

Introdução à criptografia

SEG786203 – CST em Análise e Desenvolvimento de Sistemas

Prof. Emerson Ribeiro de Mello

mello@ifsc.edu.br

Licenciamento



Slides licenciados sob [Creative Commons "Atribuição 4.0 Internacional"](https://creativecommons.org/licenses/by/4.0/)

Sumário

- 1 Introdução e criptografia clássica
- 2 Criptografia moderna
- 3 Comunicação segura entre partes
- 4 Função de dispersão criptográfica

Introdução e criptografia clássica

Objetivos dessa aula

- Entender criptografia moderna requer aprofundamento em conceitos matemáticos
- Para esse curso a criptografia será apresentada como uma ferramenta de suporte à segurança da informação
- Visão geral dos principais conceitos e técnicas de criptografia que são relevantes para a segurança da informação

Propriedades básicas de segurança

Confidentiality, Integrity, Availability (CIA triad)



Fonte: Operational Technology

Cybersecurity for Energy Systems

■ Confidencialidade

- A informação só deve ser revelada para usuários autorizados

■ Integridade

- A informação não deve ser alterada, destruída ou perdida, de maneira não autorizada ou acidental

■ Disponibilidade

- A informação deve estar disponível quando necessário

É necessário garantir as três propriedades para que a informação seja considerada segura

Problemas

- 1 Salvar suas senhas pessoais em um arquivo no seu computador e quer garantir que ninguém mais tenha acesso
- 2 Enviar um arquivo por email para um amigo e você e seu amigo terem certeza de que o arquivo não foi alterado por um terceiro
- 3 Salvar as senhas dos usuários do seu sistema em um banco de dados

Problemas

- 1 Salvar suas senhas pessoais em um arquivo no seu computador e quer garantir que ninguém mais tenha acesso
- 2 Enviar um arquivo por email para um amigo e você e seu amigo terem certeza de que o arquivo não foi alterado por um terceiro
- 3 Salvar as senhas dos usuários do seu sistema em um banco de dados

Exercício em duplas (10 minutos)

- Pense em possíveis soluções para os problemas acima
- Use o conhecimento que você já possui sobre tecnologias e ferramentas
- É importante indicar como a solução proposta garante a segurança da informação

Problema 1

Discussão

- Salvar o arquivo com um nome que não chame a atenção?

Problema 1

Discussão

- **Salvar o arquivo com um nome que não chame a atenção?**
 - Exemplo: `senhas.txt` → `notas.txt`

Problema 1

Discussão

- **Salvar o arquivo com um nome que não chame a atenção?**
 - Exemplo: `senhas.txt` → `notas.txt`
- **Criar um ZIP com senha?**

Problema 1

Discussão

- **Salvar o arquivo com um nome que não chame a atenção?**
 - Exemplo: `senhas.txt` → `notas.txt`
- **Criar um ZIP com senha?**
 - Exemplo: `zip -e senhas.zip senhas.txt`

Problema 1

Discussão

- **Salvar o arquivo com um nome que não chame a atenção?**
 - Exemplo: `senhas.txt` → `notas.txt`
- **Criar um ZIP com senha?**
 - Exemplo: `zip -e senhas.zip senhas.txt`
- **Criptografar o arquivo com uma chave simétrica?**

Problema 1

Discussão

- **Salvar o arquivo com um nome que não chame a atenção?**
 - Exemplo: `senhas.txt` → `notas.txt`
- **Criar um ZIP com senha?**
 - Exemplo: `zip -e senhas.zip senhas.txt`
- **Criptografar o arquivo com uma chave simétrica?**
 - Exemplo: `openssl aes-256-cbc -in senhas.txt -out senhas.enc -pbkdf2 -k secreta`

Problema 1

Discussão

- **Salvar o arquivo com um nome que não chame a atenção?**
 - Exemplo: `senhas.txt` → `notas.txt`
- **Criar um ZIP com senha?**
 - Exemplo: `zip -e senhas.zip senhas.txt`
- **Criptografar o arquivo com uma chave simétrica?**
 - Exemplo: `openssl aes-256-cbc -in senhas.txt -out senhas.enc -pbkdf2 -k secreta`
- **Criptografar o arquivo com uma chave assimétrica?**

Problema 1

Discussão

- **Salvar o arquivo com um nome que não chame a atenção?**
 - Exemplo: `senhas.txt` → `notas.txt`
- **Criar um ZIP com senha?**
 - Exemplo: `zip -e senhas.zip senhas.txt`
- **Criptografar o arquivo com uma chave simétrica?**
 - Exemplo: `openssl aes-256-cbc -in senhas.txt -out senhas.enc -pbkdf2 -k secreta`
- **Criptografar o arquivo com uma chave assimétrica?**
 - Exemplo: `openssl pkeyutl -encrypt -in senhas.txt -out senhas.enc -pubin -inkey chave-publica.pem`

Problema 1

Discussão

- **Salvar o arquivo com um nome que não chame a atenção?**
 - Exemplo: `senhas.txt` → `notas.txt`
- **Criar um ZIP com senha?**
 - Exemplo: `zip -e senhas.zip senhas.txt`
- **Criptografar o arquivo com uma chave simétrica?**
 - Exemplo: `openssl aes-256-cbc -in senhas.txt -out senhas.enc -pbkdf2 -k secreta`
- **Criptografar o arquivo com uma chave assimétrica?**
 - Exemplo: `openssl pkeyutl -encrypt -in senhas.txt -out senhas.enc -pubin -inkey chave-publica.pem`
- **Usar um gerenciador de senhas?**

