Usando criptografia de forma prática e descomplicada

Marcos Azevedo aka psylinux

BSidesSP - Edição 0xF - Maio 2018

Sumário

Apresentação Pessoal

- Marcos Azevedo aka psylinux
- Consultor de Segurança (Pentester) na Cipher
- Apaixonado por Segurança Ofensiva
- Faixa Preta de Jiu-Jitsu
- Atirador esportivo
- e-mail1: lembrei@email.com
- e-mail2: esqueci@email.com

Apresentação Pessoal

Figura: Goiânia, Goiás

- **Dinheiro pra sapecar porco**: Muito dinheiro.
- **Custoso (a)**: Difícil, pessoa sapeca.
- **Demais da conta**: Muito, muito mesmo.
- **Encabulado**: Impressionado.
- **Pizêro**: Bagunça.
- **Pulá o corguim**: Passar dos limites.
- **Apiar**: Descer.
- **Rudeia/Rudiá**: Dar a volta.

- **Dinheiro pra sapecar porco**: Muito dinheiro.
- **Custoso (a)**: Difícil, pessoa sapeca.
- **Demais da conta**: Muito, muito mesmo.
- **Encabulado**: Impressionado.
- **Pizêro**: Bagunça.
- **Pulá o corguim**: Passar dos limites.
- **Apiar**: Descer.
- **Rudeia/Rudiá**: Dar a volta.

- **Dinheiro pra sapecar porco**: Muito dinheiro.
- **Custoso (a)**: Difícil, pessoa sapeca.
- **Demais da conta**: Muito, muito mesmo.
- **Encabulado**: Impressionado.
- **Pizêro**: Bagunça.
- **Pulá o corguim**: Passar dos limites.
- **Apiar**: Descer.
- **Rudeia/Rudiá**: Dar a volta.

- **Dinheiro pra sapecar porco**: Muito dinheiro.
- **Custoso (a)**: Difícil, pessoa sapeca.
- **Demais da conta**: Muito, muito mesmo.
- **Encabulado**: Impressionado.
- **Pizêro**: Bagunça.
- **Pulá o corguim**: Passar dos limites.
- **Apiar**: Descer.
- **Rudeia/Rudiá**: Dar a volta.

- **Dinheiro pra sapecar porco**: Muito dinheiro.
- **Custoso (a)**: Difícil, pessoa sapeca.
- **Demais da conta**: Muito, muito mesmo.
- **Encabulado**: Impressionado.
- **Pizêro**: Bagunça.
- **Pulá o corguim**: Passar dos limites.
- **Apiar**: Descer.
- **Rudeia/Rudiá**: Dar a volta.

- **Dinheiro pra sapecar porco**: Muito dinheiro.
- **Custoso (a)**: Difícil, pessoa sapeca.
- **Demais da conta**: Muito, muito mesmo.
- **Encabulado**: Impressionado.
- **Pizêro**: Bagunça.
- **Pulá o corguim**: Passar dos limites.
- **Apiar**: Descer.
- **Rudeia/Rudiá**: Dar a volta.

- **Dinheiro pra sapecar porco**: Muito dinheiro.
- **Custoso (a)**: Difícil, pessoa sapeca.
- **Demais da conta**: Muito, muito mesmo.
- **Encabulado**: Impressionado.
- **Pizêro**: Bagunça.
- **Pulá o corguim**: Passar dos limites.
- **Apiar**: Descer.
- **Rudeia/Rudiá**: Dar a volta.

- **Dinheiro pra sapecar porco**: Muito dinheiro.
- **Custoso (a)**: Difícil, pessoa sapeca.
- **Demais da conta**: Muito, muito mesmo.
- **Encabulado**: Impressionado.
- **Pizêro**: Bagunça.
- **Pulá o corguim**: Passar dos limites.
- **Apiar**: Descer.
- **Rudeia/Rudiá**: Dar a volta.

Criptografia vs Codificação
Tem diferença?

1. **Objetivo da Codificação:** transformar os dados para que eles possam ser adequadamente utilizados por um diferente tipo de sistema, por exemplo, caracteres especiais de uma página web. **Não serve para manter em segredo** as informações, mas sim garantir que o sistema interprete de outra forma os dados contidos na mensagem.
2. **Exemplos:** ASCII, Unicode, URL Encoding, Base64.

1. **Objetivo da Codificação:** transformar os dados para que eles possam ser adequadamente utilizados por um diferente tipo de sistema, por exemplo, caracteres especiais de uma página web. **Não serve para manter em segredo** as informações, mas sim garantir que o sistema interprete de outra forma os dados contidos na mensagem.
2. **Exemplos:** ASCII, Unicode, URL Encoding, Base64.

1. **Objetivo da Criptografia:** transformar os dados, afim de manter uma "mensagem" em segredo e garantir que os dados sejam inteligíveis apenas para o destinatário com a chave para reverter a criptografia.

Parte 1

Descomplicando a Critografia


Cripto (do grego *kriptos*)
+
Grafia (do grego *graphein*)
=
Criptografia
Esconder/Ocultar
Escrever
Escrita oculta ou secreta

”Criptografia”
Qual a relevância desse tema?

Quando queremos que apenas o **EMISSOR** e o **DESTINATÁRIO** compreendam o conteúdo da mensagem.

Qual a relevância desse tema?
Quando usamos criptografia?

- **"Em trânsito"**, neste contexto, é quando você envia informações através da Internet, por e-mail ou quando precisa armazená-la em outro lugar que não seja o seu próprio dispositivo.
- **"Em repouso"**, quando estão armazenados em seu dispositivo, que pode ser parte integrada como um disco rígido, ou em um meio removível, como uma unidade USB.

Qual a relevância desse tema?
Quando usamos criptografia?

