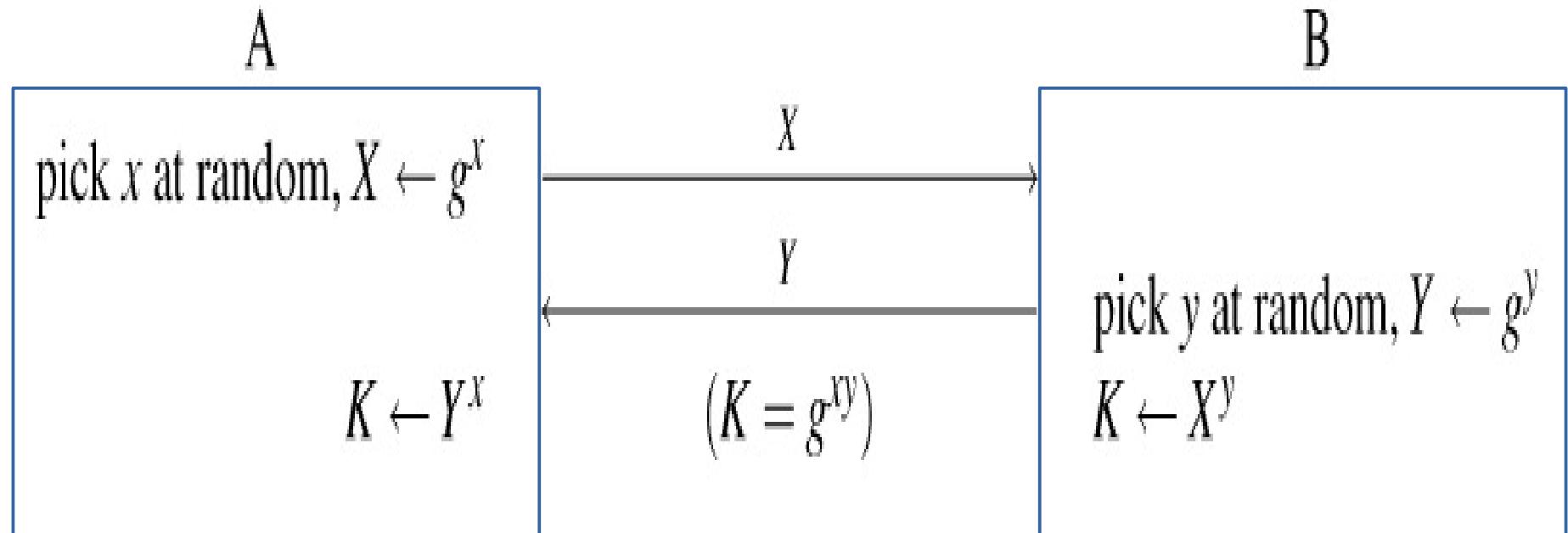
Martin Hellman



# O cenário geral – por quê DH ?



# O protocolo de acordo de chaves Diffie-Hellman



# Ataque Man-in-the-Middle no prot. DH



A

pick  $x, X \leftarrow g^x$

$X$

$K_1 \leftarrow (Y')^x$

E

pick  $x', X' \leftarrow g^{x'}$

$X'$

pick  $y', Y' \leftarrow g^{y'}$

$Y$

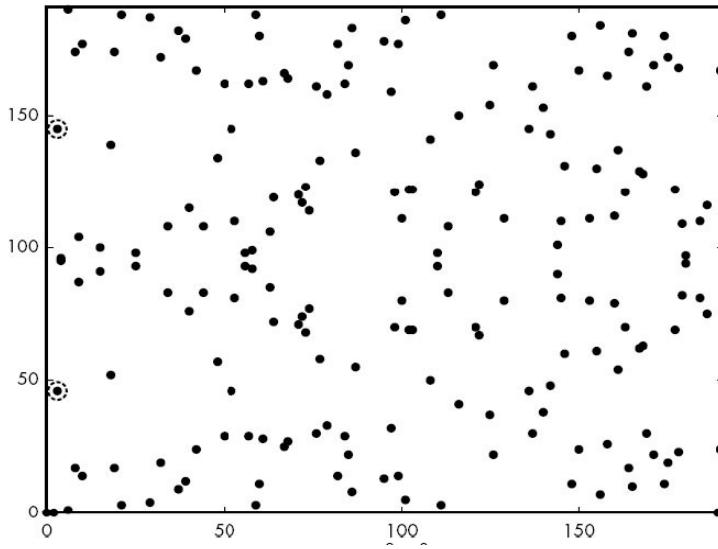
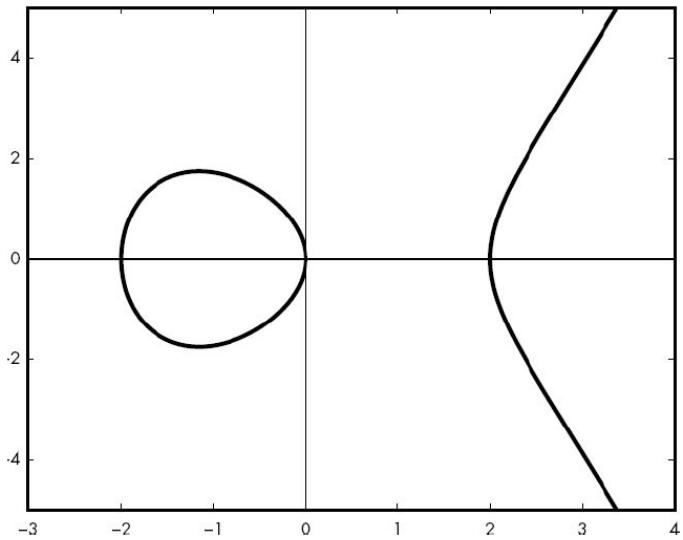
$K_1 \leftarrow X^{y'}, K_2 \leftarrow Y^{x'}$

B

pick  $y, Y \leftarrow g^y$

$K_2 \leftarrow (X')^y$

# Criptografia de curvas elípticas



- O que é uma curva elíptica para criptografia?
  - À esquerda, a curva  $y^2 = x^3 - 4x$  sobre os números reais.
  - À direita, a curva  $y^2 = x^3 - 4x \text{ mod } 191$ , sobre inteiros no corpo primo  $Z_{191}$ .
- Aplicações criptográficas de curvas elípticas incluem troca de chaves, assinaturas digitais e encriptação. Os esquemas/protocolos sobre curvas elípticas mais utilizados atualmente:
  - Protocolo de acordo de chaves *Elliptic Curve Diffie-Hellman* (ECDH)
  - Esquema de assinaturas digitais *Elliptic Curve Digital Signature Algorithm* (ECDSA)
  - Encriptação *Elliptic Curve Integrated Encryption Scheme* (ECIES)

Figuras: Jean-Philippe Aumasson . Serious Cryptography: A Practical Introduction to Modern Encryption. No Starch Press, 2018, capítulo 12.



# Curvas elípticas seguras e inseguras

- Curvas NIST:
  - P-192 (secp192r1), P-224 (secp224r1), P-256 (secp256r1), P-384 (secp384r1) e P-521 (secp521r1)
  - Curve448 com 244 bits de segurança para uso no ECDH
- A curva de Bernstein Curve25519, Ed25519, veloz e com 128 bits de segurança\*

## Exemplos de curvas elípticas seguras no padrão SEC-2 v2

Segurança	Curvas sobre $F_p$	Curvas sobre $F_{2^m}$
80	—	sect163k1, sect163r1, sect163r2
96	secp192k1, secp192r1	—
112	secp224k1, secp224r1*	sect233k1, sect233r1
115	—	sect239k1
128	secp256k1*, secp256r1*	sect283k1, sect283r1
192	secp384r1*	sect409k1, sect409r1
256	secp521r1	sect571k1, sect571r1

## Exemplos de curvas elípticas inseguras conforme padrão SEC-2 v2

Segurança	Curvas sobre $F_p$	Curvas sobre $F_{2^m}$
56	secp112r1, secp112r2	sect113r1, sect113r2
64	secp128r1, secp128r2	sect131r1, sect131r2
80	secp160k1, secp160r1, secp160r2	—
96	—	sect193r1, sect193r2