Problema 1

Discussão

- **Salvar o arquivo com um nome que não chame a atenção?**
 - Exemplo: `senhas.txt` → `notas.txt`
- **Criar um ZIP com senha?**
 - Exemplo: `zip -e senhas.zip senhas.txt`
- **Criptografar o arquivo com uma chave simétrica?**
 - Exemplo: `openssl aes-256-cbc -in senhas.txt -out senhas.enc -pbkdf2 -k secreta`
- **Criptografar o arquivo com uma chave assimétrica?**
 - Exemplo: `openssl pkeyutl -encrypt -in senhas.txt -out senhas.enc -pubin -inkey chave-publica.pem`
- **Usar um gerenciador de senhas?**
 - Exemplo: KeePass, Bitwarden, ProtonPass, Google, Apple, etc.

Problema 2

Discussão

- Criptografar o arquivo com uma chave simétrica e enviar a chave por email?

Problema 2

Discussão

- **Criptografar o arquivo com uma chave simétrica e enviar a chave por email?**
 - É necessário garantir a confidencialidade da chave e compartilhar a chave de forma segura

Problema 2

Discussão

- **Criptografar o arquivo com uma chave simétrica e enviar a chave por email?**
 - É necessário garantir a confidencialidade da chave e compartilhar a chave de forma segura
- **Criptografar o arquivo com uma chave assimétrica e enviar a chave pública por email?**

Problema 2

Discussão

- **Criptografar o arquivo com uma chave simétrica e enviar a chave por email?**
 - É necessário garantir a confidencialidade da chave e compartilhar a chave de forma segura
- **Criptografar o arquivo com uma chave assimétrica e enviar a chave pública por email?**
 - É necessário garantir a autenticidade da chave pública e compartilhar a chave de forma segura

Problema 2

Discussão

- **Criptografar o arquivo com uma chave simétrica e enviar a chave por email?**
 - É necessário garantir a confidencialidade da chave e compartilhar a chave de forma segura
- **Criptografar o arquivo com uma chave assimétrica e enviar a chave pública por email?**
 - É necessário garantir a autenticidade da chave pública e compartilhar a chave de forma segura
- **Enviar por email o arquivo e o resumo criptográfico sobre o mesmo?**

Problema 2

Discussão

- **Criptografar o arquivo com uma chave simétrica e enviar a chave por email?**
 - É necessário garantir a confidencialidade da chave e compartilhar a chave de forma segura
- **Criptografar o arquivo com uma chave assimétrica e enviar a chave pública por email?**
 - É necessário garantir a autenticidade da chave pública e compartilhar a chave de forma segura
- **Enviar por email o arquivo e o resumo criptográfico sobre o mesmo?**
 - Exemplo: `sha256sum arquivo.txt` → `508e5e725bd1bf3bcd...`

Problema 2

Discussão

- **Criptografar o arquivo com uma chave simétrica e enviar a chave por email?**
 - É necessário garantir a confidencialidade da chave e compartilhar a chave de forma segura
- **Criptografar o arquivo com uma chave assimétrica e enviar a chave pública por email?**
 - É necessário garantir a autenticidade da chave pública e compartilhar a chave de forma segura
- **Enviar por email o arquivo e o resumo criptográfico sobre o mesmo?**
 - Exemplo: `sha256sum arquivo.txt` → `508e5e725bd1bf3bcd...`
- **Compartilhar previamente um segredo com o amigo e usar o segredo para gerar um resumo criptográfico sobre o arquivo?**

Problema 2

Discussão

- **Criptografar o arquivo com uma chave simétrica e enviar a chave por email?**
 - É necessário garantir a confidencialidade da chave e compartilhar a chave de forma segura
- **Criptografar o arquivo com uma chave assimétrica e enviar a chave pública por email?**
 - É necessário garantir a autenticidade da chave pública e compartilhar a chave de forma segura
- **Enviar por email o arquivo e o resumo criptográfico sobre o mesmo?**
 - Exemplo: `sha256sum arquivo.txt` → `508e5e725bd1bf3bcd...`
- **Compartilhar previamente um segredo com o amigo e usar o segredo para gerar um resumo criptográfico sobre o arquivo?**
 - Exemplo: `openssl dgst -sha256 -hmac segredo arquivo.txt`

Problema 3

Discussão

- Salvar as senhas em texto claro no banco de dados?

Problema 3

Discussão

- **Salvar as senhas em texto claro no banco de dados?**
 - Se o banco de dados for comprometido, as senhas estarão expostas

Problema 3

Discussão

- **Salvar as senhas em texto claro no banco de dados?**
 - Se o banco de dados for comprometido, as senhas estarão expostas
- **Salvar as senhas criptografadas no banco de dados?**

Problema 3

Discussão

- **Salvar as senhas em texto claro no banco de dados?**
 - Se o banco de dados for comprometido, as senhas estarão expostas
- **Salvar as senhas criptografadas no banco de dados?**
 - Se o banco de dados for comprometido, as senhas estarão seguras