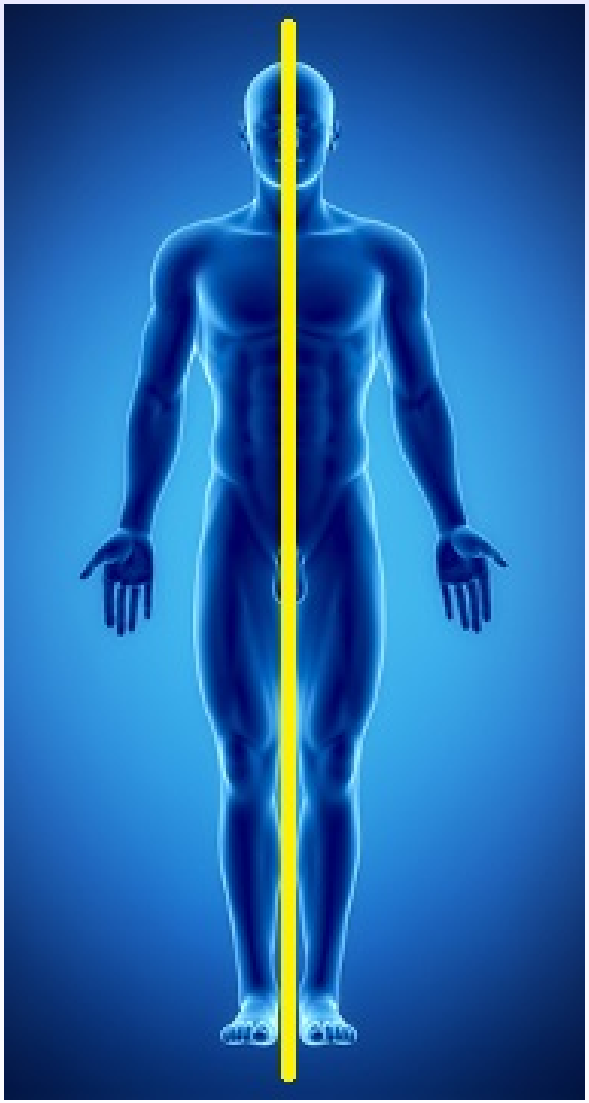
- **"Em trânsito"**, neste contexto, é quando você envia informações através da Internet, por e-mail ou quando precisa armazená-la em outro lugar que não seja o seu próprio dispositivo.
- **"Em repouso"**, quando estão armazenados em seu dispositivo, que pode ser parte integrada como um disco rígido, ou em um meio removível, como uma unidade USB.

Criptologia: disciplina que reúne os conhecimentos e as técnicas necessários à criptoanálise ('solução de criptogramas') e à criptografia ('modificação codificada').

Texto Claro: Texto original, não cifrado.

Texto Cifrado: Texto ilegível, não compreensível.

Simetria: conformidade, em medida, forma e posição relativa, entre as partes dispostas em cada lado de uma linha divisória, um plano médio, um centro ou um eixo.



Assimetria: 1. ausência de simetria. 2. grande diferença; disparidade, discrepância.

Figura: "Abaporu", da Pintora Brasileira Tarsila do Amaral.

Encriptar/Cifrar/Criptografar: 1. reproduzir (mensagem) em código não conhecido, tornando-a, desse modo, intencionalmente ininteligível para os que não têm acesso às suas convenções. 2. inf codificar (informação) de modo que somente destinatários autorizados possam ter acesso a ela; encriptar.

Decriptar/Decifrar/Descriptografar: traduzir ou decifrar mensagens ou códigos cifrados ou criptografados

Bit: menor parcela de informação processada por um computador. Algarismo do sistema binário que somente pode assumir as formas 0 ou 1

Algoritmo: 1. mat sequência finita de regras, raciocínios ou operações que, aplicada a um número finito de dados, permite solucionar classes semelhantes de problemas. 2. inf conjunto das regras e procedimentos lógicos perfeitamente definidos que levam à solução de um problema em um número finito de etapas.

Tipos de Criptografia

1. **Criptografia Clássica:** Sustenta-se em leis matemáticas. Já era utilizado nos hieróglifos de monumentos do Antigo Egito (cerca de 4500 anos atrás).
2. **Criptografia Moderna:** Utiliza basicamente os mesmos princípios da Criptografia Clássica, mas faz uso de computadores para processar os dados. Consideramos seu inicio em 1949 com Matemático Americano Claude Shannon.
3. **Criptografia Quântica:** Utiliza-se de elementos físicos, tal com fótons, para gerar chaves quânticas *inquebráveis*.

Tipos de Criptografia

1. **Criptografia Clássica:** Sustenta-se em leis matemáticas. Já era utilizado nos hieróglifos de monumentos do Antigo Egito (cerca de 4500 anos atrás).
2. **Criptografia Moderna:** Utiliza basicamente os mesmos princípios da Criptografia Clássica, mas faz uso de computadores para processar os dados. Consideramos seu inicio em 1949 com Matemático Americano Claude Shannon.
3. **Criptografia Quântica:** Utiliza-se de elementos físicos, tal com fótons, para gerar chaves quânticas *inquebráveis*.

Tipos de Criptografia

1. **Criptografia Clássica:** Sustenta-se em leis matemáticas. Já era utilizado nos hieróglifos de monumentos do Antigo Egito (cerca de 4500 anos atrás).
2. **Criptografia Moderna:** Utiliza basicamente os mesmos princípios da Criptografia Clássica, mas faz uso de computadores para processar os dados. Consideramos seu inicio em 1949 com Matemático Americano Claude Shannon.
3. **Criptografia Quântica:** Utiliza-se de elementos físicos, tal com fótons, para gerar chaves quânticas *inquebráveis*.

1. **Algoritmo de transposição:** rearranja a ordem dos caracteres de uma mensagem. Um exemplo simples é a transformação de **"muito obrigado"** em **"omtui oobdraig"**. Esta categoria de algoritmo criptográfico é composta por uma função bijetora para efetuar encriptações e sua inversa faz a mensagem voltar à forma original;
2. **Algoritmo de substituição:** substitui caracteres ou grupos de caracteres por outros caracteres ou grupos de caracteres. Um exemplo simples: **"muito obrigado"** é transformado em **"nvjup pcsjhbep"**, substituindo cada letra pela próxima na sequência alfabética.

1. **Algoritmo de transposição:** rearranja a ordem dos caracteres de uma mensagem. Um exemplo simples é a transformação de **"muito obrigado"** em **"omtui oobdraig"**. Esta categoria de algoritmo criptográfico é composta por uma função bijetora para efetuar encriptações e sua inversa faz a mensagem voltar à forma original;
2. **Algoritmo de substituição:** substitui caracteres ou grupos de caracteres por outros caracteres ou grupos de caracteres. Um exemplo simples: **"muito obrigado"** é transformado em **"nvjup pcsjhbep"**, substituindo cada letra pela próxima na sequência alfabética.

Alguns algoritmos da Criptografia Moderna

1. **Algoritmo de Euclides:** Ele é um elemento-chave dos algoritmos RSA (Criptografia de Chaves Assimétricas).
2. **Transformada de Fourier:** Utilizada em esteganografia.
3. **Algoritmos de Aritmética Modular:** Também conhecida como aritmética do relógio, aplicado a números inteiros. Utilizada em Criptografia de Chave Simétrica, CPF, Código de Barras.

