





Escola  
Superior  
de Redes  
RNP

# Teste de Invasão de Aplicações Web

## Capítulo 9

### Mecanismos criptográficos

- **Apresentar técnicas para identificar problemas de segurança na configuração de canais de comunicação e no uso de mecanismos criptográficos.**

- **Canal de comunicação com segurança mal configurada, suítes criptográficas, protocolos proprietários, BASE64, criptoanálise de cifras clássicas, índice de coincidência, chaves embutidas, chaves com baixa entropia, modo de operação inadequado, nível de segurança de criptossistemas, proteção de senhas.**

- **Introdução**
- **Vulnerabilidades no transporte de informações**
- **Contramedidas**
- **Vulnerabilidades no armazenamento de informações**
- **Contramedidas**

**A criptografia moderna está presente no nosso cotidiano de maneira profusa, em protocolos de comunicação, caixas eletrônicos, telefonia celular, proteção de software e de conteúdo, urnas eletrônicas e TV digital, dentre inúmeros outros exemplos.**

**Em sistemas de informação, a proteção criptográfica compreende a última barreira para evitar que o sigilo das informações sensíveis seja violado.**

**Infelizmente, porém, a implantação de mecanismos criptográficos requer diversos cuidados, que, na grande maioria das vezes, não são observados.**

**No mínimo, os seguintes aspectos devem ser considerados:**

**Utilização de algoritmos e protocolos conhecidos e extensivamente analisados**

**Uso de primitivas criptográficas adequadas para cada situação;**

**Emprego de bibliotecas que contenham implementações corretas dos criptossistemas adotados**

**Gerenciamento das chaves criptográficas**

**Comunicação segura de rede ocorre quando os seguintes requisitos de segurança da informação são satisfeitos:**

**Autenticação de  
entidades**

**Integridade**

**Autenticação da origem  
da mensagem**

**Confidencialidade**



**Em aplicações web, esses requisitos são atendidos, normalmente, pelo uso dos protocolos SSL e TLS, para transporte de dados HTTP.**

**Infelizmente, na prática, servidores não são corretamente configurados, do ponto de vista de segurança.**

**Inúmeros são os casos de aplicações que empregam certificados auto-assinados, expirados ou emitidos por autoridades certificadoras caseiras, das quais os navegadores e sistemas clientes não possuem a chave pública autêntica, para validação de assinatura.**

**Outro problema resultante de má configuração é o suporte a suítes criptográficas fracas e a versões antigas do protocolo SSL.**

**Um ponto que nunca recebe a atenção que merece é a proteção da chave privada utilizada por esses protocolos.**

**Outro exemplo de vulnerabilidade no lado do servidor compreende aplicações que protegem o transporte de informações sigilosas, por meio do protocolo HTTPS, e, não obstante, permitem que o mesmo recurso seja acessado, também, por meio de HTTP simples.**

**Do lado cliente, os problemas surgem quando as informações do certificado digital apresentado pelo servidor não são validadas, ou quando o protocolo não é seguido à risca.**

# Versão vulnerável de SSL

- ▶ O protocolo SSL apresenta diversas vulnerabilidades e, assim, não deve ser mais utilizado em ambientes de produção, e sim TLS v1.3.

- ▶ **Histórico:**

SSL 1.0 – nunca divulgado publicamente devido a questões de segurança.

SSL 2.0 – lançado em 1995. Depreciado em 2011. Conheceu problemas de segurança.

SSL 3.0 – lançado em 1996. Depreciado em 2015. Conheceu problemas de segurança.

TLS 1.0 – lançado em 1999 como uma atualização para SSL 3.0. Depreciação planejada para 2020.

TLS 1.1 – lançado em 2006. Depreciação planejada para 2020.

TLS 1.2 – lançado em 2008.

TLS 1.3 – lançado em 2018.

## Versão vulnerável de SSL

- ▶ O utilitário OpenSSL permite testar se um servidor está configurado para aceitar uma determinada versão de SSL/TLS, por meio das opções `-ssl2`, `-ssl3`, `-tls1`, `-tls1_1`, `-tls1_2`, `-tls1_3` :

```
~$ openssl s_client -ssl2 -connect ex.esr.rnp.br:443
...
SSL handshake has read 1416 bytes and written 364 bytes
---
New, SSLv2, Cipher is DES-CBC3-MD5
Server public key is 2048 bit
SSL-Session:
    Protocol    : SSLv2
    Cipher      : DES-CBC3-MD5
```

**Uma suíte criptográfica,  
no contexto dos  
protocolos SSL e TLS,  
descreve um conjunto  
de algoritmos que  
devem ser empregados  
na proteção do canal de  
comunicação.**

**Algumas delas,  
entretanto, utilizam  
criptossistemas fracos,  
que fornecem pouca ou  
nenhuma segurança.**

**O seguinte cenário ilustra um ataque bem sucedido contra esta vulnerabilidade:**

**1**

Por meio de outra vulnerabilidade no cliente, o atacante força que a lista de suítes enviadas na mensagem “client\_hello” contenha apenas a “NULL-SHA”.

**2**

O servidor escolhe os algoritmos para proteção do túnel SSL, com base na intersecção das listas de suítes suportadas por ambos que, no caso, é a própria “NULL-SHA”.

**3**

O atacante captura os pacotes de rede e extrai as informações desejadas que, embora encapsuladas por SSL, não estão cifradas.

# Suporte a suítes criptográficas fracas

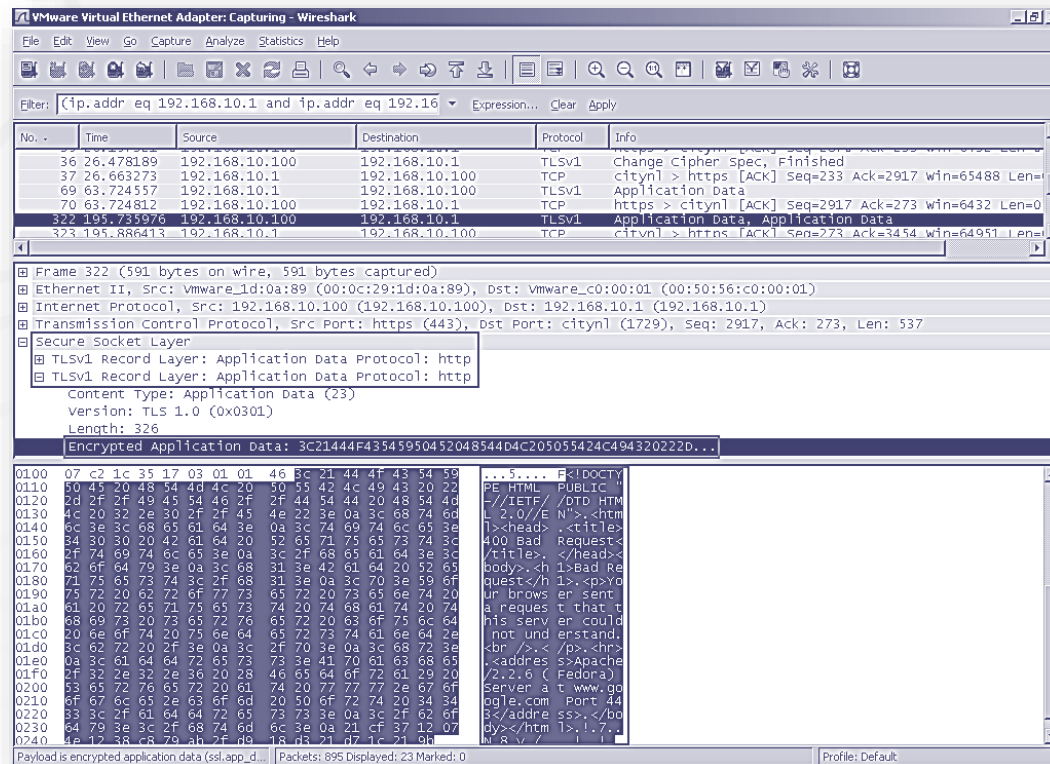


Figura 9.2 - Tráfego TLS em claro.