Problema 3

Discussão

- **Salvar as senhas em texto claro no banco de dados?**
 - Se o banco de dados for comprometido, as senhas estarão expostas
- **Salvar as senhas criptografadas no banco de dados?**
 - Se o banco de dados for comprometido, as senhas estarão seguras
 - Mas é necessário garantir a confidencialidade da chave criptográfica

Problema 3

Discussão

- **Salvar as senhas em texto claro no banco de dados?**
 - Se o banco de dados for comprometido, as senhas estarão expostas
- **Salvar as senhas criptografadas no banco de dados?**
 - Se o banco de dados for comprometido, as senhas estarão seguras
 - Mas é necessário garantir a confidencialidade da chave criptográfica
- **Salvar resumos criptográficos das senhas no banco de dados?**

Problema 3

Discussão

- **Salvar as senhas em texto claro no banco de dados?**
 - Se o banco de dados for comprometido, as senhas estarão expostas
- **Salvar as senhas criptografadas no banco de dados?**
 - Se o banco de dados for comprometido, as senhas estarão seguras
 - Mas é necessário garantir a confidencialidade da chave criptográfica
- **Salvar resumos criptográficos das senhas no banco de dados?**
 - Se o banco de dados for comprometido, as senhas estarão seguras

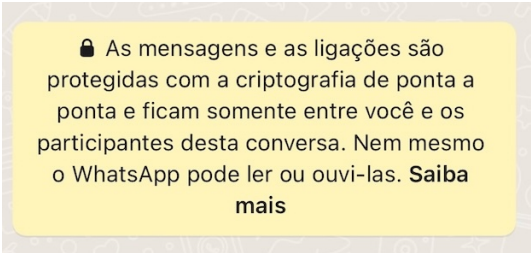
Problema 3

Discussão

- **Salvar as senhas em texto claro no banco de dados?**
 - Se o banco de dados for comprometido, as senhas estarão expostas
- **Salvar as senhas criptografadas no banco de dados?**
 - Se o banco de dados for comprometido, as senhas estarão seguras
 - Mas é necessário garantir a confidencialidade da chave criptográfica
- **Salvar resumos criptográficos das senhas no banco de dados?**
 - Se o banco de dados for comprometido, as senhas estarão seguras
 - Mas é necessário uso de funções de dispersão criptográfica seguras e *salt* aleatório para cada senha

Criptografia

- A palavra “criptografia” é oriunda do grego e significa “escrita secreta”
- Consiste na prática e no estudo de técnicas para comunicação segura na presença de adversários
- Pode ser usada para garantir a confidencialidade, a autenticidade e a integridade da informação

A screenshot of a WhatsApp security notice. It features a yellow rounded rectangle with a lock icon on the left. The text inside explains that messages and calls are protected with end-to-end encryption, meaning only the participants can read or hear them, and even WhatsApp cannot access them. The word 'Saiba' is in bold, followed by 'mais' on a new line.

🔒 As mensagens e as ligações são protegidas com a criptografia de ponta a ponta e ficam somente entre você e os participantes desta conversa. Nem mesmo o WhatsApp pode ler ou ouvi-las. **Saiba mais**

Fonte: Captura de tela do aplicativo WhatsApp

Criptosistema

Componente básico da criptografia, consiste em uma tupla de cinco elementos: $(\mathcal{M}, \mathcal{K}, \mathcal{C}, \mathcal{E}, \mathcal{D})$

- \mathcal{M} é o conjunto de mensagens em texto claro
- \mathcal{K} é o conjunto de chaves
- \mathcal{C} é o conjunto de mensagens cifradas
- \mathcal{E} é o algoritmo para cifrar, $\mathcal{E} : \mathcal{M} \times \mathcal{K} \rightarrow \mathcal{C}$
- \mathcal{D} é o algoritmo para decifrar, $\mathcal{D} : \mathcal{C} \times \mathcal{K} \rightarrow \mathcal{M}$

Criptanálise

Técnicas para quebrar criptosistemas com o objetivo de recuperar a mensagem original sem conhecer a chave ou o algoritmo de cifragem

- **Ataque com o texto cifrado**

- adversário possui o texto cifrado e através dele tenta obter o texto em claro ou até mesmo a chave

- **Ataque com o texto em claro**

- adversário possui o texto cifrado e o texto em claro tendo o objetivo de descobrir a chave

Criptografia clássica

- Na antiguidade a criptografia tinha como **objetivo principal** garantir a **confidencialidade** das mensagens
- A chave era compartilhada entre o emissor e o receptor
- A segurança do sistema dependia do sigilo da chave e do sigilo do algoritmo
- **Exemplos**
 - Cifra de substituição
 - Cifra de transposição

Criptografia clássica

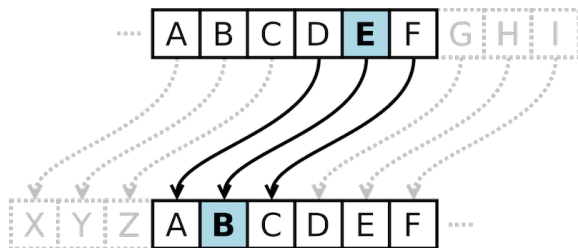
Cifra de substituição

- Os símbolos do texto claro são substituídos por outros símbolos
- Podem ser monoalfabéticas ou polialfabéticas
- Exemplos
 - Cifra de César
 - Cifra de Vigenère
 - Cifra de Enigma