Entendendo o Algoritmo de Euclides

O Algoritmo de Euclides nos fornece a seguinte propriedade: na *k*-ésima iteração, vale que

$$r_{k+1} = r_{k-1} - r_k q_k$$

em que $q_k = \frac{r_{k-1}}{r_k}$ é uma divisão inteira.

O algoritmo acaba quando $r_{k+1} = 0$, definindo o resto atual como o máximo divisor comum: $r_k = MDC(a, b)$.

Para estender o algoritmo, queremos também manter a seguinte propriedade:

$$r_k = au_k + bv_k$$

dessa forma, quando o algoritmo acabar, teremos valores u_k e v_k que satisfazem o teorema de Bézout.

Para isso, assuma que nós temos esses valores para a iteração k e para a iteração anterior, $k-1$: ou seja, assuma que já temos os valores que satisfazem as duas igualdades a seguir:

$$r_k = au_k + bv_k$$

e

$$r_{k-1} = au_{k-1} + bv_{k-1}$$

então, para o próximo resto, teremos

$$\begin{aligned} r_{k+1} &= r_{k-1} - r_k q_k \\ &= (au_{k-1} + bv_{k-1}) - (au_k + bv_k)q_k \\ &= au_{k-1} - au_k q_k + bv_{k-1} - bv_k q_k \\ &= a(u_{k-1} - u_k q_k) + b(v_{k-1} - v_k q_k) \\ &= au_{k+1} + bv_{k+1} \end{aligned}$$

Ou seja, se a igualdade de Bézout vale para a iteração atual do algoritmo e para a iteração anterior, então, ela vale para a próxima e os valores de Bézout são

$$u_{k+1} = u_{k-1} - u_k q_k$$

e

$$v_{k+1} = v_{k-1} - v_k q_k$$

Entendendo o Algoritmo de Euclides

O Algoritmo de Euclides nos fornece a seguinte propriedade: na *k*-ésima iteração, vale que

$$r_{k+1} = r_{k-1} - r_k q_k$$

em que $q_k = \frac{r_{k-1}}{r_k}$ é uma divisão inteira.

O algoritmo acaba quando $r_{k+1} = 0$, definindo o resto atual como o máximo divisor comum: $r_k = MDC(a, b)$.

Para estender o algoritmo, queremos também manter a seguinte propriedade:

$$r_k = au_k + bv_k$$

dessa forma, quando o algoritmo acabar, teremos valores u_k e v_k que satisfazem o teorema de Bézout.

Para isso, assuma que nós temos esses valores para a iteração k e para a iteração anterior, $k-1$: ou seja, assuma que já temos os valores que satisfazem as duas igualdades a seguir:

$r_k = au_k + bv_k$

e

$r_{k-1} = au_{k-1} + bv_{k-1}$

então, para o próximo resto, teremos

$$\begin{aligned} r_{k+1} &= r_{k-1} - r_k q_k \\ &= (au_{k-1} + bv_{k-1}) - (au_k + bv_k)q_k \\ &= au_{k-1} - au_k q_k + bv_{k-1} - bv_k q_k \\ &= a(u_{k-1} - u_k q_k) + b(v_{k-1} - v_k q_k) \\ &= au_{k+1} + bv_{k+1} \end{aligned}$$

Ou seja, se a igualdade de Bézout vale para a iteração atual do algoritmo e para a iteração anterior, então, ela vale para a próxima e os valores de Bézout são

$$u_{k+1} = u_{k-1} - u_k q_k$$

e

$$v_{k+1} = v_{k-1} - v_k q_k$$

BRINCADEIRA GENTE! Nós não iremos nos aprofundar na matemática. Podem ficar tranquilos... :-)

Mensagem: **CRIPTOGRAFIA NA BSIDES SAO PAULO 2018**

	1	2	3	4	5	6	7	8
1	C	R	I	P	T	O	G	R
2	A	F	I	A	N	A	B	S
3	I	D	E	S	S	A	O	P
4	A	U	L	O	2	0	1	8

Chave criptográfica: **24176835**

	1	2	3	4	5	6	7	8
1	I	C	G	R	R	T	P	O
2	I	A	B	F	S	N	A	A
3	E	I	O	D	P	S	S	A
4	L	A	1	U	8	2	O	0

3M UM D14 D3 VER40, 3S7AVA N4 PR4I4,O853RV4NDO DU4S CR14NÇ4S 8B1NC4ND0 N4 4REI4. EL45 TR4B4LH4V4M MUI7O C0N57R1ND0 UM C4ATEL0 D3 AR3I4, C0M 70RR35, P4554R3L4S 3 P4554G3N5 1N7ERN4S. QU4ND0 ES74V4M QU4S3 T3RM1N4ND0, V310 UM4 0ND4 3 3S7RU1U 7UDO, R3DU21NDO 0 C4S7EL0 4 UM MON73 D3 4REI4 3 3SPUM4. 4CH31 QU3 D3P01S D3 74N70 35FORÇ0 3 CU1D4D0, 45 CR1ANC4S C4IR4M N0 CH0R0, CORR3R4M P3L4 PR41A, FUG1ND0 DA 4GU4, R1NDO D3 M405 D4D4S 3 C0M3C4R4M 4 C0NS7RU1R 0UTR0 C4573LO.

Algumas Aplicações da Criptografia Atualmente

1. Sigilo em banco de dados;
2. Censos;
3. Investigações governamentais;
4. Dossiês de pessoas sob investigação;
5. Dados hospitalares;
6. Informações de crédito pessoal;
7. Decisões estratégicas empresariais;
8. Sigilo em comunicação de dados;
9. Comandos militares;
10. Mensagens diplomáticas;
11. Operações bancárias;
12. Comércio eletrônico;
13. Transações por troca de documentos eletrônicos (EDI);
14. Estudo de idiomas desconhecidos;
15. Recuperação de documentos arqueológicos, hieróglifos;
16. E até tentativas de comunicações extraterrestres.

O processo é unidirecional e impossibilita descobrir o conteúdo original a partir do Hash. O valor de conferência ("Soma de verificação") muda se um único bit for alterado, acrescentado ou retirado da mensagem.

- **Objetivo das funções de HASH:** Sua única finalidade é fazer um "resumo" representado em base hexadecimal que permite visualização em letras (A a F) e números (0 a 9). O conceito teórico diz que "hash" é a transformação de uma grande quantidade de dados em uma pequena quantidade de informações".
- **Exemplos:** MD4, MD5, SHA-1, SHA-512.

O processo é unidirecional e impossibilita descobrir o conteúdo original a partir do Hash. O valor de conferência ("Soma de verificação") muda se um único bit for alterado, acrescentado ou retirado da mensagem.