# Suporte a suítes criptográficas fracas

- Existem diversas ferramentas que podem ser utilizadas para identificar as suítes criptográficas suportadas por um servidor web.
- A primeira delas é o próprio OpenSSL, com a opção “-cipher”:  

```
~$ openssl s_client -connect gmail.google.com:443 -  
cipher NULL-SHA
```
- Outra ferramenta é a SSL Scan, (Titania, DinoTool, rbsec sslscan2)  

```
~$ sslscan gmail.google.com
```
- Também existe a ferramenta SSL Server Test, da Qualys SSL Labs =>  
<https://www.ssllabs.com/ssltest/>

```
~$ sslscan w3s.esr.rnp.br
```

```
...
```

```
Testing SSL server w3s.esr.rnp.br on port 443
```

## Supported Server Cipher(s) :

Accepted	SSLv2	168 bits	DES-CBC3-MD5
Accepted	SSLv2	56 bits	DES-CBC-MD5
Accepted	SSLv2	40 bits	EXP-RC2-CBC-MD5
Accepted	SSLv2	128 bits	RC2-CBC-MD5
Accepted	SSLv2	40 bits	EXP-RC4-MD5
Accepted	SSLv2	128 bits	RC4-MD5
Accepted	SSLv3	256 bits	ADH-AES256-SHA
Accepted	SSLv3	256 bits	DHE-RSA-AES256-SHA
Rejected	SSLv3	256 bits	DHE-DSS-AES256-SHA

**Para verificar se há algum problema com o certificado digital utilizado por um servidor, basta acessar com um navegador web qualquer página da aplicação servida por HTTPS.**

**Situações relacionadas ao certificado instalado, que impedem que a negociação SSL/TLS seja realizada com sucesso incluem:**

**Certificado expirado**

**Certificado válido a partir de data posterior à atual**

**Certificado auto-assinado**

**Certificado emitido por autoridade certificadora desconhecida pelo navegador**

**Situações relacionadas ao certificado instalado, que impedem que a negociação SSL/TLS seja realizada com sucesso incluem:**

**Certificado revogado**

**Assinaturas digitais inválidas na cadeia de certificação**

**Nome de domínio do sítio web não incluso no(s) contido(s) no certificado**

**Este problema decorre de um cliente web que não verifica se o nome de domínio da aplicação sendo acessada é o mesmo que o contido no certificado digital apresentado pelo servidor.**

**Quando este passo não é executado, a conexão pode ser estabelecida com um servidor falsificado, caso seja possível redirecionar o usuário.**

**Somente clientes proprietários devem ter este aspecto verificado em um teste de invasão.**

1

O atacante obtém um certificado digital e chave privada correspondente válidos, de um domínio XYZ qualquer.

2

Um servidor web é criado com conteúdo clonado do domínio ABC e com HTTPS configurado com o par de chaves do primeiro passo.

3

Um e-mail é enviado a um conjunto de vítimas para que acessem o servidor malicioso, como sendo do domínio ABC.

4

Durante a negociação SSL, o servidor clonado envia o certificado do domínio XYZ para o navegador, que não valida o domínio. Como nenhum erro ocorre, a chave pública é extraída e utilizada para compor a mensagem “client\_key\_exchange”, enviada, em seguida, ao servidor.

5

O servidor é capaz de extrair o conteúdo da mensagem “client\_key\_exchange”, pois possui a chave privada associada ao certificado do domínio XYZ, e completar as operações para estabelecimento de chaves.

6

As mensagens “change\_cipher\_spec” e “finished” são trocadas entre as duas partes e o protocolo encerra-se normalmente.



**Projetar protocolos seguros é uma tarefa tão difícil quanto criar bons algoritmos criptográficos. Assim, a chance de um protocolo caseiro ser livre de vulnerabilidades é extremamente baixa, ainda mais quando ele é a criação de uma equipe não especializada.**

**Situações relacionadas ao certificado instalado, que impedem que a negociação SSL/TLS seja realizada com sucesso incluem:**

**Personificação**

**Intercalação**

**Espera  
forçada**

**Repetição**

**Reflexão**

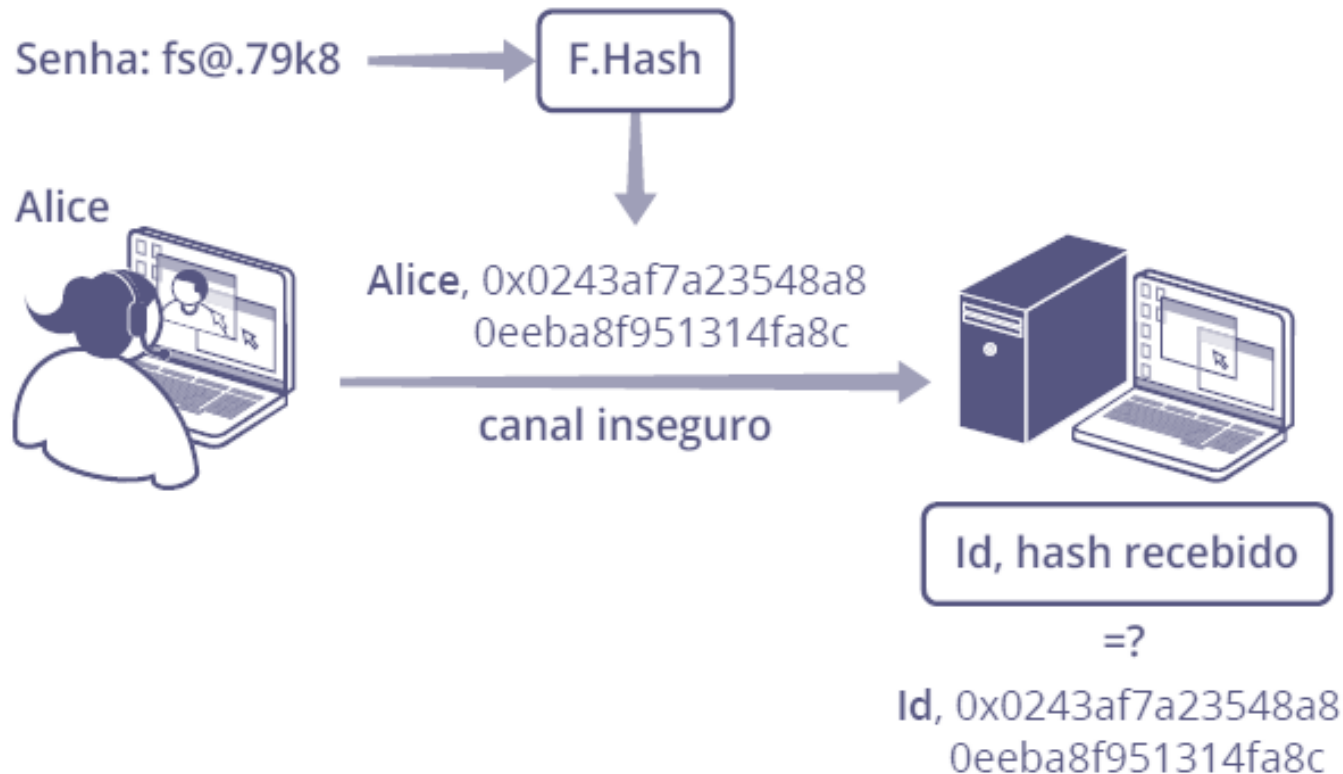


Figura 9.3 - Protocolo de autenticação vulnerável.

**Os seguintes pontos devem ser observados para evitar-se uma comunicação insegura com a aplicação web:**

- **Não utilize protocolos de segurança desenvolvidos caseiramente.**

- **Configure o servidor para aceitar apenas suítes criptográficas fortes.**

- **Não utilize versões de SSL anteriores a 3.0 e prefira o uso do protocolo TLS.**

**Os seguintes pontos devem ser observados para evitar-se uma comunicação insegura com a aplicação web:**

**Compre e instale um certificado digital de uma autoridade certificadora conhecida e confiável. Não deixe que o certificado expire, adquirindo um novo, antes que isso aconteça.**

**Para aplicações internas, caso uma infra-estrutura de chaves públicas própria seja utilizada, instale o certificado raiz correspondente nos navegadores web dos usuários.**

**Não permita que um recurso servido por meio do protocolo HTTPS também seja acessível por HTTP.**

**Os seguintes pontos devem ser observados para evitar-se uma comunicação insegura com a aplicação web:**

- **Proteja a chave privada do servidor, preferencialmente, partilhando-a entre vários custodiantes.**

- **Caso implemente o lado cliente dos protocolos SSL/TLS, lembre-se sempre de verificar a validade do certificado recebido e conferir o nome de domínio acessado contra o contido no certificado.**

**É relativamente comum, encontrar sistemas que apenas se preocupam em proteger as informações em trânsito e que se esquecem (ou ignoram) de protegê-las durante o armazenamento.**

**Por outro lado, quando existe a preocupação em cifrar dados sigilosos, a maior vulnerabilidade encontrada é com relação à proteção das chaves criptográficas utilizadas. Normalmente, os desenvolvedores as embutem no código, achando que, uma vez compilados os programas, será difícil que alguém as recupere.**

**Pensando no ciclo de vida das chaves criptográficas, um ponto que merece, mas não recebe, bastante atenção é a fase de criação, que deve empregar métodos que garantam um bom nível de aleatoriedade.**

**Muitas empresas utilizam cifras caseiras ou clássicas na proteção de informações valiosas.**

**Aspectos mais sutis, com relação ao armazenamento seguro de informações, estão relacionados a detalhes de desenvolvimento da solução.**



**BASE64 é o nome dado a um grupo de esquemas de codificação, que representam cadeias de octetos como caracteres imprimíveis, pertencentes a um conjunto de 65 caracteres ASCII.**

**A entrada é processada da esquerda para a direita, três octetos por vez, os quais resultam em quatro caracteres codificados.**

**Um ponto importante, que deve ser notado, é que os processos de conversão não dependem de nenhum segredo e, assim, qualquer pessoa é capaz de executá-los.**

**Caso o tamanho da mensagem não seja múltiplo de três, o último bloco conterá um ou dois octetos e deverá ser tratado de maneira especial.**

**Isto implica que o mecanismo  
não deve nunca ser utilizado  
para proteção do sigilo de  
informações.**

Valor	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Código	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P

Valor	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Código	Q	R	S	T	U	V	W	X	Y	Z	a	b	c	d	e	f

Valor	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
Código	G	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v

Valor	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
Código	W	x	y	z	0	1	2	3	4	5	6	7	8	9	+	/

Figura 9.4 - mapeamento utilizado em BASE64.