Cifra de César

Exemplo de cifra de substituição monoalfabética

- Uma das técnicas mais simples para cifrar mensagens que foi usada por Júlio César na Roma antiga
- Usa um alfabeto deslocado de n posições para cifrar o texto claro



Fonte: Wikipedia

Chave:	3 para esquerda
Texto claro:	IFSC
Texto cifrado:	FCPZ

$$\mathcal{E}_n(x) = (x - n) \mod 26 \quad (1)$$

$$\mathcal{D}_n(x) = (x + n) \mod 26 \quad (2)$$

, sendo x a posição da letra no alfabeto e n o deslocamento

Quebra de cifras de substituição monoalfabética

- A cifra de César é facilmente quebrada por um ataque de força bruta
 - Existem apenas 25 chaves possíveis (deslocamentos de 1 a 25)
- Análise de frequência de letras é uma técnica comum para quebrar cifras de substituição monoalfabética
 - Em português a letra mais frequentemente usada é a vogal “a” (14,63%)¹

¹ https://pt.wikipedia.org/wiki/Alfabeto_portugu%C3%AAs

Cifra de Vigenère

Exemplo de cifra de substituição polialfabética

- Usa uma palavra-chave para cifrar o texto em claro
- A palavra-chave é repetida até que tenha o mesmo tamanho do texto em claro
- A cifragem é feita letra a letra usando a cifra de César e a letra correspondente da palavra-chave

Chave: IFSC

Palavra-chave: IFSCIFSCI FSC

Texto em claro: SEGURANCA ADS

Texto cifrado: AJYWZFFEI FVU

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

Enigma

- Máquina eletromecânica com rotores que faz uso de cifra polialfabética
- Patenteada por Arthur Scherbius em 1918 e utilizada pela Alemanha nazista durante a Segunda Guerra Mundial (versão militar em 1928)
- Em 1933 o código foi quebrado por matemáticos poloneses
- Em 1940, Alan Turing e sua equipe conseguiram acelerar a quebra do código



Fonte: Acervo pessoal

Criptografia clássica

Cifras de transposição

- Os símbolos do texto claro são rearranjados (permutados), ou seja, a ordem dos símbolos é alterada
- Exemplos
 - Cítala
 - *Rail fence*



Cítala usada pelos espartanos

Fonte: <https://en.wikipedia.org/wiki/Scytale>



Uma cerca com 3 trilhos (*rail fence*)

Fonte: <https://rocketfence.com/>

Cifras de transposição

Exemplo: *rail fence*

- O texto em claro é escrito em uma diagonal vertical em trilhos alternados de uma cerca que ao atingir o último trilho inverte a direção
 - Chave = número de trilhos (linhas)
 - Comprimento da cerca = número de letras do texto em claro
- O texto cifrado é obtido lendo-se as linhas individuais e concatenando-as
 - Escreve-se cada letra do texto em claro em linhas alternadas e depois lê-se em linhas

S				R				A
	E		U		A		Ç	
		G			N			

Mensagem: SEGURANÇA

chave: 3

Texto cifrado: SRAEUAÇGN

Criptografia moderna

Criptografia moderna

- Computadores e redes de computadores permitiram o desenvolvimento de novos algoritmos e protocolos
- Além **confidencialidade**, a criptografia moderna tem como objetivo **garantir a autenticidade e a integridade** das mensagens
 - Baseada em problemas matemáticos difíceis de resolver
 - Algoritmos são públicos e a segurança é garantida pela dificuldade de resolver esses problemas
- Criptosistemas
 - Simétricos – mesma chave para cifrar e decifrar
 - Assimétricos – chaves diferentes para cifrar e decifrar

Aplicações de criptosistema simétrico

■ Comunicação segura entre duas partes

- Garante a confidencialidade, a autenticidade e a integridade das mensagens trocadas entre duas partes mesmo sobre um canal inseguro
- A chave precisa ser compartilhada entre as partes previamente

■ Armazenamento seguro de dados

- Arquivo individual, disco rígido ou partição, banco de dados
- Exige que a chave secreta seja armazenada de forma segura

Aplicações de criptosistema assimétrico

■ Troca de chaves de sessão

- Criptografia simétrica é mais eficiente que a assimétrica
- Criptografia assimétrica é usada para trocar chaves simétricas

■ Comunicação segura entre duas partes

- E-mail seguro (PGP, S/MIME)
- Comunicação segura em redes sociais

■ Assinatura eletrônica de documentos

- Contratos, documentos fiscais, etc.

■ Certificados digitais

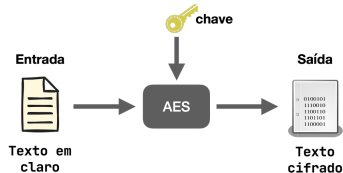
- Parte da Infraestrutura de Chaves Públicas (ICP), usados para autenticar a identidade de uma entidade