- **Objetivo das funções de HASH:** Sua única finalidade é fazer um "resumo" representado em base hexadecimal que permite visualização em letras (A a F) e números (0 a 9). O conceito teórico diz que "hash" é a transformação de uma grande quantidade de dados em uma pequena quantidade de informações".
- **Exemplos:** MD4, MD5, SHA-1, SHA-512.

O Princípio de Kerckhoff

O Princípio de Kerckhoff é um princípio fundamental na criptografia moderna:

"Um sistema de criptografia deve ser seguro mesmo se o adversário conhecer todos os detalhes do sistema, com exceção da chave secreta."

Algoritmos Cifradores

1. **Cifradores de blocos:** divide a mensagem em blocos de tamanho fixo (ex: 256 bits). Por exemplo, DES, AES, 3DES
2. **Cifradores de fluxo:** cifra cada digito do texto plano por vez. Por exemplo, o RC4

Algoritmos Cifradores

1. **Cifradores de blocos:** divide a mensagem em blocos de tamanho fixo (ex: 256 bits). Por exemplo, DES, AES, 3DES
2. **Cifradores de fluxo:** cifra cada digito do texto plano por vez. Por exemplo, o RC4

Tipos de Chaves Criptográficas

1. Criptografia de Chave Simétrica
2. Criptografia de Chaves Assimétricas
3. Esquemas Híbridos

Tipos de Chaves Criptográficas

1. Criptografia de Chave Simétrica
2. Criptografia de Chaves Assimétricas
3. Esquemas Híbridos

Tipos de Chaves Criptográficas

1. Criptografia de Chave Simétrica
2. Criptografia de Chaves Assimétricas
3. Esquemas Híbridos

Criptografia de Chave Simétrica

1. Todos os esquemas de encriptação desde a antiguidade até 1976 eram simétricos.
2. Também conhecido como: criptografia de chave única ou criptografia de chave secreta.
3. A chave precisa ser transmitida através de um canal seguro.
4. Transmissão Wireless com protocolo **WPA (Wi-Fi Protected Access)** [1] utilizam esse modelo criptográfico.

[1] Em 16 de Outubro de 2017, foi divulgado uma vulnerabilidade crítica (KRACK) que afeta milhões de dispositivos.

Criptografia de Chave Simétrica

1. Todos os esquemas de encriptação desde a antiguidade até 1976 eram simétricos.
2. Também conhecido como: criptografia de chave única ou criptografia de chave secreta.
3. A chave precisa ser transmitida através de um canal seguro.
4. Transmissão Wireless com protocolo **WPA (Wi-Fi Protected Access)** [1] utilizam esse modelo criptográfico.

[1] Em 16 de Outubro de 2017, foi divulgado uma vulnerabilidade crítica (KRACK) que afeta milhões de dispositivos.

Criptografia de Chave Simétrica

1. Todos os esquemas de encriptação desde a antiguidade até 1976 eram simétricos.
2. Também conhecido como: criptografia de chave única ou criptografia de chave secreta.
3. A chave precisa ser transmitida através de um canal seguro.
4. Transmissão Wireless com protocolo **WPA (Wi-Fi Protected Access)** [1] utilizam esse modelo criptográfico.

[1] Em 16 de Outubro de 2017, foi divulgado uma vulnerabilidade crítica (KRACK) que afeta milhões de dispositivos.

Criptografia de Chave Simétrica

1. Todos os esquemas de encriptação desde a antiguidade até 1976 eram simétricos.
2. Também conhecido como: criptografia de chave única ou criptografia de chave secreta.
3. A chave precisa ser transmitida através de um canal seguro.
4. Transmissão Wireless com protocolo **WPA (Wi-Fi Protected Access)** [1] utilizam esse modelo criptográfico.

[1] Em 16 de Outubro de 2017, foi divulgado uma vulnerabilidade crítica (KRACK) que afeta milhões de dispositivos.

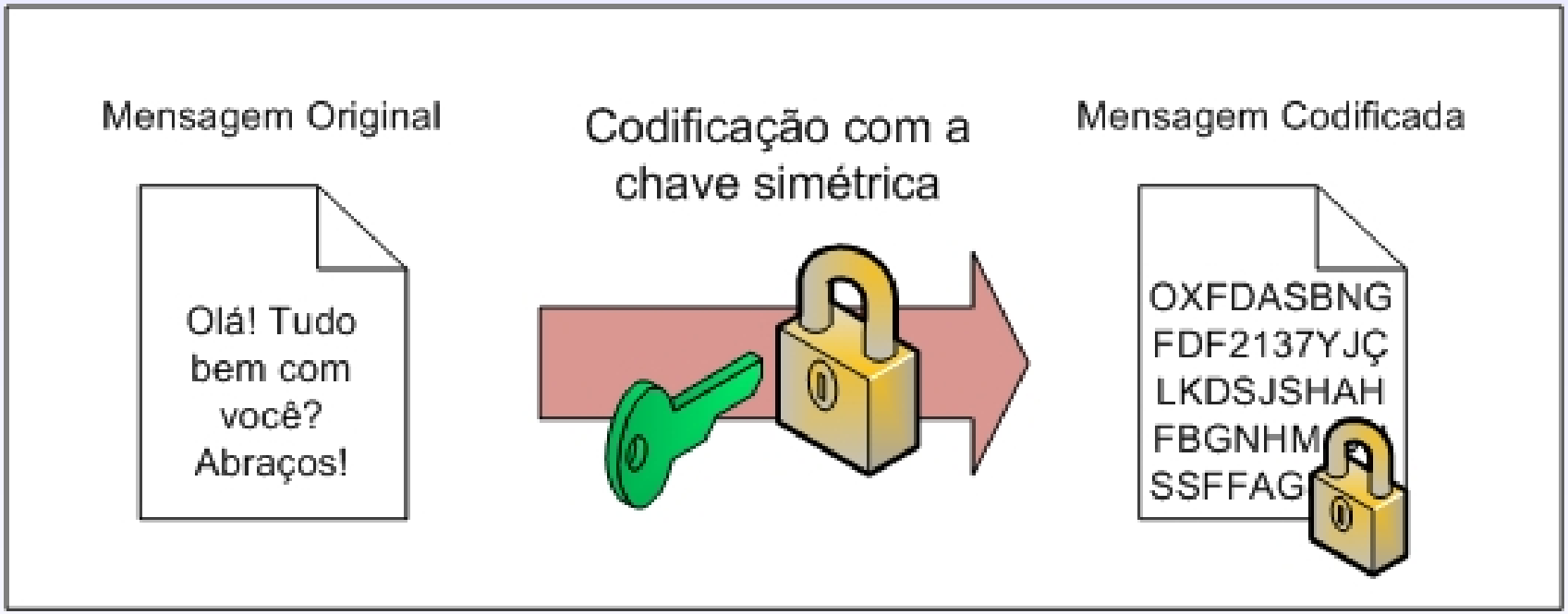


Figura: Criptografando com chave simétrica[2]

[2]Imagem extraída de: https://www.gta.ufrj.br

Figura: Descriptografando com chave simétrica[3]

[3]Imagem extraída de: https://www.gta.ufrj.br

Criptografia de Chave Simétrica

Exemplos de algoritmos simétricos populares:

1. AES
2. Twofish
3. Serpent
4. Blowfish
5. CAST5
6. RC4
7. 3DES (baseado no DES)
8. IDEA

- Como distribuir as chaves de maneira segura?
- Como verificar se a mensagem não foi modificada?
- Como ter certeza que a mensagem foi realmente enviada por quem diz ter enviado?