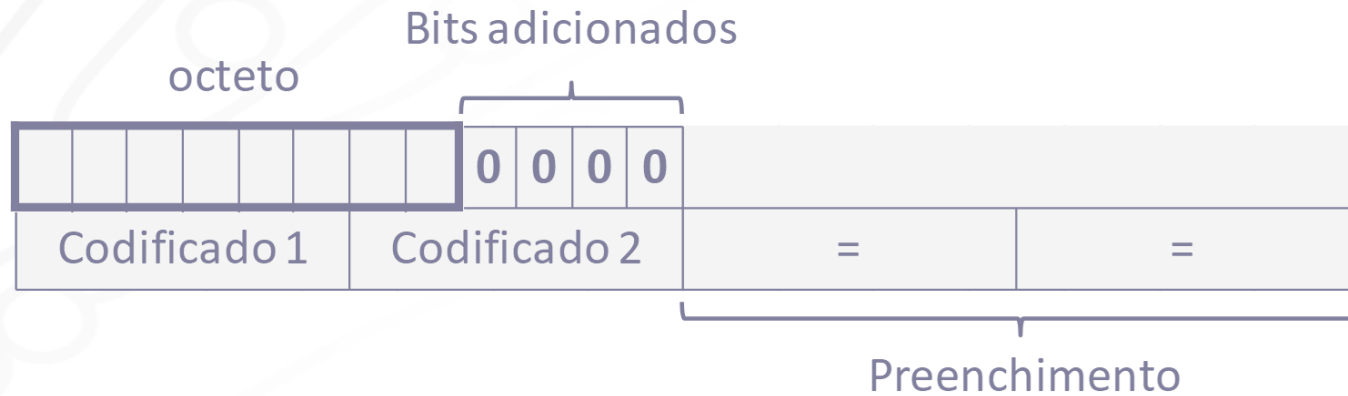


Figura 9.5 - Codificação em BASE64, quando o último bloco possui um octeto.



Figura 9.6 - Codificação em BASE64, quando o último bloco possui dois octetos.

T		e		s		t		e			
010101 00		0110 0101		01 110011		011101 00		0110 0101		0	0
010101	000110	010101	110011	011101	000110	010100					
V	G	V	z	d	G	U		=			

Figura 9.7 - Exemplo de codificação em BASE64.



## Exercício de Fixação 2

### BASE64

---

1. É seguro utilizar BASE64 para proteção de informações sensíveis? Justifique sua resposta.



**Infelizmente, ainda hoje, encontram-se sistemas que utilizam cifras clássicas na proteção de informações sensíveis.**

**Esses algoritmos criptográficos históricos são quebrados facilmente, mesmo sem a ajuda de computadores, o que torna o fato bem preocupante.**

**Cifras de substituição simples e de transposição são duas classes básicas, mas importantes, de cifras simétricas.**

**As primeiras, também chamadas de cifras monoalfabéticas, definem um mapeamento entre o alfabeto em claro para o de ciframento, que é utilizado na substituição de cada elemento do texto original por outro.**

# Conceitos adicionais sobre cifras

Alfabeto original																									
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
↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕
⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	⌘	①	②	③	④	⑤	⑥	⑦	⑧	⑨	⑩	⑪	⑫	⑬	⑭	⑮	⑯	⑰
Alfabeto de ciframento																									
Texto em claro:		e	l	e	m	e	n	t	o																
		↓	↓	↓	↓	↓	↓	↓	↓																
Texto cifrado:		⌘	③	⌘	④	⌘	⑤	⑪	⑮																

Figura 9.8 - Exemplo de cifra de substituição simples.

**As cifras de transposição operam sobre blocos de tamanho fixo pré-estabelecido, permutando os elementos de cada um deles, segundo uma regra definida.**

**Uma terceira classe de cifras, a de substituição polialfabética, trabalha com um conjunto de  $n$  mapeamentos de substituição, que são aplicados ordenada e ciclicamente aos caracteres do texto em claro.**

**Cifras de substituição homofônica mapeiam para cada caractere do alfabeto original um conjunto de  $t$  símbolos, com a restrição de não haver elementos comuns entre os diversos conjuntos.**

**Por convenção, quando cifras clássicas são utilizadas, o texto em claro é representado em letras minúsculas e o cifrado em letras maiúsculas. Além disso, o alfabeto original e o de ciframento são idênticos.**

**Finalmente, assume-se que o atacante sabe a língua da mensagem original e a cifra que foi utilizada para protegê-la.**

**O primeiro passo antes de atacar um texto cifrado é identificar o tipo de algoritmo que foi utilizado, pois os métodos de criptoanálise diferem para cada classe.**

**Especificamente para os criptossistemas históricos, o objetivo é determinar se foram empregadas cifras monoalfabéticas, polialfabéticas ou de transposição.**

O índice de coincidência, introduzido por William Friedman em 1922, é uma ferramenta estatística que pode ser utilizada para este propósito.

$$IC(n) = \sum_{\alpha \in A} p_{\alpha} = \sum_{\alpha \in A} \frac{n_{\alpha} \times (n_{\alpha} - 1)}{n \times (n - 1)}$$

Tipo de cifra	Mono - português	Mono - inglês	Poli ou homofônica
IC	0,0788	0,0657	0,03846



## Alfabeto original

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
↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕
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	C

<b>Texto em claro:</b>	e	l	e	m	e	n	t	o
	↓	↓	↓	↓	↓	↓	↓	↓
<b>Texto cifrado:</b>	H	O	H	P	H	Q	W	R

Figura 9.10 - Mapeamento da cifra de Cesar e exemplo.

# Cifra de deslocamento e ROT13

## Alfabeto original

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
↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕
K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J

<b>Texto em claro:</b>	e	l	e	m	e	n	t	o
	↓	↓	↓	↓	↓	↓	↓	↓
<b>Texto cifrado:</b>	O	V	O	W	O	X	D	Y

Figura 9.11 - Exemplo de cifra de deslocamento com chave  $k = 10$ .

# Cifra de deslocamento e ROT13

k	Mapeamento																										Texto candidato
0	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	YKMAXGTIG
1	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	xjlzwfshf
2	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	wikyverge
3	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	C	vhjxudqfd
4	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	ugiwtcpec
5	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	tfhvsbodb
6	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	seguranca

Figura 9.12 - Exemplo de criptoanálise da cifra de deslocamento.

**Uma cifra de substituição monoalfabética genérica permite que o mapeamento seja realizado para qualquer permutação do alfabeto de entrada. Com isto, o total de chaves sobe para 26!.**

**Uma vez que o espaço de chaves deste esquema tem tamanho razoável para os dias atuais, resta saber se o algoritmo também é resistente.**

**O primeiro fato que deve ser levado em conta é que cifras de substituição simples apenas transferem as frequências individuais dos símbolos do texto em claro para outros caracteres no texto cifrado.**

**O segundo aspecto importante é que toda linguagem natural é redundante, o que determina uma estrutura característica de frequências e agrupamentos de letras.**

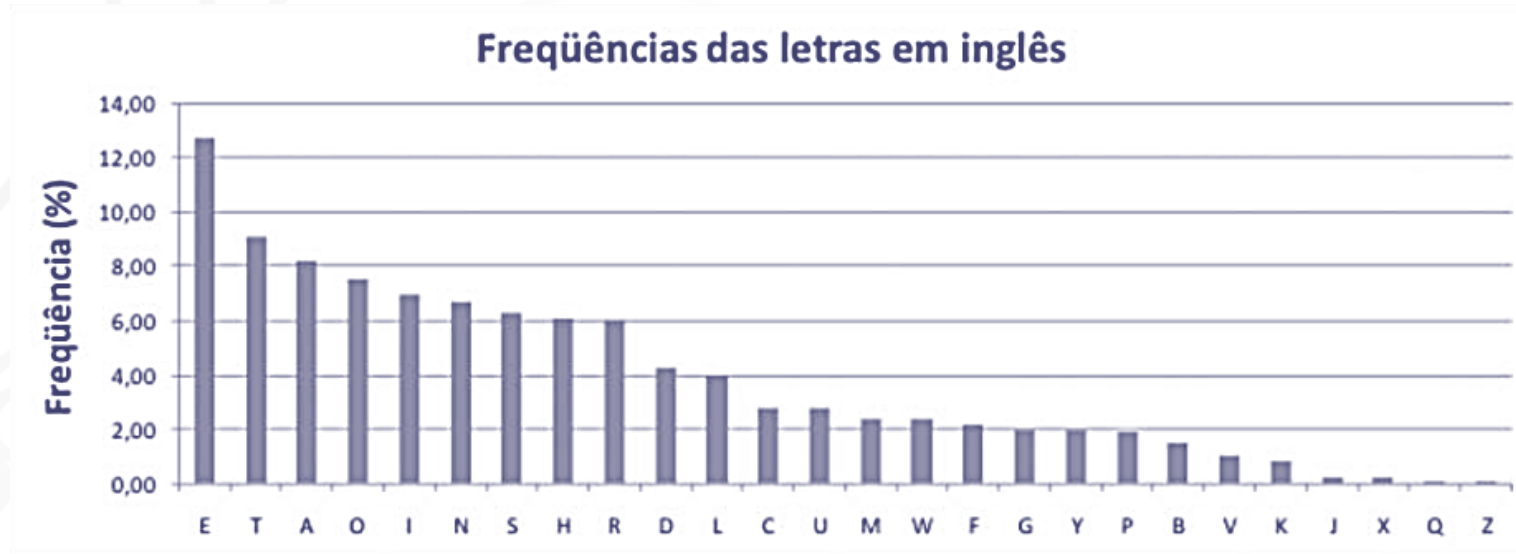


Figura 9.13 - Frequências das letras na língua inglesa.