Criptosistema simétrico

- Exemplos de algoritmos simétricos: AES, DES, 3DES, RC4, Twofish

```
# Gerando uma chave simétrica de 256 bits com OpenSSL e algoritmo AES
openssl aes-256-cbc -P -pbkdf2

# Saída
salt=8E9A5DAD3B08F20A
key=D51DF6FB2ED1674D31C49EC33A61634824312E17F4888D99B07DE27F27182C64
iv =D4CC1075948FA964FF87331032AE29E4
```



```
# Cifrar um arquivo com OpenSSL, algoritmo AES e codificado em base64
openssl aes-256-cbc -in arquivo.txt -out arquivo.enc -pbkdf2 -salt -a

# Decifrar o arquivo cifrado acima
openssl aes-256-cbc -d -in arquivo.enc -out saida.txt -pbkdf2 -a

# Cifrar um arquivo com OpenSSL, algoritmo AES e chave e IV definidos
openssl aes-256-cbc -in arquivo.txt -out arquivo.enc -S 8E9A5DAD3B08F20A -iv D4CC1075948FA964FF87331032AE29E4
-K D51DF6FB2ED1674D31C49EC33A61634824312E17F4888D99B07DE27F27182C64
```

Criptosistema assimétrico

- Uso de um par de chaves complementares: pública e privada
- Informação cifrada com a chave pública só pode ser decifrada com a chave privada correspondente
- Exemplos de algoritmos assimétricos: RSA, ElGamal, ECC

```
# Gerando um par de chaves RSA com OpenSSL de 2048 bits
openssl genrsa -out chave-privada.pem 2048
openssl rsa -in chave-privada.pem -pubout -out chave-publica.pem

# Gerando um par de chaves ECC com OpenSSL
openssl ecparam -name prime256v1 -genkey -noout -out chave-privada.pem
openssl ec -in chave-privada.pem -pubout -out chave-publica.pem

# Cifrar um arquivo com OpenSSL, algoritmo RSA e chave pública
openssl pkeyutl -encrypt -in arquivo.txt -out arquivo.enc -pubin -inkey chave-publica.pem

# Decifrar o arquivo cifrado acima
openssl pkeyutl -decrypt -in arquivo.enc -out saida.txt -inkey chave-privada.pem
```

Par de chaves RSA de 2048 bits

-----BEGIN PRIVATE KEY-----

MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBCwggSjAgEAAoIBAQD1QHK3p5
BWQnILZ3d6/0WVBt3bEb/SahB07jxvff0yEW1d687uuv9IPeoXr3z0Jbgs
ML1WwBiFG1X0vJorhtD0yxKemHy4c04dgS0JaZmYVq9fEa+e6CUDF6AKAs
OJKycDUGdJw59KmPrYuYAiUfZpImEs7JKVzFVeaYIIfJe2i7DxBJqZsRMf
GoRSMRilvKpu2E3Q6tEG+e3tkG2S0Ee1q/NY57RiYt9Kz41HyRNhq5Y6X2
WZDUeJXp3AyLWD2rhh4drMXoosnhmSUhIrVQIFBxJstpx0jDD06TS56dBQ
WEc/SegYAszS4z1sGVZNdXcW6Vd80y6cMDmlpxSB09MVAgMBAAEcggEAEZ
d5dhhYnhXpNm+ThPaAic5uFdJt2tKiv7v0TWvokv/9WXiv32fgeQYXQYWi
1Z9TW6i88bv55bTsSqRyXCazPTseCad/4pNf4km3lWLSE3uLZ4su483CkV
Mtx7bglFUIYFu6IVHulkXY8eqCJ6AMfZLSzCS3NuGAJ4eJ4xrUdYBrfENZ
/WmUJpkWuY2B60HU2x58SgCKZy649megp63RcQCu019uzmJnjq1ipX0wU
4ES331DFdbjJkZhL9T3wRyH+5mMv3S6UnCHOL+Xdkx19T1bVedHWCf7pMZ
bQ4ORZJkRsNAHoo+2fYccuQbULaIH02JOIKu5d97MGWRxKi6PQKBgQDz1J
r7UWVdespScGKG6ksGP24X7b+hIMYe8nGRnhqGwaQDcT8670JIwMDmSORs
0+KQnqkSQCoOPvm0sz1/MvPP7AOWgKha3nY80PWK9Sog2/Ng+b8W+qvEnk
H3JuLLYYncVsLy0VrRm9uYy1b9CTdxQOE21wUCIHUoDyU441vU4wkBQDw
sZJiT37KC7CcYG3UFaTnfaU+nFzamif5Gb1BpsNn08s726I7A2ebE0ecWp
CYirEQ7yHcHPOS4WvX9M/Z6AwlroFb5UpFi4NE6EvrmpL1EBsCoH1A9ofV
jT/ASjWHksmj6ez6LGvx1oQvphNt8W2Tgy0bzpohQYb9bz8csTmxpwKBgH
no38At9y9g2E2VtT2BYRdysVunbDtkhoosj6nN0ddAdGegNIwCqjTT4t6B
3WlIGyxR15jfafqLTf0SEvpJAGwBxd124DXmqlj75ZCfeXvKXQosdGVJ2Z
2Vvy+SSqyDTQ1Ue84p7G2GrqnbaNfExuqiFLh1kzR4A/u0bv7c9isfAoGB
AKzGithHPPrA6W/ikr7ICJ3GUm72p+G1jIzaI53XtA29jWKwLKRu7iG+1EO
6FJC6bwCbrwV6D2cSh1AwdS+abyxH3hhw1J29sijo41eb1+m+3s+vaza94
SGFfF1gs70f40o27KcUbhhD/b7vUdzYAJCvLieHGhGhuu0fuyk5dBEOZAO
GAXApi9ae5bZSHepys9H3kMSKcla4f6NgsBoDxytow1HOQrQwny4SZphwM
7DFZR45kX9tPa6jqPqQKEFmcMd88LA6SHepHjt4d0fMtLJdEfDqZYUD0Enm
uV2Ef5gmI47KWHzx0ExH1IxEIwQfmvQnP2Qqm8klx76Dc0Uu5gLi2pGk8=
-----END PRIVATE KEY-----