\*Fonte <https://cr.yp.to/ecdh.html>

# Tamanhos de chave e níveis de segurança

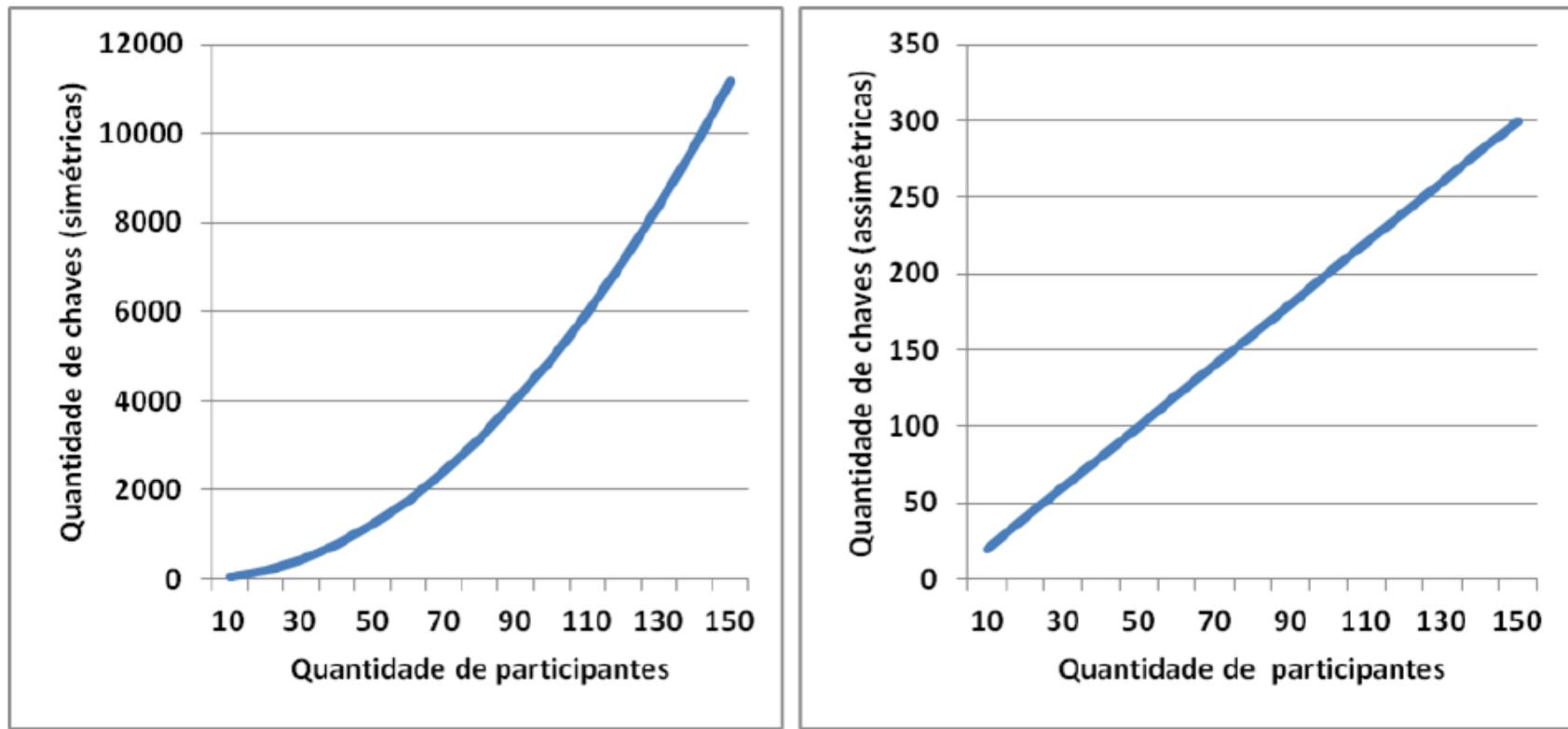


**Tabela 2.1. Níveis de segurança e tamanhos de chave (de [BlueKrypt]).**

Nível de segurança	Fatoração (IFP)	DLP		Curva Elíptica	Hash	
		chave	Corpo		Assinatura	KDF/HMAC/PRNG
80	1024	160	1024	160–163	SHA1 (160)	–
112	2048	224	2048	224–233	SHA-224/512 SHA3-224	–
128	3072	256	3072	256–283	SHA-256/512 SHA3-256	SHA1(160)
192	7680	384	7680	384–409	SHA-384 SHA3-384	SHA-224 SHA-512
256	15360	512	15360	512–571	SHA-512 SHA3-512	SHA-256/384/512 SHA-512/SHA3-512

# Gestão de chaves e certificação digital

# Distribuição de chaves com certificação digital



**Figura 2.3. Quantidades de chaves assimétricas e simétricas para até 150 participantes.**

# Distribuição de chaves com certificação digital

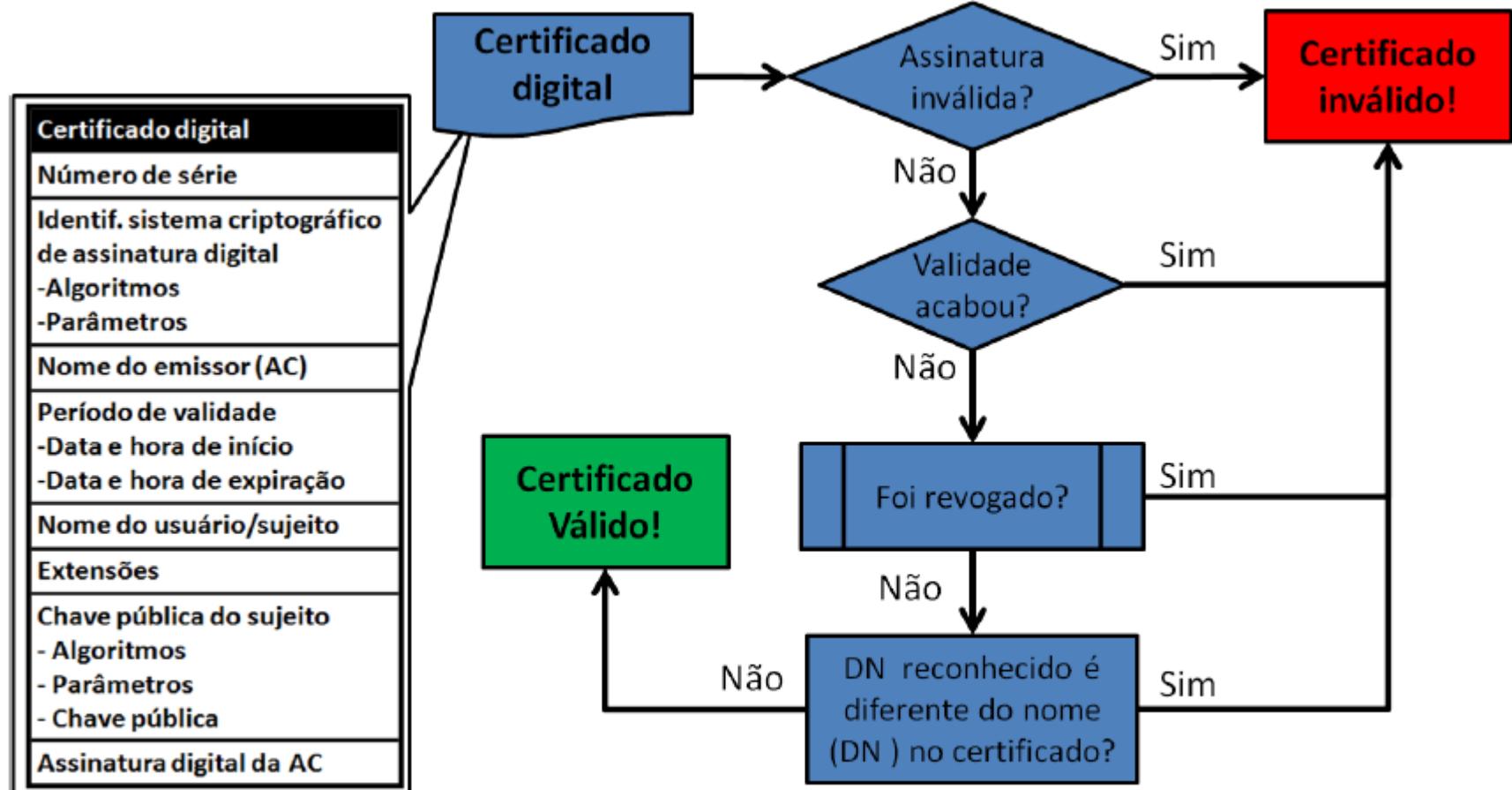


Figura 2.4. Fluxograma de validação de um certificado digital (de [Braga and Dahab 2015b]).