“Some 28,000 years ago in what is now the British territory of Gibraltar, a group of Neandertals eked out a living along the rocky Mediterranean coast. They were quite possibly the last of their kind. Elsewhere in Europe and western Asia, Neandertals had disappeared thousands of years earlier, after having ruled for more than 200,000 years. The Iberian Peninsula, with its comparatively mild climate and rich array of animals and plants, seems to have been the final stronghold. Soon, however, the Gibraltar population, too, would die out, leaving behind only a smattering of their stone tools and the charred remnants of their campfires.”, (Wong, 2009).



Figura 9.15 - Comparação das frequências das letras em inglês e em um trecho de artigo.



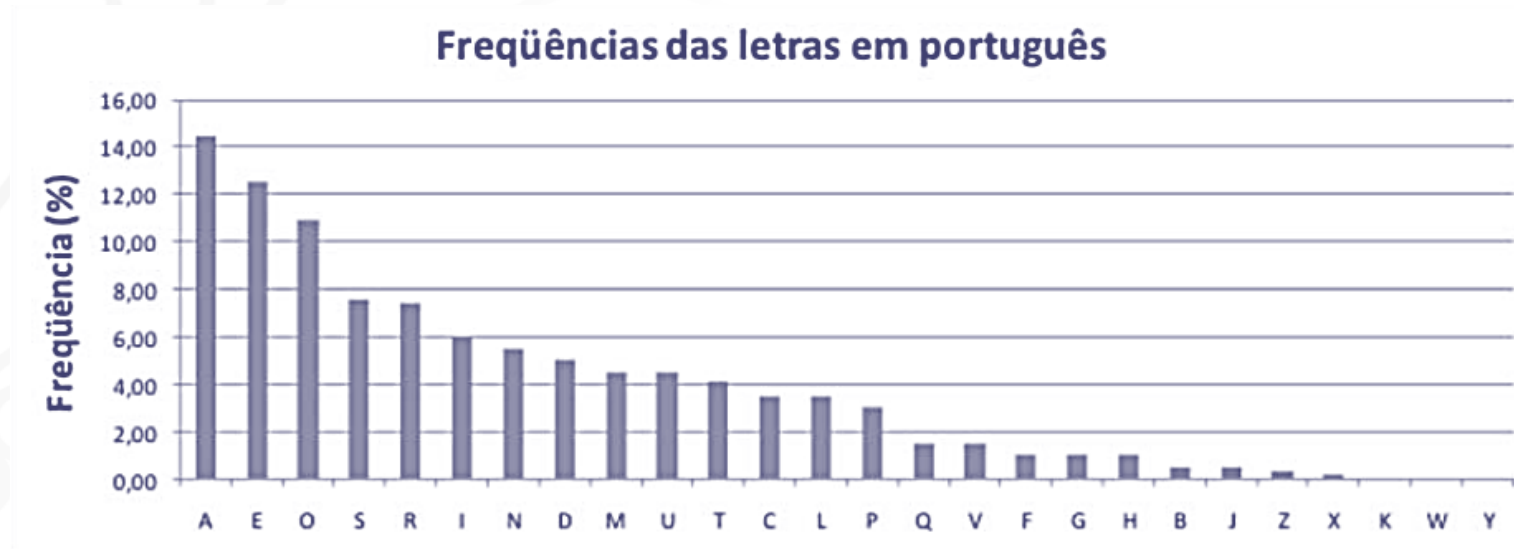


Figura 9.14 - Frequências das letras na língua portuguesa.

**O método conhecido por análise de frequências, introduzido pelo polímata árabe al-Kindī, emprega os conceitos recém apresentados, para quebrar cifras de substituição simples.**

**O primeiro passo consiste em descobrir o idioma da mensagem original, o que é fundamental para saber quais são as frequências esperadas de cada letra do alfabeto.**

**Em seguida, deve-se realizar a contagem de cada símbolo presente no texto cifrado sendo analisado.**

**Como cifras monoalfabéticas apenas alteram a “face” de cada símbolo, mantendo a distribuição geral de frequências, é razoável assumir que cada elemento do texto cifrado corresponde a uma letra similarmente frequente do alfabeto original.**

DSFY CQLGRBGJOFUT XJTG SJY IQBG YQXSFYDFCJDGL GAGIGUDY JD DSG RQDDQI  
QP DSG XJTG PQB GZJIXAG MQW CJU JAYQ GZJIFUG DSG PBGHWGUCM QP XJFBGL  
AGDDGBY JUL CQIXJBG DSGI DQ DSG PBGHWGUCM QP XJFBGL AGDDGBY FU  
GUTAFYS QB MQW CJU AQQO JD BGXGJDGL AGDDGBY QB DSG EQVGA DBQVGA

**IC = 0,0723 → Espanhol??**

Figura 9.19 - Texto cifrado com cifra monoalfabética.

# Cifra de substituição monoalfabética




Figura 9.20 - Frequências individuais das letras do texto cifrado comparadas com as da língua inglesa.

t\*\* \*a\*e\*\*e\*\*\*\*\* \*\*\*e \*\*\* \*a\*e \*a\*\*\*\*\*t\*\*\*\*te\* e\*e\*e\*t\* \*t  
t\*e \*atta\* a\* t\*e \*\*\*e \*a\* e\*\*\*\*\*e \*a\* \*\*\* \*\*\*a e\*\*\*\*\*e  
t\*e \*\*e\*\*e\*\*\* a\* \*\*\*\*\*e \*ette\*\* \*\*\* \*a\*\*\*\*\*e t\*e\* ta t\*e  
\*\*e\*\*e\*\*\* a\* \*\*\*\*\*e \*ette\*\* \*\* e\*\*\*\*\* a\* \*a\* \*\*\* \*aa\* \*t  
\*e\*e\*te\* \*ette\*\* a\* t\*e \*a\*e\* t\*a\*e\*

Figura 9.21 - Análise de frequências (1/11) – substituição de “G”, “D” e “Q” por “e”, “t” e “a”, respectivamente.

# Cifra de substituição monoalfabética

			t			e												h									
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		

th\*\* \*a\*e\*\*e\*\*\*\*\* \*\*\*e h\*\* \*a\*e \*a\*h\*\*t\*\*\*te\* e\*e\*e\*t\* \*t  
the \*atta\* a\* the \*\*\*e \*a\* e\*\*\*\*\*e \*a\* \*\*\* \*\*\*a e\*\*\*\*\*e  
the \*\*e\*\*e\*\*\* a\* \*\*\*\*\*e\* \*ette\*\* \*\*\* \*a\*\*\*\*\*e the\* ta the  
\*\*e\*\*e\*\*\* a\* \*\*\*\*\*e\* \*ette\*\* \*\* e\*\*\*\*\*h a\* \*a\* \*\*\* \*aa\* \*t  
\*e\*e\*te\* \*ette\*\* a\* the \*a\*e\* t\*a\*e\*

DSFY CQLGRBGJOFUT XJTG SJY IQBG YQXSFYDFCJDGL GAGIGUDY JD  
DSG RQDDQI QP DSG XJTG PQB GZJIXAG MQW CJU JAYQ GZJIFUG  
DSG PBGHWGUCM QP XJFBGL AGDDGBY JUL CQIXJBG DSGI DQ DSG  
PBGHWGUCM QP XJFBGL AGDDGBY FU GUTAFYS QB MQW CJU AQOQ JD  
BGXGJDGL AGDDGBY QB DSG EQVGA DBQVGA

Figura 9.22 - Análise de frequências (2/11); substituição de “S” por “h”.

# Cifra de substituição monoalfabética

			t			e												h										
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			

th\*\* \*\*\*e\*\*e\*\*\*\*\* \*\*\*e h\*\* \*\*\*e \*\*\*h\*\*t\*\*\*te\* e\*e\*e\*t\* \*t  
the \*\*tt\*\* \*\* the \*\*\*e \*\*\* e\*\*\*\*\*e \*\*\* \*\*\* e\*\*\*\*\*e  
the \*\*e\*\*e\*\*\* \*\* \*\*\*\*\*e\* \*ette\*\* \*\*\* \*\*\*\*\*e the\* t\* the  
\*\*e\*\*e\*\*\* \*\* \*\*\*\*\*e\* \*ette\*\* \*\* e\*\*\*\*\*h \*\* \*\*\* \*\*\* \*\*\*\*\* \*t  
\*e\*e\*te\* \*ette\*\* \*\* the \*\*\*e\* t\*\*\*e\*

DSFY CQLGRBGJOFUT XJTG SJY IQBG YQXSFYDFCJDGL GAGIGUDY JD  
DSG RQDDQI QP DSG XJTG PQB GZJIXAG MQW CJU JAYQ GZJIFUG  
DSG PBGHWGUCM QP XJFBGL AGDDGBY JUL CQIXJBG DSGI DQ DSG  
PBGHWGUCM QP XJFBGL AGDDGBY FU GUTAFYS QB MQW CJU AQOQ JD  
BGXGJDGL AGDDGBY QB DSG EQVGA DBQVGA

Figura 9.23 - Análise de frequências (3/11); cancelamento da substituição de “Q” por “a”.