-----BEGIN PUBLIC KEY-----

MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAA5Ubyt6eQVkJyJW
d3ev9F1Qbd2xG/0moQTu48b339MhFtXev07rr/SD3qF6989CW4LDCyFsAY
hRtV9LyaK4bQzssSnph8uHDuHYEjiWmZmFavXxGvnuglAxegCgLDiSsnA1
BnSc0fSpj62LmAlIH2aSJhL0ySlcxVXmmmCHyXtouw8QsambETHxqEUjES
JbyqbthN00rRBvnt7ZBtkbBHTavzW0e0YmE/Ss+JR8kTYauW019lmQ1HiV
6dwMi1g9q4W+HazF6KLJ4ZklISK1UCBQcSbLacTowwzukOuenQUFHHPOno
GALM0uM9bBlWTXV3FulXfDsunDA5pacUgaPTFTFIDAQAB
-----END PUBLIC KEY-----

Par de chaves ECC de 256 bits

Curva Elíptica Criptográfica prime256v1

■ Chave privada

```
-----BEGIN EC PRIVATE KEY-----
MHcCAQEEIEABStAz2WkyjzXHd7SknEbvZ8ZAvEZeWLisrdWXnhxxoAoGCCqGSM49
AwEHoUQDQgAEuHb4bzVS2hhqGeXNkm9EaHuTmU9KNBy5HuQR8klQAdZz0SzRpy7q
9IsezK0yckuPzzgeDR928FevebG9ot6ecA==
-----END EC PRIVATE KEY-----
```

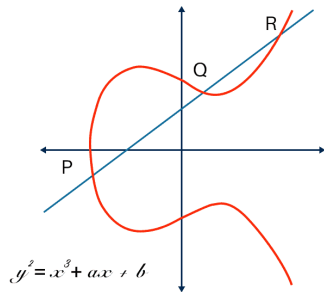
■ Chave pública

```
-----BEGIN PUBLIC KEY-----
MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEuHb4bzVS2hhqGeXNkm9EaHuTmU9K
NBy5HuQR8klQAdZz0SzRpy7q9IsezK0yckuPzzgeDR928FevebG9ot6ecA==
-----END PUBLIC KEY-----
```

Criptosistema assimétrico I

Curva Elíptica Criptográfica (ECC) vs RSA

- RSA baseia-se na dificuldade de fatorar o produto de dois números primos grandes
 - Dado dois números primos p e q , calcular $n = p \times q$ é fácil
 - Dado n é difícil encontrar p e q
- ECC baseia-se na dificuldade de resolver o problema do logaritmo discreto entre pontos em uma curva elíptica, mesmo que um dos pontos seja conhecido
- Chaves menores em ECC fornecem o mesmo nível de segurança que chaves maiores em RSA



Fonte: <https://avinetworks.com/glossary/elliptic-curve-cryptography>

Criptosistema assimétrico II

Curva Elíptica Criptográfica (ECC) vs RSA

Tabela: Comparação do tamanho das chaves para garantir o mesmo nível de segurança

Tamanho da chave (bits)			
Simétrica	RSA	ECC	
80	1.024	160	
112	2.048	224	
128	3.072	256	
192	7.680	384	
256	15.360	521	

- ECC é mais eficiente que RSA
- ECC é mais resistente a ataques quânticos
 - RSA é vulnerável a ataques baseados no algoritmo de Shor
- Até 2030, *draft* do NIST^a recomenda chaves de 256 bits para ECC e 2.048 bits para RSA

^a<https://csrc.nist.gov/pubs/sp/800/78/5/ipd>

Criptografia depende de entropia I

- A segurança de um sistema criptográfico **depende da qualidade dos números aleatórios** que são usados
 - Na geração de chaves criptográficas
 - *Salts* para senhas com *hash*
 - Número de uso único (*nonce*)
- Para algumas aplicações, **números pseudoaleatórios** são suficientes
- Para outras são necessários **números aleatórios criptograficamente seguros**

Criptografia depende de entropia II

Entropia na computação

Medida da incerteza associada a um número aleatório. Quanto maior a entropia, mais imprevisível é o número aleatório

- Boas fontes de entropia estão associadas a processos físicos imprevisíveis
 - Ruído atmosférico ou térmico
 - Variação de atraso (jitter) de relógio
 - Interação de usuário (movimento do mouse, teclas pressionadas)
 - Interrupções de hardware, como movimento de disco rígido, tráfego de rede
- Entropia é um recurso finito e pode ser esgotado, gerando um bloqueio momentâneo na execução da aplicação por falta de entropia

Gerador de números aleatórios

■ *Pseudo-Random Number Generator (PRNG)*

- Algoritmo determinístico que gera sequências de números que parecem ser aleatórios

■ *Cryptographically Secure Pseudo-Random Number Generator (CSPRNG)*

- Algoritmo que gera números que parecem ser aleatórios adequados para criptografia

■ *True Random Number Generator (TRNG)*

- Dispositivo que gera números aleatórios verdadeiros



Fonte: <https://dilbert.com/>

Gerador de números pseudoaleatórios (PRNG)