# Distribuição de chaves com certificação digital

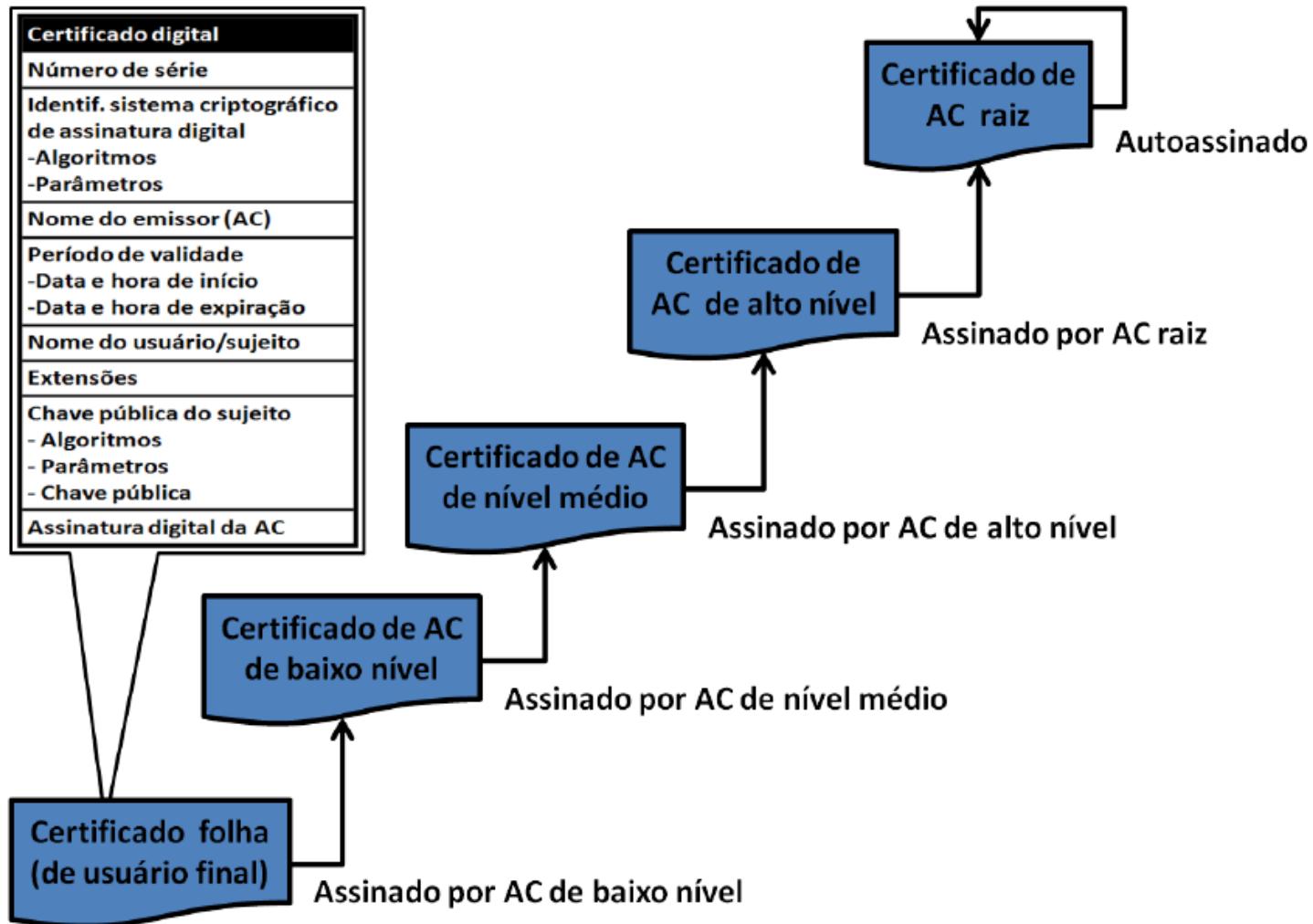


Figura 2.5. Cadeia hierárquica de certificação digital (de [Braga and Dahab 2015b]).

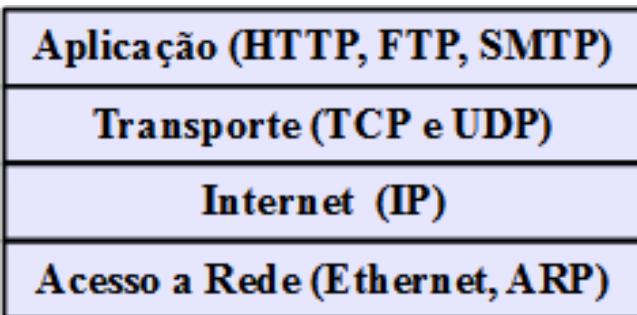


# A vida de uma chave criptográfica

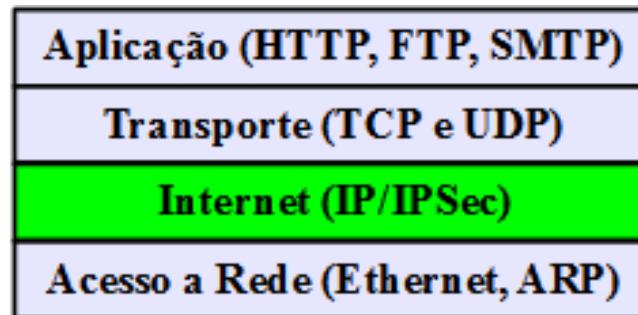
Estados da chave	Fases do ciclo de vida	Atividades de cada fase
<pre>graph TD; A[Pré-ativada] --&gt; B[Ativada]; B --&gt; C[Desativada]; B --&gt; D[Comprometida]; C --&gt; E[Destruída]; D --&gt; E; E --&gt; F[Destruída Comprometida];</pre>	Pré-operacional	<ul style="list-style-type: none"><li>• Registro do usuário;</li><li>• Inicialização do sistema;</li><li>• Inicialização do usuário;</li><li>• Instalação do material da chave;</li><li>• Estabelecimento (criação) da chave;</li><li>• Registro da chave.</li></ul>
	Operacional	<ul style="list-style-type: none"><li>• Armazenamento e acesso;</li><li>• Back-up e recuperação;</li><li>• Substituição (reemissão) de chaves;</li><li>• Derivação de chaves.</li></ul>
	Pós-operacional	<ul style="list-style-type: none"><li>• Arquivamento e recuperação;</li><li>• Bloqueio/remoção do usuário;</li><li>• Bloqueio da chave para uso normal;</li><li>• Destrução da chave;</li><li>• Revogação da chave.</li></ul>
	Destruída	<ul style="list-style-type: none"><li>• Apagamento seguro da chave;</li><li>• Apagamento seguro de dados;</li><li>• Asseguração de apagamento;</li><li>• Re-criptação de dados.</li></ul>

# Security Sockets Layer / Transport Layer Security (SSL/TLS)

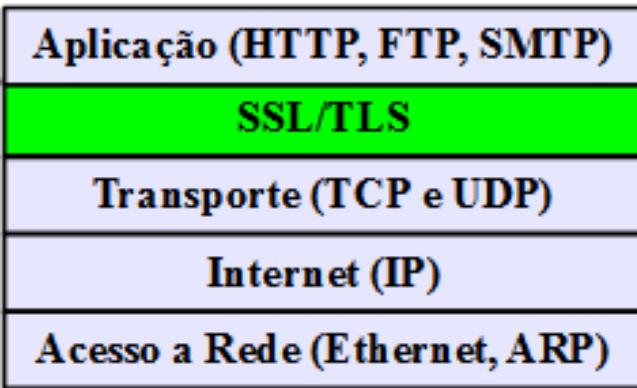
# SSL/TLS e outros protocolos cripto. comuns



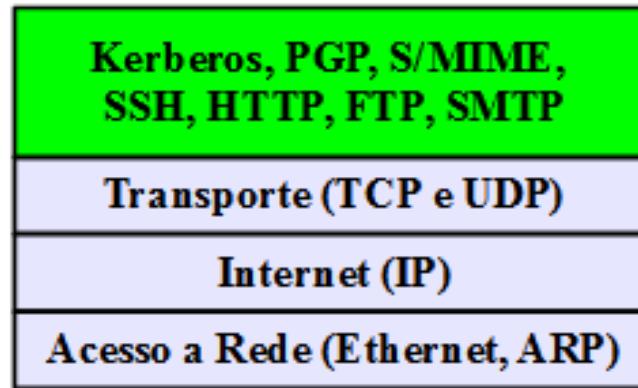
a) Pilha de Protocolos TCP/IP



b) IPSec na Camada de Rede



c) SSL e TLS com o Serviços da Camada de Transporte



d) Vários Protocolos de Segurança e Criptografia na Camada de Aplicação

# Uma breve história do SSL



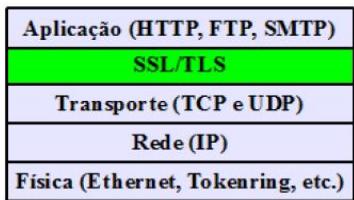
- A história do protocolo SSL se confunde com a história do comércio eletrônico
- A Netscape dominava os mercados de web browsers e de servidores web
  - Ela distribuía gratuitamente o browser, mas vendia o servidor web
  - O único na época com proteções contra monitoramento e interceptação da comunicação
- A segurança da comunicação era garantida por um protocolo criptográfico na camada de transporte do TCP/IP que ficou conhecido como SSL
- A última versão do SSL original foi a SSLv3, com inovações disponíveis até hoje:
  - Autenticação do servidor com assinaturas digitais e certificados X.509, com os algoritmos RSA, DSA e ECDSA, ou uma chave simétrica pré-configurada
  - Autenticação (opcional) do cliente com assinaturas digitais e certificados X.509
  - Sigilo com encriptação da informação trocada entre cliente e servidor
  - Integridade e autenticidade da informação trocada entre cliente e servidor