# Cifra de substituição monoalfabética

			t			e										o		h									
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		

th\*\* \*o\*e\*\*e\*\*\*\*\* \*\*\*e h\*\* \*o\*e \*o\*h\*\*t\*\*\*te\* e\*e\*e\*t\* \*t  
the \*otto\* o\* the \*\*\*e \*o\* e\*\*\*\*\*e \*o\* \*\*\* \*\*\*o e\*\*\*\*\*e  
the \*\*e\*\*e\*\*\* o\* \*\*\*\*\*e \*ette\*\* \*\*\* \*o\*\*\*\*\*e the\* to the  
\*\*e\*\*e\*\*\* o\* \*\*\*\*\*e \*ette\*\* \*\* e\*\*\*\*\*h o\* \*o\* \*\*\* \*oo\* \*t  
\*e\*e\*te\* \*ette\*\* o\* the \*o\*e\* t\*o\*e\*

DSFY CQLGRBGJOFUT XJTG SJY IQBG YQXSFYDFCJDGL GAGIGUDY JD  
DSG RQDDQI QP DSG XJTG PQB GZJIXAG MQW CJU JAYQ GZJIFUG  
DSG PBGHWGUCM QP XJFBGL AGDDGBY JUL CQIXJBG DSGI DQ DSG  
PBGHWGUCM QP XJFBGL AGDDGBY FU GUTAFYS QB MQW CJU AQOQ JD  
BGXGJDGL AGDDGBY QB DSG EQVGA DBQVGA

Figura 9.24 - Análise de frequências (4/11) – substituição de “Q” por “o”.

# Cifra de substituição monoalfabética

	r		t			e									f	o		h									
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		

th\*\* \*o\*e\*re\*\*\*\*\* \*\*\*e h\*\* \*ore \*o\*h\*\*t\*\*\*te\* e\*e\*e\*t\* \*t  
the \*otto\* of the \*\*\*e for e\*\*\*\*\*e \*o\* \*\*\* \*\*\*o e\*\*\*\*\*e  
the fre\*\*e\*\*\* of \*\*\*re\* \*etter\* \*\*\* \*o\*\*\*re the\* to the  
fre\*\*e\*\*\* of \*\*\*re\* \*etter\* \*\* e\*\*\*\*\*h or \*o\* \*\*\* \*oo\* \*t  
re\*e\*te\* \*etter\* or the \*o\*e\* tro\*e\*

DSFY CQLGRBGJOFUT XJTG SJY IQBG YQXSFYDFCJDGL GAGIGUDY JD  
DSG RQDDQI QP DSG XJTG PQB GZJIXAG MQW CJU JAYQ GZJIFUG  
DSG PBGHWGUCM QP XJFBGL AGDDGBY JUL CQIXJBG DSGI DQ DSG  
PBGHWGUCM QP XJFBGL AGDDGBY FU GUTAFYS QB MQW CJU AQOQ JD  
BGXGJDGL AGDDGBY QB DSG EQVGA DBQVGA

Figura 9.25 - Análise de frequências (5/11) - substituição de “P” por “f” e de “B” por “r”.

# Cifra de substituição monoalfabética

	r		t			e			a						f	o		h									
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		

th\*\* \*o\*e\*rea\*\*\*\* \*a\*e ha\* \*ore \*o\*h\*\*t\*\*ate\* e\*e\*e\*t\* at  
the \*otto\* of the \*a\*e for e\*a\*\*\*e \*o\* **\*a\*** a\*\*o e\*a\*\*\*e  
the fre\*\*e\*\*\* of \*a\*re\* \*etter\* **a\*\*** \*o\*\*are the\* to the  
fre\*\*e\*\*\* of \*a\*re\* \*etter\* \*\* e\*\*\*\*\*h or \*o\* \*a\* \*oo\* at  
re\*eate\* \*etter\* or the \*o\*e\* tro\*e\*

DSFY CQLGRBGJOFUT XJTG SJY IQBG YQXSFYDFCJDGL GAGIGUDY JD  
DSG RQDDQI QP DSG XJTG PQB GZJIXAG MQW **CJU** JAYQ GZJIFUG  
DSG PBGHWGUCM QP XJFBGL AGDDGBY **JUL** CQIXJBG DSGI DQ DSG  
PBGHWGUCM QP XJFBGL AGDDGBY FU GUTAFYS QB MQW CJU AQOQ JD  
BGXGJDGL AGDDGBY QB DSG EQVGA DBQVGA

Figura 9.26 - Análise de frequências (6/11) – substituição de “J” por “a”.

# Cifra de substituição monoalfabética

	r	c	t			e			a		d				f	o		h		n						
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	

th\*\* code\*rea\*\*n\* \*a\*e ha\* \*ore \*o\*h\*\*t\*cated e\*e\*ent\* at  
the \*otto\* of the \*a\*e for e\*a\*\*\*e \*o\* can a\*\*o e\*a\*\*ne  
the fre\*\*enc\* of \*a\*red \*etter\* and co\*\*are the\* to the  
fre\*\*enc\* of \*a\*red \*etter\* \*n en\*\*\*h or \*o\* can \*oo\* at  
re\*eated \*etter\* or the \*o\*e\* tro\*e\*

DSFY CQLGRBGJOFUT XJTG SJY IQBG YQXSFYDFCJDGL GAGIGUDY JD  
DSG RQDDQI QP DSG XJTG PQB GZJIXAG MQW CJU JAYQ GZJIFUG  
DSG PBGHWGUCM QP XJFBGL AGDDGBY JUL CQIXJBG DSGI DQ DSG  
PBGHWGUCM QP XJFBGL AGDDGBY FU GUTAFYS QB MQW CJU AQOQ JD  
BGXGJDGL AGDDGBY QB DSG EQVGA DBQVGA

Figura 9.27 - Análise de frequências (7/11) – substituição de “C”, “L” e “U” por “c”, “d” e “n”, respectivamente.

# Cifra de substituição monoalfabética

	r	c	t			e		m	a		d				f	o		h		n			p	s	
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

th\*s code\*rea\*\*n\* pa\*e has more soph\*st\*cated e\*ements at the \*ottom of the pa\*e for e\*amp\*e \*o\* can a\*so e\*am\*ne the fre\*\*enc\* of pa\*red \*etters and compare them to the fre\*\*enc\* of pa\*red \*etters \*n en\*\*\*sh or \*o\* can \*oo\* at repeated \*etters or the \*o\*e\* tro\*e\*

DSFY CQLGRBGJOFUT XJTG SJY IQBG YQXSFYDFCJDGL GAGIGUDY JD  
DSG RQDDQI QP DSG XJTG PQB GZJIXAG MQW CJU JAYQ GZJIFUG  
DSG PBGHWGUCM QP XJFBGL AGDDGBY JUL CQIXJBG DSGI DQ DSG  
PBGHWGUCM QP XJFBGL AGDDGBY FU GUTAFYS QB MQW CJU AQOQ JD  
BGXGJDGL AGDDGBY QB DSG EQVGA DBQVGA

Figura 9.28 - Análise de frequências (8/11) – substituição de “X”, “Y” e “I” por “p”, “s”, e “m”, respectivamente.

# Cifra de substituição monoalfabética

l	r	c	t		i	e		m	a		d				f	o		h	g	n			p	s	
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

this code\*rea\*ing page has more sophisticated elements at the \*ottom of the page for e\*ample \*o\* can also e\*amine the fre\*\*enc\* of paired letters and compare them to the fre\*\*enc\* of paired letters in english or \*o\* can loo\* at repeated letters or the \*o\*el tro\*el

DSFY CQLGRBGJOFUT XJTG SJY IQBG YQXSFYDFCJDGL GAGIGUDY JD  
DSG RQDDQI QP DSG XJTG PQB GZJIXAG MQW CJU JAYQ GZJIFUG  
DSG PBGHWGUCM QP XJFBGL AGDDGBY JUL CQIXJBG DSGI DQ DSG  
PBGHWGUCM QP XJFBGL AGDDGBY FU GUTAFYS QB MQW CJU AQOQ JD  
BGXGJDGL AGDDGBY QB DSG EQVGA DBQVGA

Figura 9.29 - Análise de frequências (9/11) – substituição de “F”, “T” e “A” por “i”, “g” e “l”, respectivamente.