- Algoritmo, derivado de uma função matemática, que gera sequências de números que parecem ser aleatórios, mas são determinísticos
- Não são adequados para gerar chaves criptográficas

```
// Classe Random gera números pseudoaleatórios de acordo com o modelo Gerador  
// Congruencial Linear  
// https://docs.oracle.com/javase/8/docs/api/java/util/Random.html  
  
// Iniciando com a semente 123. Se não for fornecida, a semente é baseada no relógio do  
// sistema  
Random random = new Random(123);  
  
// Será gerada sempre a mesma sequência: 82, 50, 76, 89 e 95  
for (int i = 0; i < 5; i++) {  
    System.out.println(random.nextInt(100));  
}
```

Criptografia

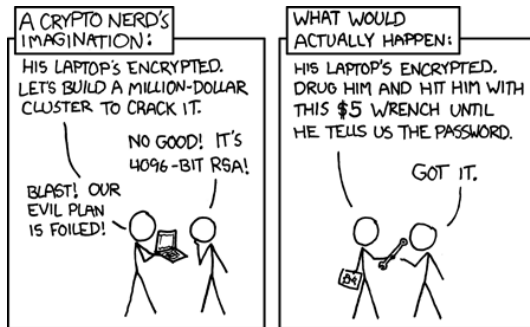
Implementações no Linux e Java

- No Linux o CSPRNG pode ser acessado via `/dev/random` com a chamada de sistema `getrandom`
- *Java Cryptography Architecture (JCA)*²
 - Conjunto de APIs para assinatura digital, criptografia simétrica e assimétrica, geração de chaves, *hash*, etc
 - Classe `SecureRandom`³ é usada para gerar números aleatórios criptograficamente seguros

²<https://docs.oracle.com/en/java/javase/22/security/java-cryptography-architecture-jca-reference-guide.html>

³<https://docs.oracle.com/javase/8/docs/api/java/security/SecureRandom.html>

Criptografia não resolve todos os problemas de segurança

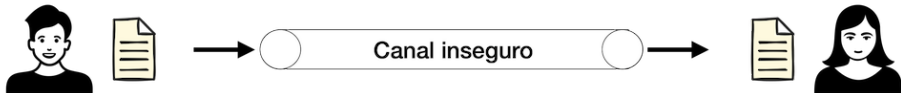


Fonte: <https://xkcd.com/538>

Comunicação segura entre partes

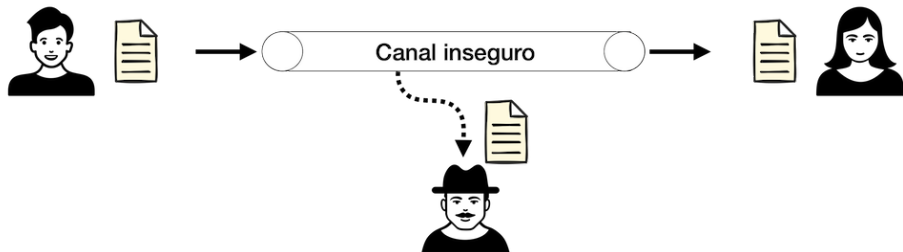
Comunicação segura entre duas partes

Criptografia simétrica



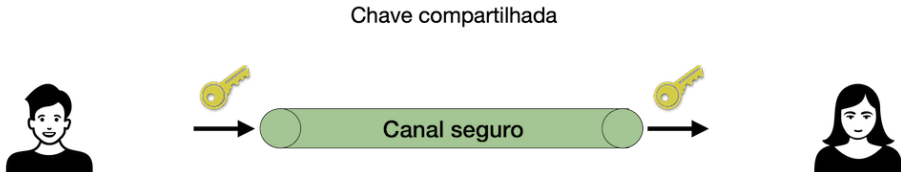
Comunicação segura entre duas partes

Criptografia simétrica



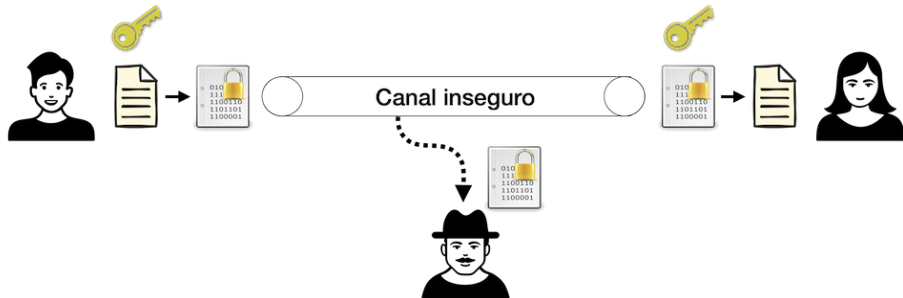
Comunicação segura entre duas partes

Criptografia simétrica



Comunicação segura entre duas partes

Criptografia simétrica



Como compartilhar a chave de forma segura?