- Como distribuir as chaves de maneira segura?
- Como verificar se a mensagem não foi modificada?
- Como ter certeza que a mensagem foi realmente enviada por quem diz ter enviado?

- Como distribuir as chaves de maneira segura?
- Como verificar se a mensagem não foi modificada?
- Como ter certeza que a mensagem foi realmente enviada por quem diz ter enviado?

1. Em 1976, Diffie, Hellman e Merkle propuseram a criptografia de chave pública, também conhecida como assimétrica.
2. Baseado no par de chaves: **pública e privada**
   - Chaves públicas são divulgadas abertamente.
   - Chaves privadas devem ser mantidas em segredo.
   - Não é possível obter a chave privada a partir da pública!
3. Provê:
   - Confidencialidade das mensagens.
   - Autenticação do remetente.
   - Verificação de integridade.
   - Não repudio.

1. Em 1976, Diffie, Hellman e Merkle propuseram a criptografia de chave pública, também conhecida como assimétrica.
2. Baseado no par de chaves: **pública e privada**
   - Chaves públicas são divulgadas abertamente.
   - Chaves privadas devem ser mantidas em segredo.
   - Não é possível obter a chave privada a partir da pública!
3. Provê:
   - Confidencialidade das mensagens.
   - Autenticação do remetente.
   - Verificação de integridade.
   - Não repudio.

1. Em 1976, Diffie, Hellman e Merkle propuseram a criptografia de chave pública, também conhecida como assimétrica.
2. Baseado no par de chaves: **pública e privada**
   - Chaves públicas são divulgadas abertamente.
   - Chaves privadas devem ser mantidas em segredo.
   - Não é possível obter a chave privada a partir da pública!
3. Provê:
   - Confidencialidade das mensagens.
   - Autenticação do remetente.
   - Verificação de integridade.
   - Não repudio.

1. Em 1976, Diffie, Hellman e Merkle propuseram a criptografia de chave pública, também conhecida como assimétrica.
2. Baseado no par de chaves: **pública e privada**
   - Chaves públicas são divulgadas abertamente.
   - Chaves privadas devem ser mantidas em segredo.
   - Não é possível obter a chave privada a partir da pública!
3. Provê:
   - Confidencialidade das mensagens.
   - Autenticação do remetente.
   - Verificação de integridade.
   - Não repudio.

1. Em 1976, Diffie, Hellman e Merkle propuseram a criptografia de chave pública, também conhecida como assimétrica.
2. Baseado no par de chaves: **pública e privada**
   - Chaves públicas são divulgadas abertamente.
   - Chaves privadas devem ser mantidas em segredo.
   - Não é possível obter a chave privada a partir da pública!
3. Provê:
   - Confidencialidade das mensagens.
   - Autenticação do remetente.
   - Verificação de integridade.
   - Não repudio.

1. Em 1976, Diffie, Hellman e Merkle propuseram a criptografia de chave pública, também conhecida como assimétrica.
2. Baseado no par de chaves: **pública e privada**
   - Chaves públicas são divulgadas abertamente.
   - Chaves privadas devem ser mantidas em segredo.
   - Não é possível obter a chave privada a partir da pública!
3. Provê:
   - Confidencialidade das mensagens.
   - Autenticação do remetente.
   - Verificação de integridade.
   - Não repudio.

1. Em 1976, Diffie, Hellman e Merkle propuseram a criptografia de chave pública, também conhecida como assimétrica.
2. Baseado no par de chaves: **pública e privada**
   - Chaves públicas são divulgadas abertamente.
   - Chaves privadas devem ser mantidas em segredo.
   - Não é possível obter a chave privada a partir da pública!
3. Provê:
   - Confidencialidade das mensagens.
   - Autenticação do remetente.
   - Verificação de integridade.
   - Não repudio.

1. Em 1976, Diffie, Hellman e Merkle propuseram a criptografia de chave pública, também conhecida como assimétrica.
2. Baseado no par de chaves: **pública e privada**
   - Chaves públicas são divulgadas abertamente.
   - Chaves privadas devem ser mantidas em segredo.
   - Não é possível obter a chave privada a partir da pública!
3. Provê:
   - Confidencialidade das mensagens.
   - Autenticação do remetente.
   - Verificação de integridade.
   - Não repudio.

1. Em 1976, Diffie, Hellman e Merkle propuseram a criptografia de chave pública, também conhecida como assimétrica.
2. Baseado no par de chaves: **pública e privada**
   - Chaves públicas são divulgadas abertamente.
   - Chaves privadas devem ser mantidas em segredo.
   - Não é possível obter a chave privada a partir da pública!
3. Provê:
   - Confidencialidade das mensagens.
   - Autenticação do remetente.
   - Verificação de integridade.
   - Não repudio.

1. Em 1976, Diffie, Hellman e Merkle propuseram a criptografia de chave pública, também conhecida como assimétrica.
2. Baseado no par de chaves: **pública e privada**
   - Chaves públicas são divulgadas abertamente.
   - Chaves privadas devem ser mantidas em segredo.
   - Não é possível obter a chave privada a partir da pública!
3. Provê:
   - Confidencialidade das mensagens.
   - Autenticação do remetente.
   - Verificação de integridade.
   - Não repudio.