# Cifra de substituição monoalfabética

l	r	c	t		i	e	q	m	a		d	y			f	o	b	h	g	n		u	p	s	x
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

this **codebrea\*ing** page has more sophisticated elements at the bottom of the page for example you can also examine the frequency of paired letters and compare them to the frequency of paired letters in english or you can **loo\*** at repeated letters or the **\*o\*el tro\*el**

DSFY **CQLGRBGJOFUT** XJTG SJY IQBG YQXSFYDFCJDGL GAGIGUDY JD  
DSG RQDDQI QP DSG XJTG PQB GZJIXAG MQW CJU JAYQ GZJIFUG  
DSG PBGHWGUCM QP XJFBGL AGDDGBY JUL CQIXJBG DSGI DQ DSG  
PBGHWGUCM QP XJFBGL AGDDGBY FU GUTAFYS QB MQW CJU **AQOQ** JD  
BGXGJDGL AGDDGBY QB DSG **EQVGA DBQVGA**

Figura 9.30 - Análise de frequências (10/11) – substituição de “R”, “Z”, “H”, “W” e “M” por “b”, “x”, “q”, “u” e “y”, respectivamente.

l	r	c	t	v	i	e	q	m	a		d	y		k	f	o	b	h	g	n	w	u	p	s	x
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

this codebreaking page has more sophisticated elements at the bottom of the page for example you can also examine the frequency of paired letters and compare them to the frequency of paired letters in english or you can look at repeated letters or the vowel trowel

Figura 9.31 - Análise de frequências (11/11) – texto original recuperado (Simon Singh).



A cifra de Vigenère é um método de criptografia que usa uma série de diferentes cifras de César baseadas em letras de uma senha. Trata-se de uma versão simplificada de uma mais geral cifra de substituição polialfabética

mensagem:	T	H	E	C	A	K	E	I	S	A	L	I	E
chave:	P	O	R	T	A	L	P	O	R	T	A	L	P
mensagem cifrada:	I	V	V	V	A	V	T	W	J	T	L	T	T

## Técnica para decodificar uma cifra de Vigenère

Cifras polialfabéticas são mais difíceis de quebrar por análise de frequência, mas têm uma fraqueza: Se a chave é curta e constantemente repetida, como resultado, palavras comuns como "DE" vão provavelmente aparecer criptografadas segundo as mesmas letras da chave, levando à descoberta de padrões repetidos no texto.

Quando no processo de teste de invasão, informação inelegível, porém textual, for encontrada, deve-se tentar criptoanalísá-la, sob a premissa de ter sido protegida por uma cifra clássica:

- 1 Calcule o índice de coincidência do texto cifrado.

**2**

Se o índice for próximo aos de um idioma, há muita redundância no texto, o que é um indicativo de que uma cifra de substituição simples foi empregada:

**2.1**

Realize busca exaustiva de chaves contra a cifra de deslocamento.

**2.2**

Se não for bem sucedido no passo anterior, aplique a técnica de análise de frequências.

**3**

**Senão, se o índice de coincidência for próximo a ICA, uma cifra polialfabética foi utilizada:**

**3.1**

**Aplique o método de Babbage para quebra da cifra de Vigenère.**

**Muitas vezes, desenvolvedores acreditam que é difícil recuperar informação a partir de binários.**

**Mesmo que fosse, deve haver separação de responsabilidades entre desenvolvimento e produção.**

**No caso mais simples, basta executar o comando “strings”, para recuperar a chave embutida.**

## Exemplo:

```
#include <stdio.h>
int main() {
    char key[] =
    "0a3bc178940fd43047027cda807409af";
    ...
    printf("Cifrando dados.
    Aguarde...\n");
    ...
}
```

# Recuperação de chaves embutidas no código

```
...  
__libc_start_main  
GLIBC_2.0  
PTRh  
0a3b  
c178  
940f  
d430  
4702  
7cda  
8074  
09af  
[^_]  
Cifrando dados. Aguarde...
```



**Boas chaves criptográficas devem ser geradas por processos que forneçam o máximo de aleatoriedade possível, impedindo, assim, que um adversário seja capaz de descobri-las por métodos determinísticos.**

**Alguns exemplos de problemas comumente encontrados neste contexto estão enumerados a seguir:**

- **Chaves fixas e com valores previsíveis, como “0x00000000...”, “0x11111111...”, “0x0123456789...”, etc.**

- **Utilização de geradores de números pseudo-aleatórios fracos, como os implementados pela classe “Random”, em Java, e pela função “rand()”, em C.**

**Alguns exemplos de problemas comumente encontrados neste contexto estão enumerados a seguir:**

- **Escolha de valores a partir de um pequeno subconjunto do domínio de chaves.**

- **Uso de sementes previsíveis em geradores de números pseudo-aleatórios.**

**Alguns exemplos de problemas comumente encontrados neste contexto estão enumerados a seguir:**

- **Inicialização de geradores de números pseudo-aleatórios com sementes de poucos bits.**

- **Reutilização integral ou parcial de chaves criptográficas.**

# Geração de chaves com baixa entropia

```
01 import java.util.Random;
02 public class GeraChave {
03     private static Random random = new Random();
04     public static byte[] geraChave(int numBits) {
05         int numBytes = numBits / 8;
06         byte chave[] = new byte[numBytes];
07         // Atribui a cada byte da chave um valor
08         // inteiro j, tal que 0 <= j <= 9
09         for (int i = 0; i < numBytes; i++) {
10             chave[i] = (byte)random.nextInt(10);
11         }
12         return chave;
13     }
14 }
```

# Modo de operação inadequado

## Modo ECB:

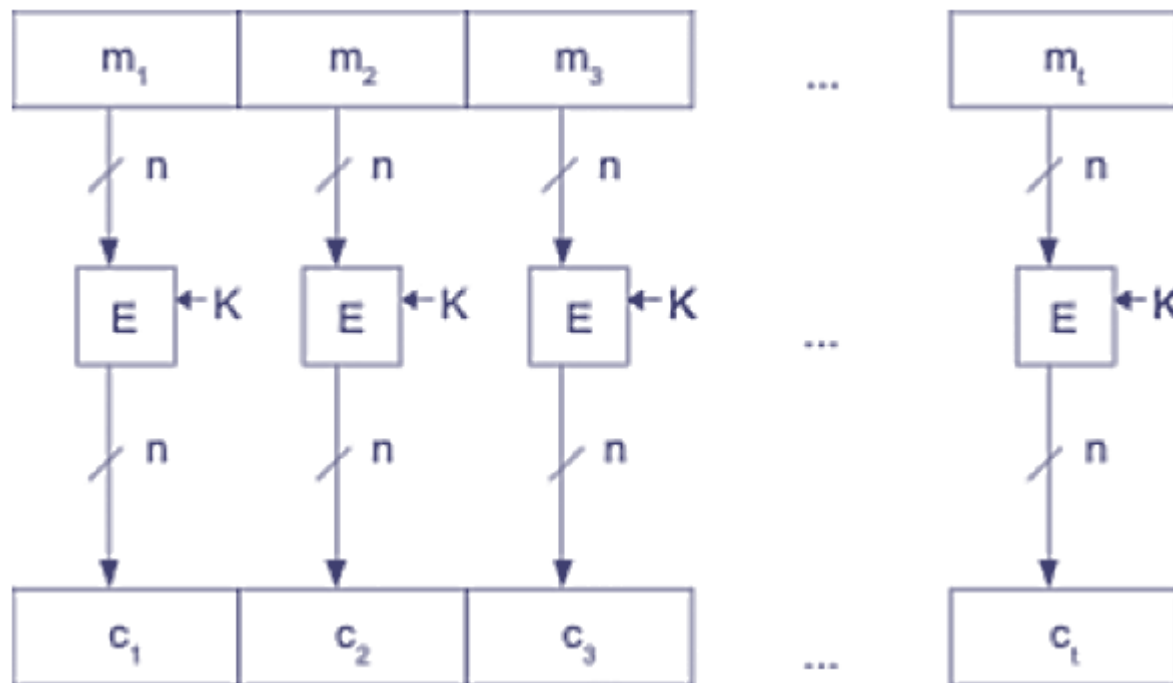


Figura 9.52 - Modo de operação ECB - Ciframento.

## Modo ECB:

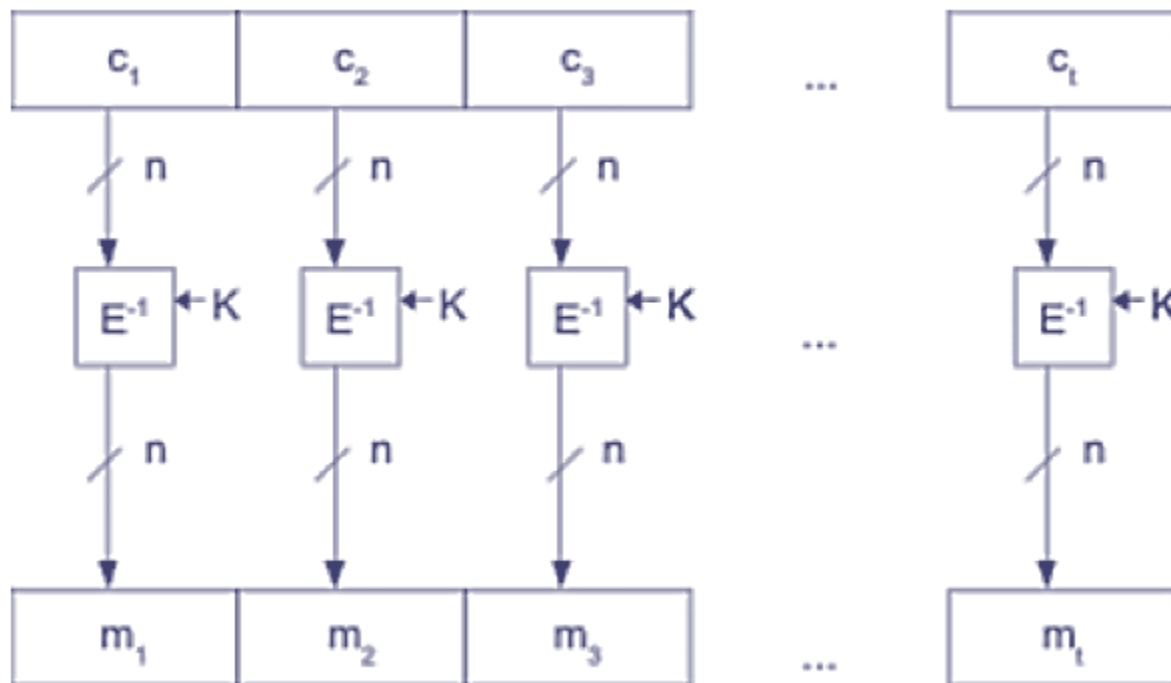


Figura 9.53 - Modo de operação ECB - Deciframento.