# SSL/TLS na pilha TCP/IP e o handshake

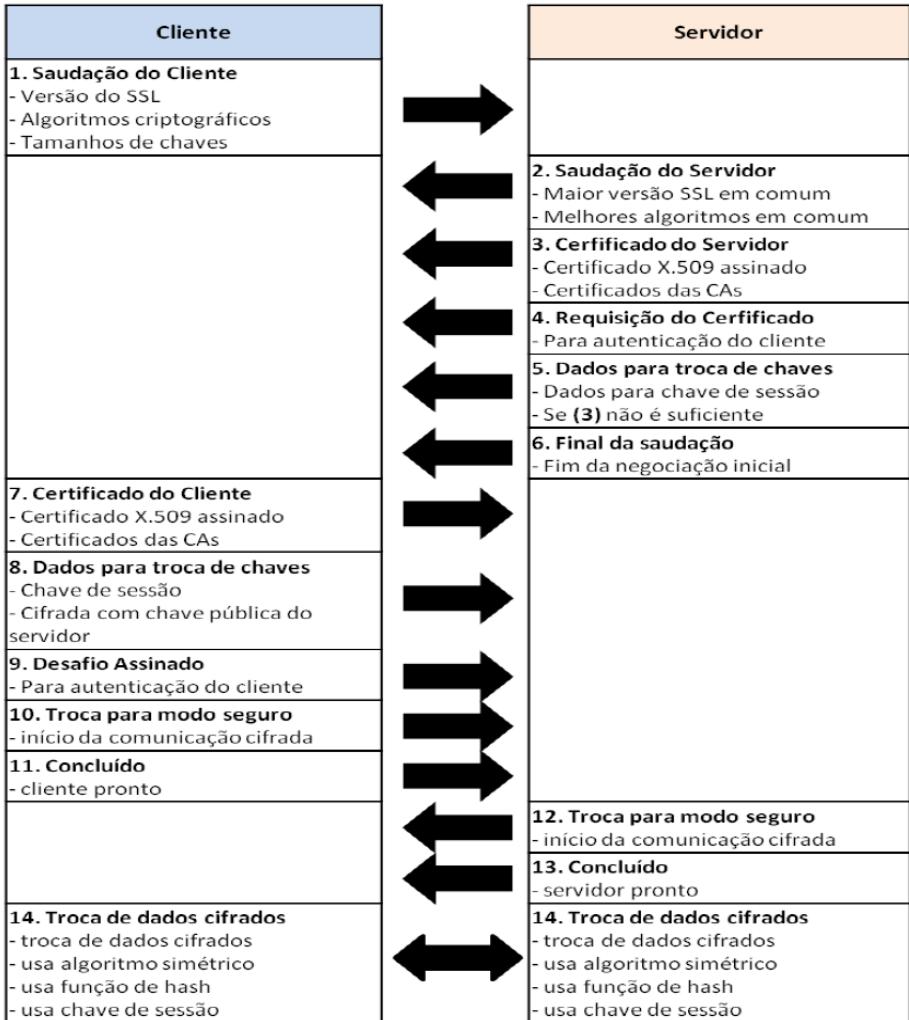


## Versões do SSL/TLS

- SSLv2 é considerado inseguro e não deve mais ser usado
- SSLv3 é obsoleto e não deve mais ser utilizado
- TLSv1.0 (RFC 2246) ainda é usada em sistemas legados
- TLSv1.1 (RFC 4346) é uma versão recente e sem vulnerabilidades sem mitigação
- TLSv1.2 (RFC 5246) era a versão atual até Agosto 2018
- TLSv1.3 (RFC 8446) foi lançado na forma final em Agosto de 2018



Localização do Protocolo SSL/TLS na Pilha TCP/IP.

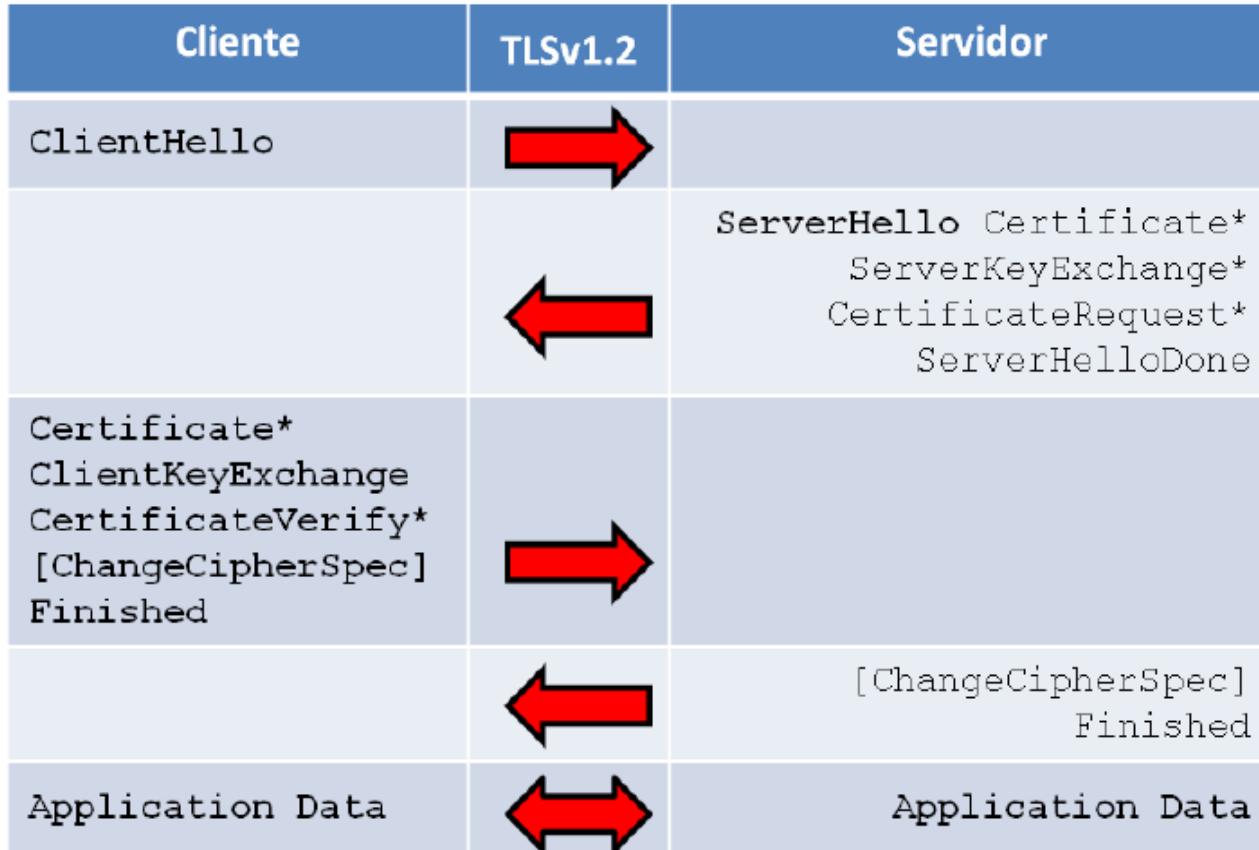


# Versões do OpenSSL e suas vulnerabilidades



- SSLv2 é vulnerável a vários ataques
  - BEAST (Browser Exploit against SSL/TLS)
  - DROWN (Decrypting RSA with Obsolete and Weakened eNcryption)
  - Vulnerabilidades Cipher Suite Rollback e ChangeCipherSpec Message Drop
- HTTPS sobre **SSLv3** vulnerável a vários ataques
  - POODLE (Padding Oracle on Downgraded Legacy Encryption)
  - Lucky 13
  - Vulnerabilidades Version Rollback e Key Exchange Algorithm Confusion
- TLSv1.0 (RFC 2246) ainda é usada em sistemas legados
  - TLSv1.0 é vulnerável aos ataques BEAST e Lucky 13. Esta versão inicial do TLS também é vulnerável aos ataques de negação de serviço e que exploram falhas na renegociação
- TLSv1.1 (RFC 4346) é uma versão relativamente recente e que não apresenta vulnerabilidades conhecidas sem mitigação, mas ainda oferece algoritmos criptográficos antigos
- TLSv1.2 (RFC 5246) era a versão atual até Agosto de 2018, quando foi lançada a versão nova, e já oferece esquemas criptográficos novos com **encriptação autenticada**
- TLSv1.3 (RFC 8446) foi lançado na forma final em Agosto de 2018 e representa o rompimento definitivo como passado pela eliminação de diversas características inseguras ou obsoletas mantidas até a versão anterior por compatibilidade com legados

# A versão 1.2 do TLS (protocolo de handshake)



**Figura 5.1. Estabelecimento de sessão e envio de mensagens SSL/TLS v1.2.**

# A versão 1.3 do TLS (protocolo de handshake)



Cliente	TLSv1.3	Servidor
<b>ClientHello</b> +key_share* +signature_algorithms* +psk_key_exchange_modes* +pre_shared_key*		
		<b>ServerHello</b> + key_share* + pre_shared_key* EncryptedExtensions} {CertificateRequest*} {Certificate*} {CertificateVerify*} {Finished} [Application Data*]
{Certificate*} {CertificateVerify*} {Finished}		
Application Data		Application Data