Criptografia de Chave Assimétrica

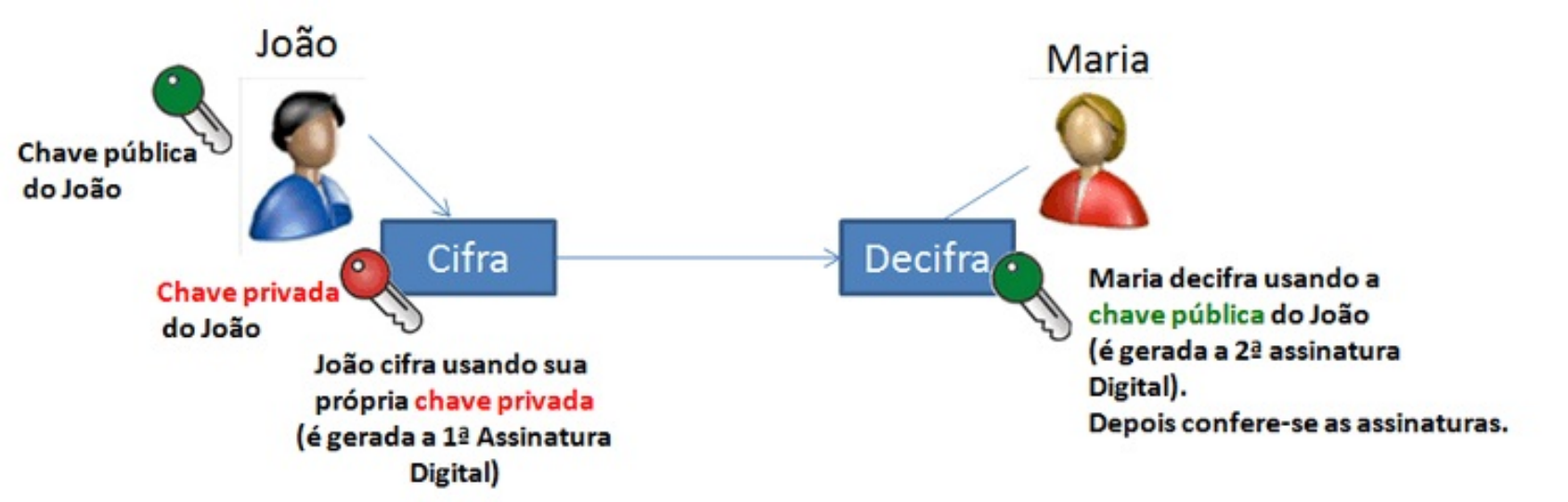


Figura: Criptografando com chaves assimétricas[4]

[4]Imagem extraída de: http://www.rtell.com.br

Exemplos de algoritmos assimétricos populares:

1. Protocolo Diffie-Hellman
2. RSA (PKCS#1)
3. DSS (Digital Signature Standard), o qual incorpora o Algoritmo de Assinatura Digital
4. ElGamal
5. Criptosistema de Paillier
6. Criptosistema de Cramer–Shoup
7. Protocolo de acordo de chave autenticada YAK
8. Criptosistema NTRUEncrypt
9. Criptosistema de McEliece

Exemplos de protocolos que usam algoritmos de chaves assimétricas:

1. S/MIME (Secure/Multipurpose Internet Mail Extension)[5]
2. PGP (Pretty Good Privacy)
3. GPG/GnuPG (GNU Privacy Guard é uma alternativa GPL ao aplicativo PGP)
4. ZRTP, um protocolo seguro VoIP
5. SSL (Secure Socket Layer)
6. TLS (Transport Layer Security)
7. SSH (Secure Shell)
8. Bitcoin
9. SILC (Secure Internet Live Conferencing)
10. OTR (Off-the-Record Messaging)

[5] Em 14 de Maio de 2018, pesquisadores anunciaram o EFAIL. Vulnerabilidade nas tecnologias de criptografia de ponta a ponta OpenPGP e S/MIME que vazam o texto puro de e-mails criptografados.

Esquemas Híbridos

A maioria dos protocolos de hoje usam esquemas híbridos, ou seja, usam os dois esquemas:

- **Chaves Simétricas:** Usada, por exemplo, para encriptação e autenticação da mensagem.
- **Chaves Assimétricas:** Usada, por exemplo, para a troca de chaves e assinatura digital.

Esquemas Híbridos

A maioria dos protocolos de hoje usam esquemas híbridos, ou seja, usam os dois esquemas:

- **Chaves Simétricas:** Usada, por exemplo, para encriptação e autenticação da mensagem.
- **Chaves Assimétricas:** Usada, por exemplo, para a troca de chaves e assinatura digital.

Assinatura Digital

É um método de autenticação de informação digital tipicamente tratada como substituta à assinatura física, já que elimina a necessidade de ter uma versão em papel do documento que necessita ser assinado.

Propriedades da Assinatura Digital:

- **Autenticidade:** o receptor deve poder confirmar que a assinatura foi feita pelo emissor.
- **Integridade:** qualquer alteração da mensagem faz com que a assinatura não corresponda mais ao documento.
- **Irretratabilidade ou não-repúdio:** o emissor não pode negar a autenticidade da mensagem.

Assinatura Digital

É um método de autenticação de informação digital tipicamente tratada como substituta à assinatura física, já que elimina a necessidade de ter uma versão em papel do documento que necessita ser assinado.

Propriedades da Assinatura Digital:

- **Autenticidade:** o receptor deve poder confirmar que a assinatura foi feita pelo emissor.
- **Integridade:** qualquer alteração da mensagem faz com que a assinatura não corresponda mais ao documento.
- **Irretratabilidade ou não-repúdio:** o emissor não pode negar a autenticidade da mensagem.

Assinatura Digital

É um método de autenticação de informação digital tipicamente tratada como substituta à assinatura física, já que elimina a necessidade de ter uma versão em papel do documento que necessita ser assinado.

Propriedades da Assinatura Digital:

- **Autenticidade:** o receptor deve poder confirmar que a assinatura foi feita pelo emissor.
- **Integridade:** qualquer alteração da mensagem faz com que a assinatura não corresponda mais ao documento.
- **Irretratabilidade ou não-repúdio:** o emissor não pode negar a autenticidade da mensagem.

Assinatura Digital

É um método de autenticação de informação digital tipicamente tratada como substituta à assinatura física, já que elimina a necessidade de ter uma versão em papel do documento que necessita ser assinado.

Propriedades da Assinatura Digital:

- **Autenticidade:** o receptor deve poder confirmar que a assinatura foi feita pelo emissor.
- **Integridade:** qualquer alteração da mensagem faz com que a assinatura não corresponda mais ao documento.
- **Irretratabilidade ou não-repúdio:** o emissor não pode negar a autenticidade da mensagem.