## Modo ECB:

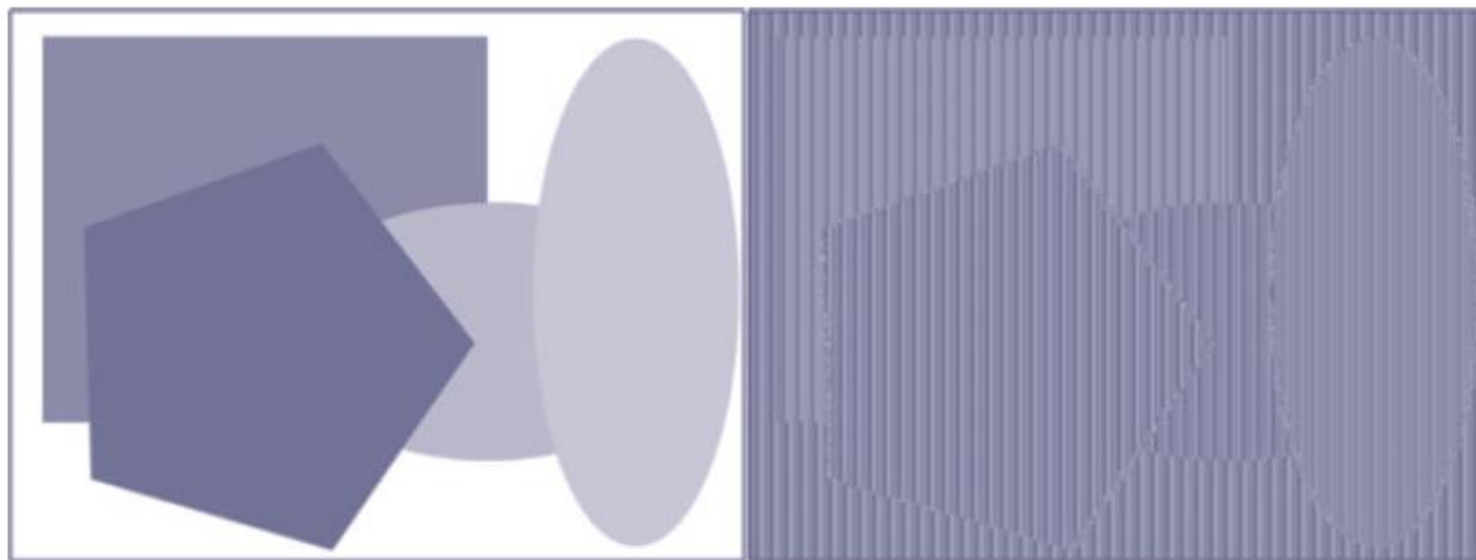


Figura 9.54 - A imagem à direita é o resultado do ciframento, em modo ECB, da outra.



## Modo CBC:

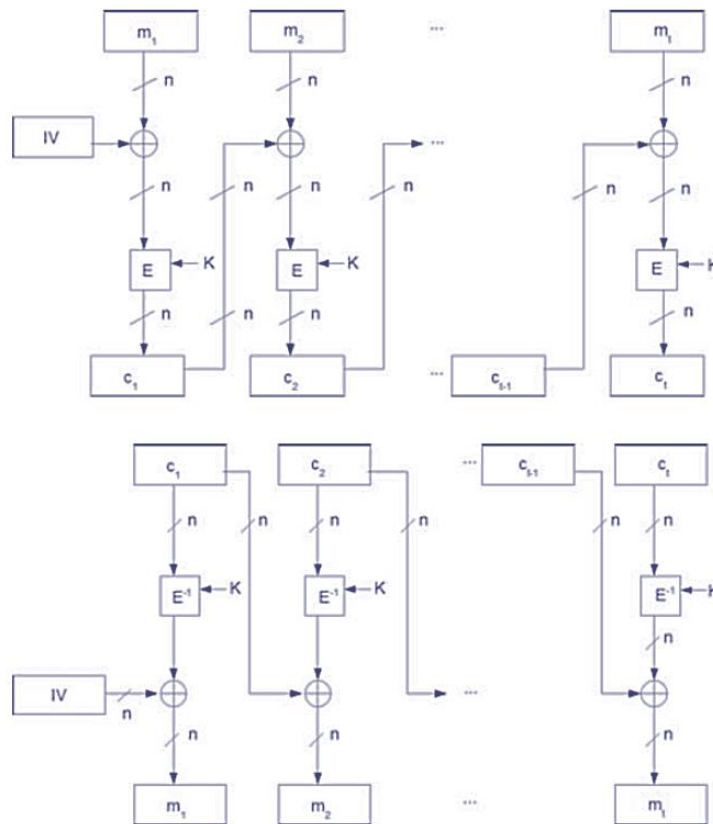


Figura 9.55 - Modo de operação CBC - Ciframento.

# Modo de operação inadequado

## Modo CBC:

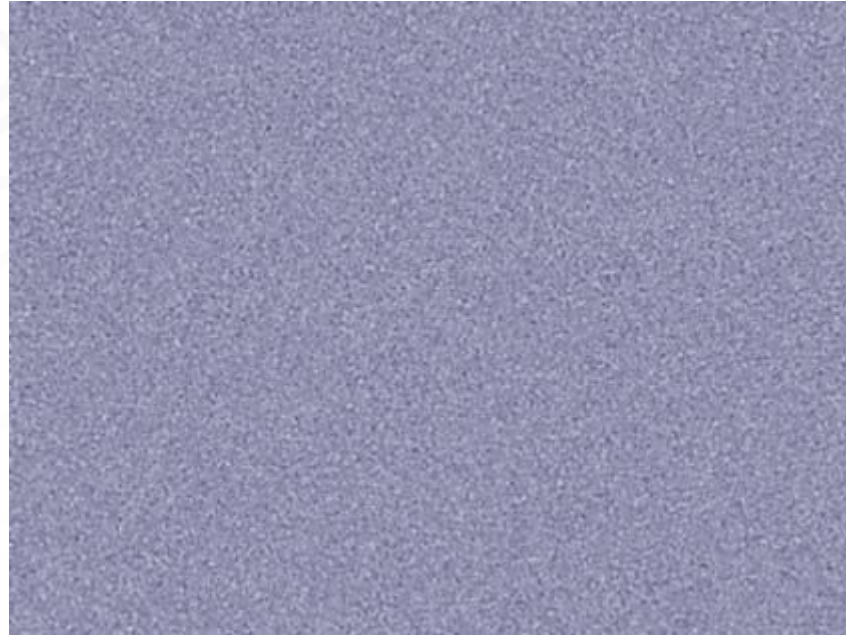


Figura 9.57 - Ciframento em modo CBC da imagem ilustrada na Figura 224.

**Cifras de fluxo aditivas e binárias são cifras síncronas, isto é, geram o fluxo de chaves independentemente dos textos em claro e cifrado.**

**Além disso, elas operam diretamente com bits e utilizam o operador XOR, para combinar a entrada com o fluxo de chaves.**

**Como ele é gerado em função apenas da chave inicial, quando esta é repetida, o mesmo fluxo é obtido.**

O uso de uma mesma chave, em cifras de fluxo aditivas e binárias, pode comprometer a segurança das informações:

Diversas mensagens  $m_1, m_2, \dots, m_n$ , de mesmo tamanho, são cifradas, individualmente, com uma mesma chave  $K$ , o que implica que o fluxo de chaves  $z$  é o mesmo para todas elas.