**Figura 5.2. Estabelecimento de sessão e envio de mensagens SSL/TLS v1.3.**

# Limitações do SSL/TLS



- O SSL/TLS não resolve todos os problemas de segurança de transações eletrônicas na Internet
  - SSL/TLS protege a comunicação entre cliente e servidor HTTPS
- RFC do SSL/TLS não diz como adicioná-lo a um protocolo de aplicação
  - Decisões de projeto dos protocolos de aplicação:
    - Em que momento o *handshake* deve ser iniciado?
    - Qual a semântica da autenticação dos certificados digitais?
- Ataques recentes abusam da confiança em sites web que possuem conexões SSL/TLS com certificados legítimos
  - Uma conexão SSL/TLS legítima (autêntica) é usada em ataques
  - Sites maliciosos (p.ex. *phishing sites*), podem ter certificados legítimos
  - Autoridades certificadoras emitem certificados gratuitos e de validade curta que podem ser usados por tempo suficiente para viabilizar ataques de engenharia social

## Parte 2-2 (Laboratório)



# Será possível viver assim?



Mafalda (Quino, 1932 - 2020)

# Criptografia para Programadores e SysAdmins



SBSeg 2015

XV Simpósio Brasileiro em Segurança da Informação de Sistemas Computacionais — SBSeg 2015

## Capítulo

1

### Introdução à Criptografia para Programadores: Evitando Maus Usos da Criptografia em Sistemas de Software

Alexandre Braga e Ricardo Dahab

#### Abstract

*Studies have shown that vulnerabilities in cryptographic software are generally caused by implementation defects and mismanagement of cryptographic parameters. In addition, we see the recurring presence of several cryptographic bad practices in various software and mobile applications, in particular. Possibly, these vulnerabilities were included unintentionally by inexperienced programmers without expert support. Along this vein, this short course addresses the programmatic use of cryptography by software developers with little or no experience in information security and cryptography. The material is introductory and aims to show software developers, through real examples and code snippets, the good and bad uses of cryptography and thus facilitate further improvements in future studies.*

#### Resumo

*Estudos têm revelado que vulnerabilidades em softwares criptográficos são causadas em geral por defeitos de implementação e pela má gestão de parâmetros criptográficos. Além disso, percebe-se a presença recorrente de diversas práticas ruins da criptografia em softwares diversos e aplicativos móveis, em particular. Possivelmente, estas vulnerabilidades foram incluídas com intenção por programadores inexperientes e sem apoio de especialistas. Desta forma, este minicurso aborda a utilização programática de criptografia por desenvolvedores de software com pouca ou nenhuma experiência em segurança da informação e criptografia. O material é introdutório e tem o objetivo de mostrar aos programadores de software, por meio de exemplos reais e trechos de código, os bons e maus usos da criptografia e, assim, facilitar o aprofundamento em estudos futuros.*

Livro-texto de Minicursos

1

©2015 SBC — Soc. Bras. de Computação

SBSeg 2018

## Capítulo

2

### Criptografia Assimétrica para Programadores – Evitando Outros Maus Usos da Criptografia em Sistemas de Software

Alexandre Braga (UNICAMP) e Ricardo Dahab (UNICAMP)

#### Abstract

*The widespread misuse of cryptography in software systems is the most frequent source of cryptography-related security problems. Several misuses of cryptography have been found to be recurrent in software in general, resulting in vulnerabilities exploitable in real attacks. There is a huge gap between what cryptologists see as misuses of cryptography and what developers see as unique uses of cryptographic technology. This chapter contributes to fill this gap by addressing the programmatic use of asymmetric (public key) cryptography by software developers with little or no experience in information security and cryptography. The text is introductory and aims at showing to software programmers, through actual examples and code snippets, the gooduses and misuses of asymmetric cryptography and facilitate further studies.*

#### Resumo

*O mau uso generalizado da criptografia em sistemas software é a fonte mais frequente de problemas de segurança relacionados à criptografia. Diversos maus usos de criptografia são considerados recorrentes em software em geral, resultando em vulnerabilidades exploráveis em ataques reais. Percebe-se uma grande lacuna entre o que os criptologos veem como maus usos de criptografia e aquilo que os desenvolvedores veem como uso inegável da tecnologia criptográfica. Este texto contribui para preencher essa lacuna, abordando a utilização programática da criptografia assimétrica (de chave pública) por desenvolvedores de software com pouca ou nenhuma experiência em segurança da informação e criptografia. O texto é introdutório e tem o objetivo de mostrar aos programadores de software, por meio de exemplos reais e trechos de código, os bons e maus usos da criptografia assimétrica e, assim, facilitar o aprofundamento em estudos futuros.*

50

SBRC 2019

## Capítulo

5

### Introdução à criptografia para administradores de sistemas com TLS, OpenSSL e Apache mod\_ssl

Alexandre Braga (UNICAMP) e Ricardo Dahab (UNICAMP)

#### Abstract

*The incorrect use of cryptographic infrastructures is one of the most common sources of cryptography-related security issues: avoidable errors related to configuration of algorithms and protocols become recurrent, often resulting in exploitable vulnerabilities and actual attacks on computing systems. Hence, there is a growing demand for practical real-world guidance on how to avoid incorrect cryptographic configurations that lead to exploitable vulnerabilities in computer networks and distributed systems. This chapter addresses the use of cryptography by students and IT professionals with little experience in information security and cryptography. The text covers OpenSSL, the TLS protocol (including the new version 1.3 launched in August 2018), and security settings for the mod\_ssl module integrated to the Apache web server.*

#### Resumo

*O uso incorreto de infraestruturas criptográficas é uma das fontes mais comuns de problemas de segurança relacionados à criptografia, quando diversos erros evitáveis na configuração de algoritmos e de protocolos se tornam recorrentes, resultando frequentemente em vulnerabilidades exploráveis em ataques reais aos sistemas de computação. Por isso, há uma demanda crescente por orientações práticas, do mundo real, sobre como evitar configurações incorretas da criptografia que levam a vulnerabilidades exploráveis em redes de computadores e sistemas distribuídos. Este capítulo aborda a utilização de criptografia por estudantes e profissionais de TI com pouca experiência em segurança da informação e criptografia. O minicurso cobre o OpenSSL, o protocolo TLS (incluindo a nova versão 1.3 de agosto de 2018) e as configurações seguras do módulo mod\_ssl do servidor web Apache.*

61

# Todos os scripts estão disponíveis



<https://bitbucket.org/alexmbraga/crypto4developers>

The screenshot shows a Bitbucket repository page. At the top, there's a header with the repository name 'alexmbraga/crypto4developers'. Below the header, the repository owner's email 'alexmbraga2007@gmail.com' and the repository name 'Crypto4Developers' are displayed. A 'Clone' button and a three-dot menu button are also visible. The main content area shows a brief description: 'Crypto4Developers is a repository for source code used in two tutorials related to teaching cryptography to ordinary developers.' Below this, there are dropdown menus for 'master' branch and 'Filter files', and a search bar. The file list table has columns for Name, Size, Last commit, and Message. The table contains the following data:

Name	Size	Last commit	Message
mc2015		2018-08-01	source code from SBSeg2015
mc2018		22 hours ago	tutorial on openssl
mc2019		19 hours ago	update
README.md	1.11 KB	21 hours ago	README.md edited to include reference to a new tutorial

# Preenchimento de blocos e modos de operação das Cifras de Bloco

# Preenchimento (padding) na encriptação de blocos



- Completar a string binária do texto claro para que ele seja um múltiplo do tamanho do bloco.
  - PKCS7Padding é um esquema de *padding* descrito no documento dos Laboratórios RSA "PKCS #5:Password-Based Encryption Standard".
  - M a mensagem original e PM é a mensagem com *padding*,  $L$  é o tamanho de M em bytes.