Parte 2
Vamos para a prática

Vamos para a Prática

Critografia no Linux e Windows

1. Criando Hash SHA-1, SHA-256, MD5 de arquivos
2. Esteganografia
3. Usando criptografia simétrica
    1. Vim
    2. WinRAR
    3. Zip
    4. Word
4. Quebrando chaves simétricas
5. Usando criptografia assimétrica
    1. Gerando uma chave GPG
    2. Assinatura Digital de Arquivo
    3. Assinatura Digital de E-mail
    4. Criptografando E-mail
    5. Chave GPG para Login em SSH
6. Atacando a criptografia assimétrica

Critografia no Linux e Windows

1. Criando Hash SHA-1, SHA-256, MD5 de arquivos
2. Esteganografia
3. Usando criptografia simétrica
    1. Vim
    2. WinRAR
    3. Zip
    4. Word
4. Quebrando chaves simétricas
5. Usando criptografia assimétrica
    1. Gerando uma chave GPG
    2. Assinatura Digital de Arquivo
    3. Assinatura Digital de E-mail
    4. Criptografando E-mail
    5. Chave GPG para Login em SSH
6. Atacando a criptografia assimétrica

Critografia no Linux e Windows

1. Criando Hash SHA-1, SHA-256, MD5 de arquivos
2. Esteganografia
3. Usando criptografia simétrica
   1. Vim
   2. WinRAR
   3. Zip
   4. Word
4. Quebrando chaves simétricas
5. Usando criptografia assimétrica
   1. Gerando uma chave GPG
   2. Assinatura Digital de Arquivo
   3. Assinatura Digital de E-mail
   4. Criptografando E-mail
   5. Chave GPG para Login em SSH
6. Atacando a criptografia assimétrica

Critografia no Linux e Windows

1. Criando Hash SHA-1, SHA-256, MD5 de arquivos
2. Esteganografia
3. Usando criptografia simétrica
   1. Vim
   2. WinRAR
   3. Zip
   4. Word
4. Quebrando chaves simétricas
5. Usando criptografia assimétrica
   1. Gerando uma chave GPG
   2. Assinatura Digital de Arquivo
   3. Assinatura Digital de E-mail
   4. Criptografando E-mail
   5. Chave GPG para Login em SSH
6. Atacando a criptografia assimétrica

Critografia no Linux e Windows

1. Criando Hash SHA-1, SHA-256, MD5 de arquivos
2. Esteganografia
3. Usando criptografia simétrica
   1. Vim
   2. WinRAR
   3. Zip
   4. Word
4. Quebrando chaves simétricas
5. Usando criptografia assimétrica
   1. Gerando uma chave GPG
   2. Assinatura Digital de Arquivo
   3. Assinatura Digital de E-mail
   4. Criptografando E-mail
   5. Chave GPG para Login em SSH
6. Atacando a criptografia assimétrica

Critografia no Linux e Windows

1. Criando Hash SHA-1, SHA-256, MD5 de arquivos
2. Esteganografia
3. Usando criptografia simétrica
    1. Vim
    2. WinRAR
    3. Zip
    4. Word
4. Quebrando chaves simétricas
5. Usando criptografia assimétrica
    1. Gerando uma chave GPG
    2. Assinatura Digital de Arquivo
    3. Assinatura Digital de E-mail
    4. Criptografando E-mail
    5. Chave GPG para Login em SSH
6. Atacando a criptografia assimétrica

Critografia no Linux e Windows

1. Criando Hash SHA-1, SHA-256, MD5 de arquivos
2. Esteganografia
3. Usando criptografia simétrica
   1. Vim
   2. WinRAR
   3. Zip
   4. Word
4. Quebrando chaves simétricas
5. Usando criptografia assimétrica
   1. Gerando uma chave GPG
   2. Assinatura Digital de Arquivo
   3. Assinatura Digital de E-mail
   4. Criptografando E-mail
   5. Chave GPG para Login em SSH
6. Atacando a criptografia assimétrica

Critografia no Linux e Windows

1. Criando Hash SHA-1, SHA-256, MD5 de arquivos
2. Esteganografia
3. Usando criptografia simétrica
    1. Vim
    2. WinRAR
    3. Zip
    4. Word
4. Quebrando chaves simétricas
5. Usando criptografia assimétrica
    1. Gerando uma chave GPG
    2. Assinatura Digital de Arquivo
    3. Assinatura Digital de E-mail
    4. Criptografando E-mail
    5. Chave GPG para Login em SSH
6. Atacando a criptografia assimétrica

Critografia no Linux e Windows

1. Criando Hash SHA-1, SHA-256, MD5 de arquivos
2. Esteganografia
3. Usando criptografia simétrica
    1. Vim
    2. WinRAR
    3. Zip
    4. Word
4. Quebrando chaves simétricas
5. Usando criptografia assimétrica
    1. Gerando uma chave GPG
    2. Assinatura Digital de Arquivo
    3. Assinatura Digital de E-mail
    4. Criptografando E-mail
    5. Chave GPG para Login em SSH
6. Atacando a criptografia assimétrica

Critografia no Linux e Windows

1. Criando Hash SHA-1, SHA-256, MD5 de arquivos
2. Esteganografia
3. Usando criptografia simétrica
    1. Vim
    2. WinRAR
    3. Zip
    4. Word
4. Quebrando chaves simétricas
5. Usando criptografia assimétrica
    1. Gerando uma chave GPG
    2. Assinatura Digital de Arquivo
    3. Assinatura Digital de E-mail
    4. Criptografando E-mail
    5. Chave GPG para Login em SSH
6. Atacando a criptografia assimétrica

Critografia no Linux e Windows

1. Criando Hash SHA-1, SHA-256, MD5 de arquivos
2. Esteganografia
3. Usando criptografia simétrica
    1. Vim
    2. WinRAR
    3. Zip
    4. Word
4. Quebrando chaves simétricas
5. Usando criptografia assimétrica
    1. Gerando uma chave GPG
    2. Assinatura Digital de Arquivo
    3. Assinatura Digital de E-mail
    4. Criptografando E-mail
    5. Chave GPG para Login em SSH
6. Atacando a criptografia assimétrica

Critografia no Linux e Windows

1. Criando Hash SHA-1, SHA-256, MD5 de arquivos
2. Esteganografia
3. Usando criptografia simétrica
    1. Vim
    2. WinRAR
    3. Zip
    4. Word
4. Quebrando chaves simétricas
5. Usando criptografia assimétrica
    1. Gerando uma chave GPG
    2. Assinatura Digital de Arquivo
    3. Assinatura Digital de E-mail
    4. Criptografando E-mail
    5. Chave GPG para Login em SSH
6. Atacando a criptografia assimétrica

Critografia no Linux e Windows

1. Criando Hash SHA-1, SHA-256, MD5 de arquivos
2. Esteganografia
3. Usando criptografia simétrica
    1. Vim
    2. WinRAR
    3. Zip
    4. Word
4. Quebrando chaves simétricas
5. Usando criptografia assimétrica
    1. Gerando uma chave GPG
    2. Assinatura Digital de Arquivo
    3. Assinatura Digital de E-mail
    4. Criptografando E-mail
    5. Chave GPG para Login em SSH
6. Atacando a criptografia assimétrica