Do ciframento da mensagem  $m_i$ , resulta  
$$c_i = m_i \oplus z.$$

**O uso de uma mesma chave, em cifras de fluxo aditivas e binárias, pode comprometer a segurança das informações:**

**Calculando o XOR de duas mensagens cifradas, é possível cancelar a ação do fluxo de chaves e obter o XOR dos textos em claro correspondentes.**

**A partir disso, é possível realizar uma análise estatística, com base na redundância do idioma, para recuperar as mensagens originais.**

**O uso de uma mesma chave, em cifras de fluxo aditivas e binárias, pode comprometer a segurança das informações:**

Um ataque mais poderoso é possível quando se conhece o texto em claro  $m_i$  correspondente a um texto cifrado  $c_i$ , pois isto permite a recuperação do fluxo de chaves  $z$ , o qual basta para decifrar todas as mensagens protegidas com a mesma chave  $K$ :

$$c_i \oplus m_i = (m_i \oplus z) \oplus m_i = m_i \oplus m_i \oplus z = 0 \oplus z = z$$

# Mistura de algoritmos com níveis diferentes de segurança

Nível de segurança	Cifras simétricas	Algoritmos assimétricos			Funções de hash criptográficas (tamanho do hash)
		Fatoração de inteiros	Logaritmo discreto	Logaritmo discreto elíptico	
80	2-TDES	RSA-1024	DH-1024	ECDSA-160	RIPEMD-160
112	3-TDES	RSA-2048	DH-2048	ECDSA-224	SHA-224
128	AES-128	RSA-3072	DH-3072	ECDSA-256	SHA-256
192	AES-192	RSA-7680	DH-7680	ECDSA-384	SHA-384
256	AES-256	RSA-15360	DH-15360	ECDSA-512	SHA-512

Figura 9.59 - Nível de segurança de diversos algoritmos criptográficos, em função do tamanho da chave (Barker et al., 2007).

**O emprego de mecanismos criptográficos com fraquezas conhecidas deve ser totalmente evitado para que a segurança da informação não seja comprometida.**

**Exemplos de algoritmos que não devem ser mais utilizados atualmente incluem:**

**Data Encryption Standard  
(DES)**

**RSA com chaves menores  
que 1024 bits**

**MD5**

**SHA-1**



**Normalmente, para proteger senhas de usuários, os sistemas armazenam apenas os hashes delas.**

**A segurança deste mecanismo, porém, depende de algumas premissas que nem sempre são satisfeitas.**

**A primeira fraqueza resulta da qualidade das senhas escolhidas pelos usuários, que, comumente, empregam palavras da língua ou informações pessoais.**

**Neste cenário, um atacante pode montar um dicionário de palavras conhecidas e variações e pré-computar os respectivos hashes.**

**Para impedir que isso aconteça, é comum o emprego de salts.**

**O segundo pecado na proteção de senhas ocorre quando um sistema utiliza hashes criptográficos para proteção de senhas pertencentes a um domínio de baixa cardinalidade, isto é, o número de senhas possíveis é pequeno.**

O requisito 3.4 do padrão PCI DSS demanda que números de cartões de pagamento (PANs) sejam protegidos, durante o armazenamento, por meio de métodos seguros.

Técnicas aceitas incluem o truncamento do número, o uso de funções de hash criptográficas, substituição por tokens e emprego de cifras simétricas.

**Um número de cartão de crédito ou débito possui de 15 a 16 dígitos, considerando-se as bandeiras Visa, Mastercard, American Express, JCB e Discover.**

**Os seis primeiros dígitos do número correspondem ao BIN, responsável pela identificação do banco, e o último é utilizado como dígito verificador.**

**Considerando que a quantidade de bancos em um único país não é extenso e que a grande maioria das compras é realizada com cartões de residentes, o conjunto de BINs aplicáveis é razoavelmente pequeno.**

Números de cartão devem satisfazer o Algoritmo de Luhn, o que restringe o dígito verificador para um único valor por PAN.

Assim, o total de dígitos que devem ser descobertos para um BIN arbitrário é, no máximo, nove.

Logo, um ataque bem sucedido contra um único BIN deve ser capaz de pré-calcular 1 bilhão de hashes, o que está ao alcance de computadores comuns.

Bandeira	Dígitos	Número de dígitos conhecidos										
		0	1	2	3	4	5	6	7	8	9	10
Visa	16	138.888,89	13.888,89	1.388,89	138,89	13,89	1,39	0,14	0,01	0,00	0,00	0,00
Mastercard	16	138.888,89	13.888,89	1.388,89	138,89	13,89	1,39	0,14	0,01	0,00	0,00	0,00
JCB	16	138.888,89	13.888,89	1.388,89	138,89	13,89	1,39	0,14	0,01	0,00	0,00	0,00
Discover	16	138.888,89	13.888,89	1.388,89	138,89	13,89	1,39	0,14	0,01	0,00	0,00	0,00
Amex	15	13.888,89	1.388,89	138,89	13,89	1,39	0,14	0,01	0,00	0,00	0,00	0,00

**Tabela:** Tempo em horas para cálculo de hashes de número de cartão, em função do número de dígitos conhecidos, com processador Core 2 Duo 2,4 GHz.

**Um armazenamento seguro de informações requer a seleção adequada de criptossistemas, o gerenciamento das chaves criptográficas utilizadas e o zelo adequado com a manipulação de chaves pelos programas. Desse modo:**

**Classifique as informações e cifre aquelas que necessitam que o requisito de sigilo seja satisfeito.**

**Se não for necessário, não armazene dados sensíveis, após serem processados, evitando, assim, ter de protegê-los.**



**Um armazenamento seguro de informações requer a seleção adequada de criptossistemas, o gerenciamento das chaves criptográficas utilizadas e o zelo adequado com a manipulação de chaves pelos programas. Desse modo:**

**Não crie seus próprios algoritmos e protocolos criptográficos.**

**Nunca empregue cifras clássicas para proteger informações sensíveis e lembre-se sempre de que BASE64 não é um algoritmo criptográfico.**

**Um armazenamento seguro de informações requer a seleção adequada de criptossistemas, o gerenciamento das chaves criptográficas utilizadas e o zelo adequado com a manipulação de chaves pelos programas. Desse modo:**

**Não utilize algoritmos criptográficos com fraquezas conhecidas.**

**Quando cifras de bloco forem utilizadas, empregue modos de operação adequados a cada situação.**

**Um armazenamento seguro de informações requer a seleção adequada de criptossistemas, o gerenciamento das chaves criptográficas utilizadas e o zelo adequado com a manipulação de chaves pelos programas. Desse modo:**

- **Use algoritmos criptográficos para satisfazerem apenas os requisitos de segurança da informação para os quais foram criados.**

- **Quando uma solução empregar diversos algoritmos criptográficos, determine o nível mínimo desejado de segurança e adote somente criptossistemas que atendam o limiar escolhido.**

**Um armazenamento seguro de informações requer a seleção adequada de criptossistemas, o gerenciamento das chaves criptográficas utilizadas e o zelo adequado com a manipulação de chaves pelos programas. Desse modo:**

**Muito cuidado ao proteger informações pertencentes a domínios de baixa cardinalidade, pois ataques de dicionários podem ser facilitados, dependendo da solução adotada.**

**Não proteja números de cartão de crédito e débito com funções de hash criptográficas, apesar de aceitas para esse propósito pelo PCI DSS.**

**Um armazenamento seguro de informações requer a seleção adequada de criptossistemas, o gerenciamento das chaves criptográficas utilizadas e o zelo adequado com a manipulação de chaves pelos programas. Desse modo:**

**Crie e utilize processos e procedimentos para gerenciamento de chaves criptográficas.**

**Utilize bons geradores de números pseudo-aleatórios para a criação de chaves.**

**Um armazenamento seguro de informações requer a seleção adequada de criptossistemas, o gerenciamento das chaves criptográficas utilizadas e o zelo adequado com a manipulação de chaves pelos programas. Desse modo:**

**Considere empregar hardware seguro, para realizar as operações criptográficas e armazenar as chaves relacionadas, quando os dados protegidos forem extremamente sensíveis.**

**Se não for possível proteger chaves por meio de hardware seguro, utilize protocolos de partilha de segredos com vários custodiantes.**

**Um armazenamento seguro de informações requer a seleção adequada de criptossistemas, o gerenciamento das chaves criptográficas utilizadas e o zelo adequado com a manipulação de chaves pelos programas. Desse modo:**

**Nunca embuta chaves criptográficas e senhas no código do programa.**

**Limpe a memória alocada para chaves criptográficas antes de liberá-la para o sistema operacional.**

**Um armazenamento seguro de informações requer a seleção adequada de criptossistemas, o gerenciamento das chaves criptográficas utilizadas e o zelo adequado com a manipulação de chaves pelos programas. Desse modo:**

**Marque as páginas de memória, contendo chaves criptográficas, como inelegíveis para swap.**





## Exercício de Nivelamento 1

### Acesso à aplicação web

---

- ▲ Você já acessou alguma aplicação web para a qual o navegador web tenha apresentado erro
- ▲ relacionado ao certificado digital?



## Exercício de Fixação 1

### Tipos de vulnerabilidades

---

1. Que tipos de vulnerabilidades podem estar presentes na configuração de um túnel SSL/TLS?
2. Que falhas recentes foram encontradas nos protocolos SSL/TLS?



## Exercício de Nivelamento 2

### Chave criptográfica

---

- ▲ Você já embutiu uma chave criptográfica em algum programa ou script que tenha desenvolvido?



## Exercício de Fixação 3

### ECB

---

1. Por que o modo de operação ECB não é adequado para mensagens com tamanho maior que
2. o tamanho do bloco da cifra utilizada?



## Exercício de Fixação 4

### Ataque a algoritmos

---

1. Que algoritmo é mais difícil de ser atacado: AES com chave de 128 bits ou RSA com chave de 1024 bits?

# Perguntas



## Caderno de Atividade 9

1

## Vulnerabilidades no transporte de informações



## Caderno de Atividade 9

2

## Vulnerabilidades no armazenamento de informações





# Teste de Invasão de Aplicações Web

## Capítulo 9

### Mecanismos criptográficos