# Modos de operação da encriptação de blocos



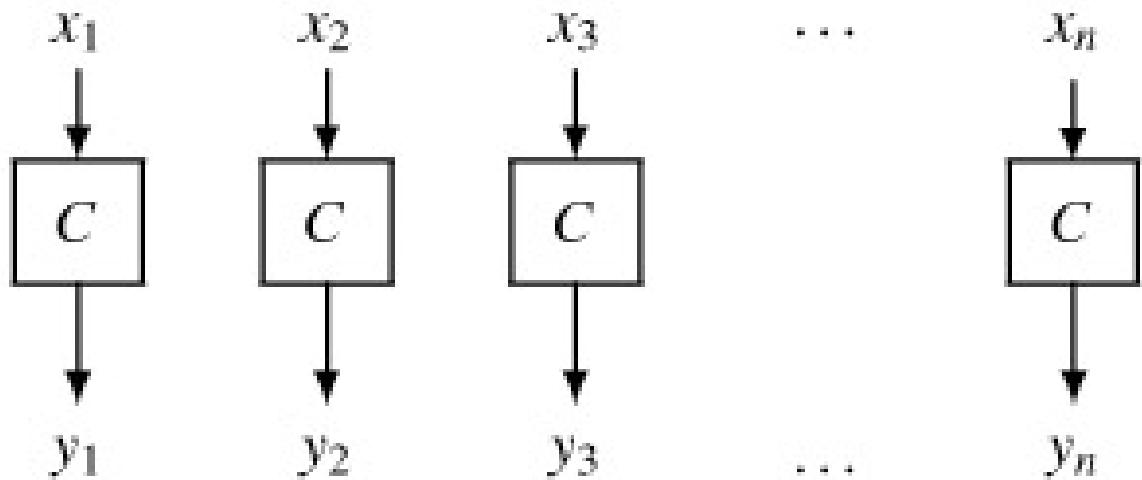
Mode	Description	Typical Application
Electronic Codebook (ECB)	Each block of 64 plaintext bits is encoded independently using the same key.	<ul style="list-style-type: none"><li>Secure transmission of single values (e.g., an encryption key)</li></ul>
Cipher Block Chaining (CBC)	The input to the encryption algorithm is the XOR of the next 64 bits of plaintext and the preceding 64 bits of ciphertext.	<ul style="list-style-type: none"><li>General-purpose block-oriented transmission</li><li>Authentication</li></ul>
Cipher Feedback (CFB)	Input is processed J bits at a time. Preceding ciphertext is used as input to the encryption algorithm to produce pseudorandom output, which is XORed with plaintext to produce next unit of ciphertext.	<ul style="list-style-type: none"><li>General-purpose stream-oriented transmission</li><li>Authentication</li></ul>
Output Feedback (OFB)	Similar to CFB, except that the input to the encryption algorithm is the preceding DES output.	<ul style="list-style-type: none"><li>Stream-oriented transmission over noisy channel (e.g., satellite communication)</li></ul>

# Modos de operação da encriptação de blocos



## ECB (Electronic CodeBook)

- $X = X[1], X[2], X[3], \dots, X[i]$  é o texto claro
- $C_k(X[i])$  é a ecriptação de  $X[i]$  por  $C()$  com chave  $k$
- $Y = Y[1], Y[2], Y[3], \dots, Y[i]$  é o criptograma
- $Y[i] = C_k(X[i])$

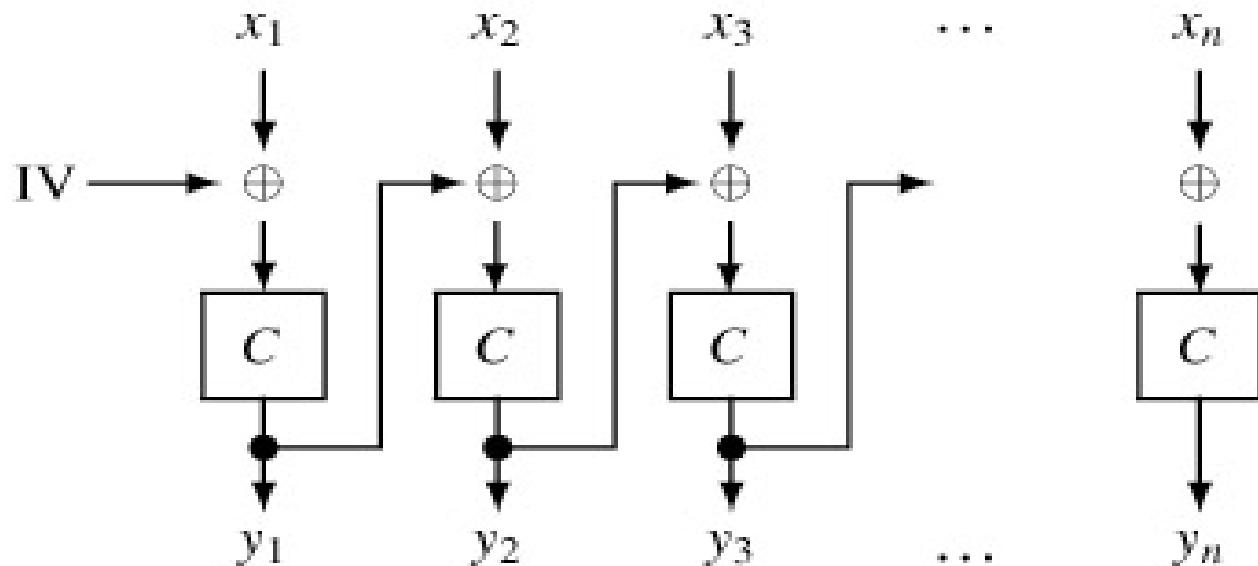


# Modos de operação da encriptação de blocos



## CBC (Cipher Block Chaining)

- IV - é o vetor de inicialização (Initial Vector)
- $Y[1] = C_k(X[1] \text{ XOR } IV)$
- $Y[i] = C_k(X[i] \text{ XOR } Y[i-1])$