Critografia no Linux e Windows

1. Criando Hash SHA-1, SHA-256, MD5 de arquivos
2. Esteganografia
3. Usando criptografia simétrica
    1. Vim
    2. WinRAR
    3. Zip
    4. Word
4. Quebrando chaves simétricas
5. Usando criptografia assimétrica
    1. Gerando uma chave GPG
    2. Assinatura Digital de Arquivo
    3. Assinatura Digital de E-mail
    4. Criptografando E-mail
    5. Chave GPG para Login em SSH
6. Atacando a criptografia assimétrica

Lições Aprendidas

1. Nunca, jamais, desenvolva o seu próprio algoritmo de criptografia, a menos que você tenha uma equipe de experientes criptoanalistas verificando o seu projeto.
2. Não utilize algoritmos de criptografia não comprovados ou protocolos não comprovados.
3. Os atacantes vão sempre olhar para o ponto mais fraco de um sistema de criptografia. Por exemplo, um grande espaço de chaves por si só não é garantia de uma cifra segura; a cifra ainda pode estar vulnerável contra ataques analíticos.
4. Os algoritmos criptográficos podem ser fortes, mas as implementações sempre terão falhas. Mantenha seu software atualizado.

Lições Aprendidas

1. Nunca, jamais, desenvolva o seu próprio algoritmo de criptografia, a menos que você tenha uma equipe de experientes criptoanalistas verificando o seu projeto.
2. Não utilize algoritmos de criptografia não comprovados ou protocolos não comprovados.
3. Os atacantes vão sempre olhar para o ponto mais fraco de um sistema de criptografia. Por exemplo, um grande espaço de chaves por si só não é garantia de uma cifra segura; a cifra ainda pode estar vulnerável contra ataques analíticos.
4. Os algoritmos criptográficos podem ser fortes, mas as implementações sempre terão falhas. Mantenha seu software atualizado.

Lições Aprendidas

1. Nunca, jamais, desenvolva o seu próprio algoritmo de criptografia, a menos que você tenha uma equipe de experientes criptoanalistas verificando o seu projeto.
2. Não utilize algoritmos de criptografia não comprovados ou protocolos não comprovados.
3. Os atacantes vão sempre olhar para o ponto mais fraco de um sistema de criptografia. Por exemplo, um grande espaço de chaves por si só não é garantia de uma cifra segura; a cifra ainda pode estar vulnerável contra ataques analíticos.
4. Os algoritmos criptográficos podem ser fortes, mas as implementações sempre terão falhas. Mantenha seu software atualizado.

Lições Aprendidas

1. Nunca, jamais, desenvolva o seu próprio algoritmo de criptografia, a menos que você tenha uma equipe de experientes criptoanalistas verificando o seu projeto.
2. Não utilize algoritmos de criptografia não comprovados ou protocolos não comprovados.
3. Os atacantes vão sempre olhar para o ponto mais fraco de um sistema de criptografia. Por exemplo, um grande espaço de chaves por si só não é garantia de uma cifra segura; a cifra ainda pode estar vulnerável contra ataques analíticos.
4. Os algoritmos criptográficos podem ser fortes, mas as implementações sempre terão falhas. Mantenha seu software atualizado.

Lições Aprendidas

5. Comprimentos de chave de algoritmos simétricos, a fim de frustrar ataques de busca exaustiva da chave:
   - 64 bits: inseguro exceto para dados cujo uso se dará num prazo muito curto.
   - 128 bits: segurança de longo prazo para várias décadas, a menos que os computadores quânticos se tornem disponíveis.
   - 256 bits: como acima, mas *provavelmente* seguros até contra ataques por computadores quânticos.

Lições Aprendidas

5. Comprimentos de chave de algoritmos simétricos, a fim de frustrar ataques de busca exaustiva da chave:
   - 64 bits: inseguro exceto para dados cujo uso se dará num prazo muito curto.
   - 128 bits: segurança de longo prazo para várias décadas, a menos que os computadores quânticos se tornem disponíveis.
   - 256 bits: como acima, mas *provavelmente* seguros até contra ataques por computadores quânticos.

Lições Aprendidas

5. Comprimentos de chave de algoritmos simétricos, a fim de frustrar ataques de busca exaustiva da chave:
   - 64 bits: inseguro exceto para dados cujo uso se dará num prazo muito curto.
   - 128 bits: segurança de longo prazo para várias décadas, a menos que os computadores quânticos se tornem disponíveis.
   - 256 bits: como acima, mas *provavelmente* seguros até contra ataques por computadores quânticos.

Lições Aprendidas

5. Comprimentos de chave de algoritmos simétricos, a fim de frustrar ataques de busca exaustiva da chave:
   - 64 bits: inseguro exceto para dados cujo uso se dará num prazo muito curto.
   - 128 bits: segurança de longo prazo para várias décadas, a menos que os computadores quânticos se tornem disponíveis.
   - 256 bits: como acima, mas *provavelmente* seguros até contra ataques por computadores quânticos.

Bibliografia

[1] MORGADO, PITOMBEIRA, CARVALHO, Combinatória e Probabilidade IMPA, 1991.
[2] SGARRO, A. Códigos Secretos: Criptografia. Editora Melhoramentos: São Paulo,1989.
[3] Tom Apostol. Introduction to analytic number theory . Springer, 1976.
[4] Sanjeev Arora e Boaz Barak. Computational Complexity: A Modern Approach . Cambridge University Press, 2009.
[5] Kenneth J. Arrow. A Dificulty in the Concept of Social Welfare. Em: Journal of Political Economy 58.4 (1950), pag. 328-346.
[6] Michael Artin. Algebra . Prentice Hall, 1991.
[7] Giuseppe Ateniese et al. Constructions and Bounds for Visual Cryptography. Em: 23rd International Colloquium on Automata, Languages and Programming . 1996, pag. 416-428.
SSL Labs: `https://www.ssllabs.com/`
GTA UFRJ: `https://www.gta.ufrj.br/`
Eff.org: `https://ssd.eff.org/pt-br/module/como-fazer-usar-otr-em-linux`
WPA2 KRACK: `https://www.krackattacks.com/`
EFAIL: `https://efail.de/`
ProtonMail: `https://protonmail.com/blog/pgp-vulnerability-efail/`
TLS/SSL: `https://www.gracefulsecurity.com/tls-ssl-vulnerabilities/`
OBMep: `http://www.obmep.org.br`