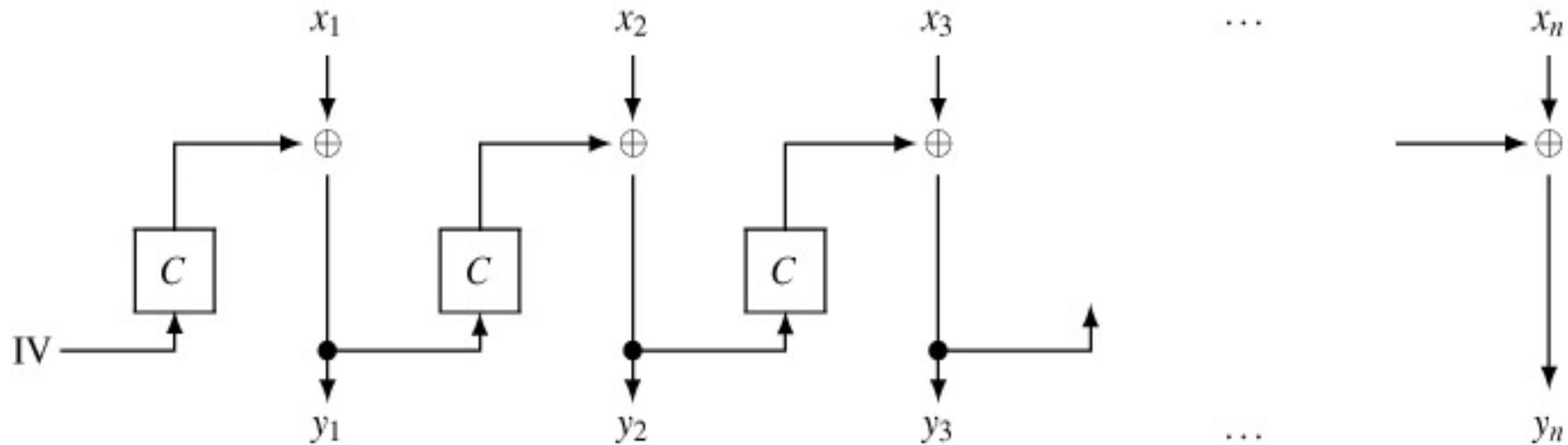


# Modos de operação da encriptação de blocos



## CFB (Cipher FeedBack)

- $Y[i] = X[i] \text{ XOR } C_k(Y[i-1])$
- $Y[1] = X[1] \text{ XOR } C_k(IV)$

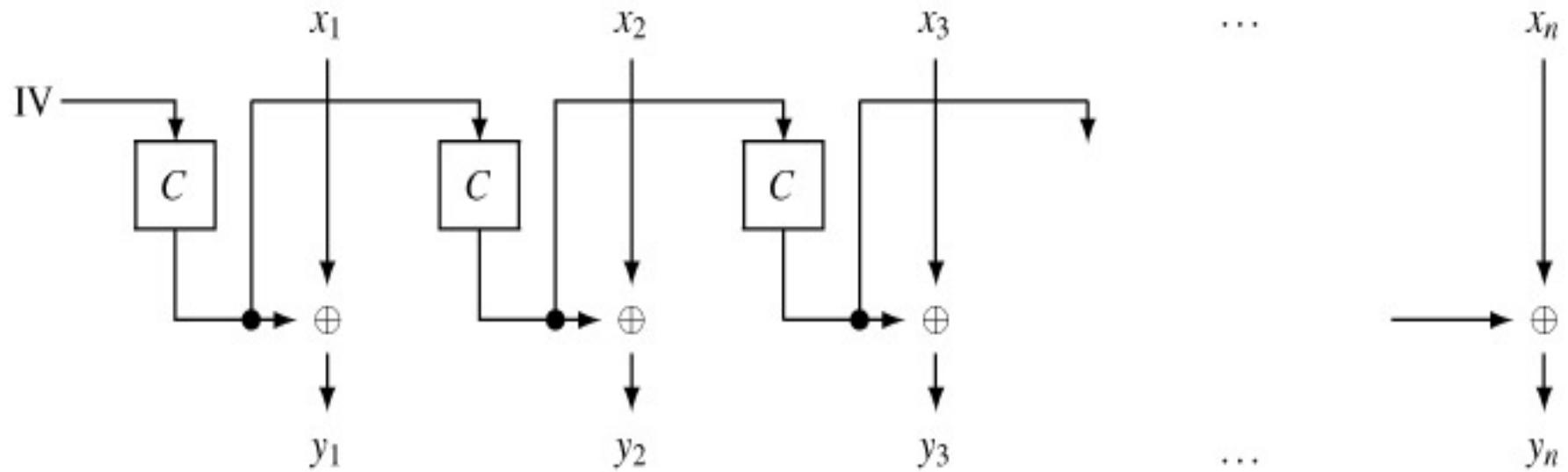


# Modos de operação da encriptação de blocos



## OFB (Output FeedBack)

- $Y[i] = X[i] \text{ XOR } O[i]$
- $O[i] = C_k(O[i-1])$
- $O[1] = C_k(IV)$

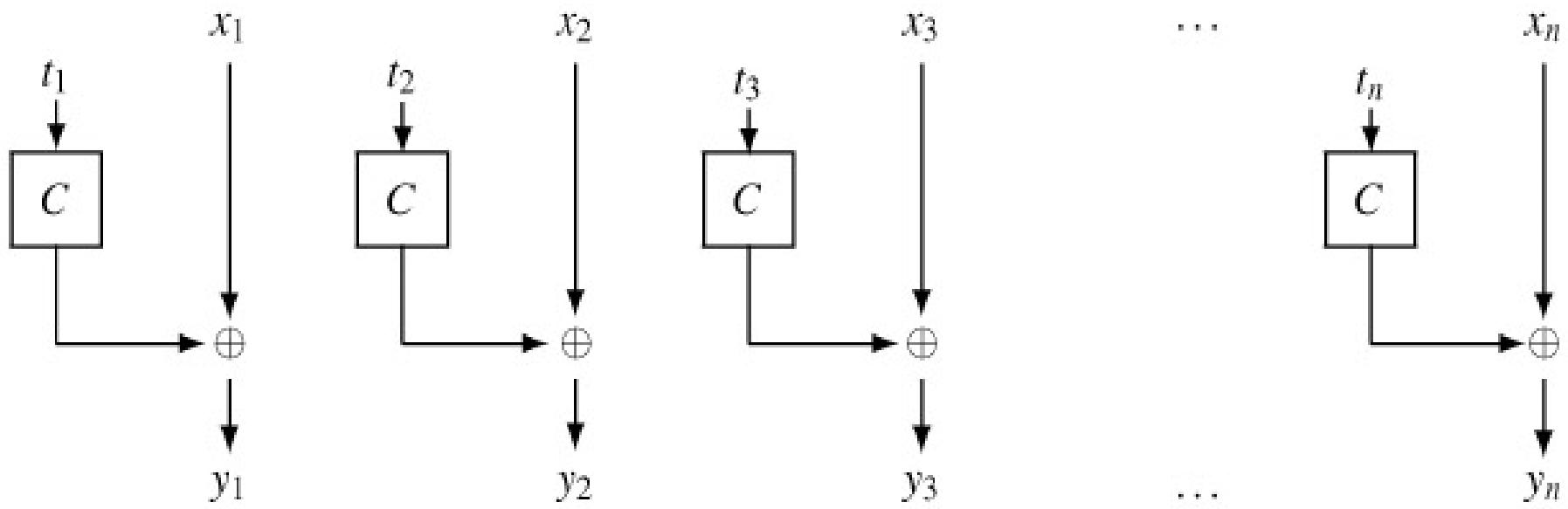


# Modos de operação da encriptação de blocos



## CTR (Counter Mode)

- $Y[i] = X[i] \text{ XOR } C_k(T[i])$
- $T = T[1], T[2], \dots, T[i]$  é um contador ou *timestamp*

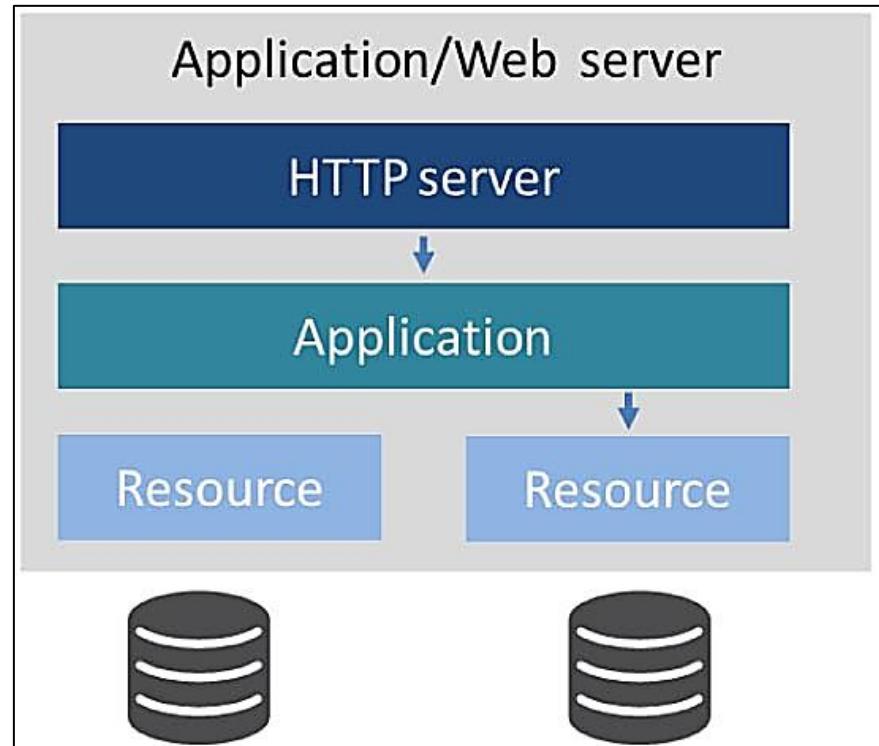


# Varredura de vulnerabilidades HTTPS

# Varredura de vulnerabilidades HTTPS



- Quem é responsável pela segurança do servidor web?
  - É uma questão de infraestrutura?
  - É uma questão de desenvolvimento?
- A figura mostra uma visão “invertida” da arquitetura de alto nível de uma aplicação: a camada superior é o servidor web
- A superfície de ataque da aplicação começa por aquilo que está mais exposto: o servidor web (a.k.a. HTTP Server)



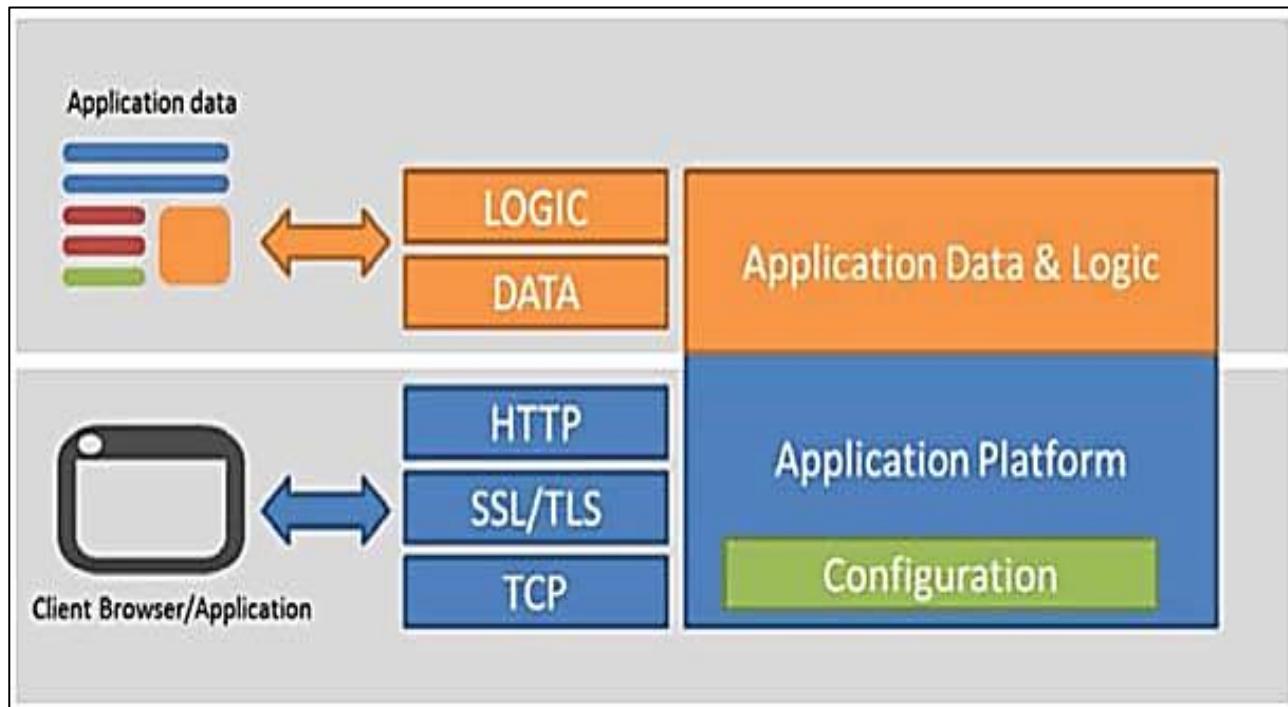
Fonte: Web Application Security is a Stack: How to CYA (Cover Your Apps) Completely by Lori Mac Vittie

# Varredura de vulnerabilidades HTTPS



As superfícies de ataque estão relacionadas aos clientes da aplicação

- Dados da aplicação
- Navegador web



Fonte: Web Application Security is a Stack: How to CYA (Cover Your Apps) Completely by Lori Mac Vittie

# Varredura de vulnerabilidades HTTPS



Pergunta:

- Existem fraquezas ou vulnerabilidades do HTTP que afetam as aplicações ?

Resposta:

- Sim!
- Busca por “HTTP” no CWE

<https://cwe.mitre.org/find/index.html>

The screenshot shows a web browser window with the address bar containing "cwe.mitre.org/find/index.html". The page title is "CWE - Search the CWE Web Site". The main content area displays search results for the keyword "http". The results list several CWE entries, each with a link to its definition page. The results are as follows:

- CWE-444: Inconsistent Interpretation of HTTP Requests ... - CWE  
<https://cwe.mitre.org/data/definitions/444.html>  
25 Jun 2020 ... CWE-444: Inconsistent Interpretation of HTTP Requests ('HTTP Request Smuggling'). Weakness ID: 444. Abstraction: Base Structure. Simple.
- CWE-79: Improper Neutralization of Input During Web Page ... - CWE  
<https://cwe.mitre.org/data/definitions/79.html>  
There are three main kinds of XSS: Type 1: Reflected XSS (or Non-Persistent) - The server reads data directly from the HTTP request and reflects it back in the ...
- CWE-650: Trusting HTTP Permission Methods on the Server ... - CWE  
<https://cwe.mitre.org/data/definitions/650.html>  
25 Jun 2020 ... The HTTP GET method and some other methods are designed to retrieve resources and not to alter the state of the application or resources on ...
- Common Weakness Enumeration: CWE  
<https://cwe.mitre.org/>  
Common Weakness Enumeration (CWE) is a list of software and hardware weaknesses.
- CWE-352: Cross-Site Request Forgery (CSRF) (4.2) - CWE  
<https://cwe.mitre.org/data/definitions/352.html>  
Check the HTTP Referer header to see if the request originated from an expected page. This could break legitimate functionality, because users or proxies may ...
- CWE-22: Improper Limitation of a Pathname to a Restricted ... - CWE  
<https://cwe.mitre.org/data/definitions/22.html>  
20 Aug 2020 ... extract the filename from the Http header BufferedReader br = new BufferedReader(new InputStreamReader(request.getInputStream()));
- CWE-78: Improper Neutralization of Special Elements used ... - CWE  
<https://cwe.mitre.org/data/definitions/78.html>  
... method gets the input coordinates from a user through a HTTP request and executes a program local to the application server that performs the transformation.
- CWE-601: URL Redirection to Untrusted Site ('Open Redirect ... - CWE  
<https://cwe.mitre.org/data/definitions/601.html>

# Varredura de vulnerabilidades HTTPS



Mozilla Observatory [observatory.mozilla.org](https://observatory.mozilla.org) Anônima

The Mozilla Observatory has helped over 240,000 websites by teaching developers, system administrators, and security professionals how to configure their sites safely and securely.

### Scan your site

enter domain name here

Don't include my site in the public results  
 Force a rescan instead of returning cached results  
 Don't scan with third-party scanners

Mozilla Observatory [observatory.mozilla.org/statistics/](https://observatory.mozilla.org/statistics/) Anônima

## HTTP Observatory

Number of Improved Websites	307.024
Number of Successfully Completed Scans	20.750.478
Number of Attempted Scans	22.628.780
Number of Unique Domains	3.565.543
Percentage of Sites Passing	17,546%

### Number of Unique Sites per Grade

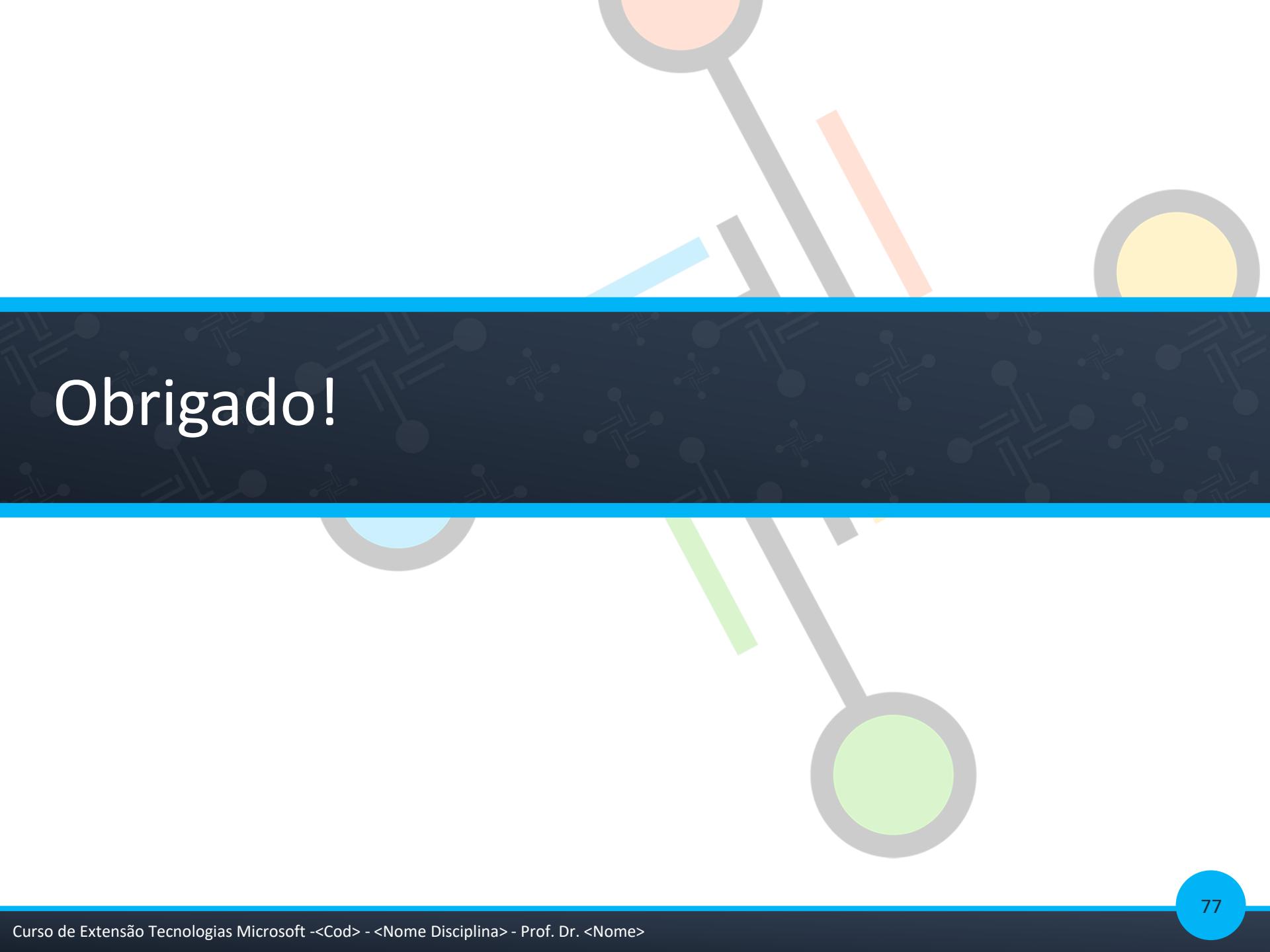
Grade	Number of Unique Sites
A+	~50,000
A	~5,000
A-	~5,000
B+	~40,000
B	~100,000
B-	~25,000
C+	~35,000
C	~65,000
C-	~25,000
D+	~85,000
D	~120,000
D-	~45,000

# Varredura de vulnerabilidades HTTPS



Qualys SSL Labs – SSL Server Test: <https://www.ssllabs.com/ssltest/>

The screenshot shows the Qualys SSL Server Test interface. At the top, there's a navigation bar with links for Home, Projects, Qualys Free Trial, and Contact. Below that, a breadcrumb trail indicates the user is at Home > Projects > SSL Server Test. The main title is "SSL Server Test". A note below it states: "This free online service performs a deep analysis of the configuration of any SSL web server on the public Internet. Please note that the information you submit here is used only to provide you the service. We don't use the domain names or the test results, and we never will." A form field labeled "Hostname:" contains a placeholder "Hostname:" and a "Submit" button. There's also a checkbox for "Do not show the results on the boards". Three panels below the form show recent activity: "Recently Seen" (listing sites like policy-read.mtsbu.ua, www.studionaut.com, etc.), "Recent Best" (listing sites with A+ grades like platzl.com, anonleaks.net, etc.), and "Recent Worst" (listing sites with failing grades like pta-demo.p1sec.fr, acsap1.kale logistics.com, etc.).



# Obrigado!