Dificuldades com criptografia simétrica

1 Encontro presencial

- Não é escalável
- Distância, custo, tempo

2 Envio por e-mail

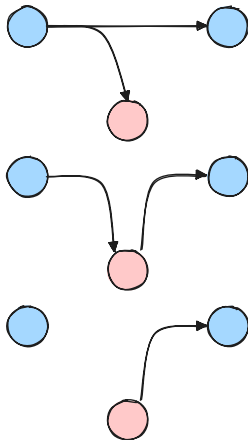
- Não é seguro
- Interceptação, MITM, personificação

3 Uso de um terceiro confiável

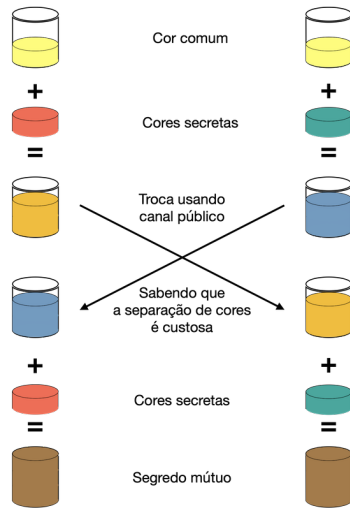
- Como estabelecer essa confiança?

4 Protocolo de acordo de chaves Diffie-Hellman

- Suscetível a ataques MITM



Protocolo de acordo de chaves Diffie-Hellman



Fonte: Adaptado de Wikipedia

Protocolo de acordo de chaves Diffie-Hellman

- Concordam com o primo $p = 23$ e como base $g = 5$

- O segredo de

- Alice é $a = 6$

- Bob é $b = 15$

- Alice calcula

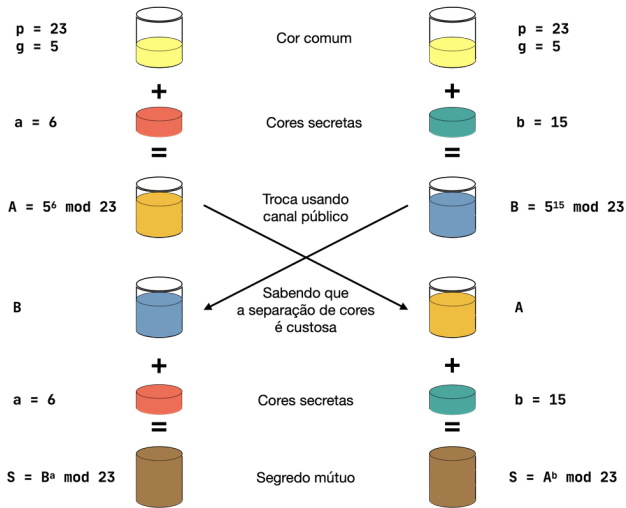
- $A = g^a \mod p$

- $s = B^a \mod p$

- Bob calcula

- $B = g^b \mod p$

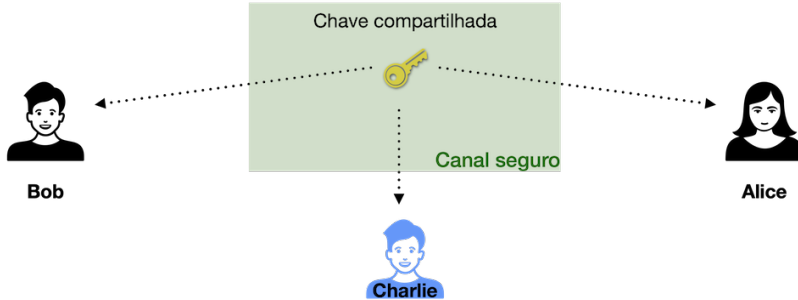
- $s = A^b \mod p$



Fonte: Adaptado de Wikipedia

E se for necessário conversar com mais de uma parte?

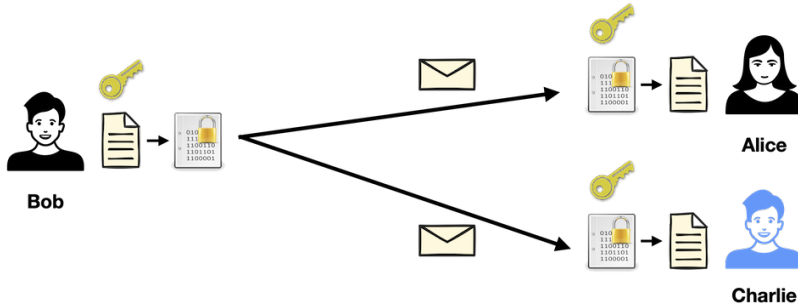
Dificuldades com criptografia simétrica



- É necessário compartilhar a mesma chave com cada uma das partes
- Se uma das partes for comprometida, todas as partes são comprometidas

E se for necessário conversar com mais de uma parte?

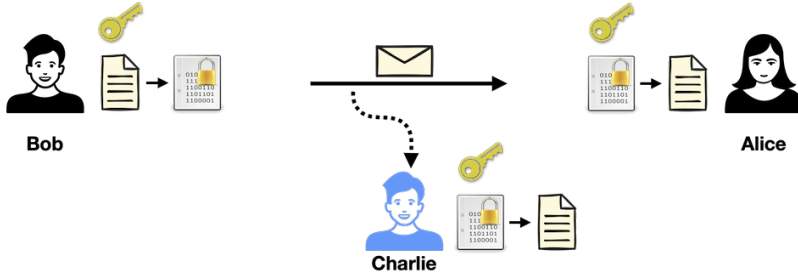
Dificuldades com criptografia simétrica



- É necessário compartilhar a mesma chave com cada uma das partes
- Se uma das partes for comprometida, todas as partes são comprometidas

E se for necessário conversar com mais de uma parte?

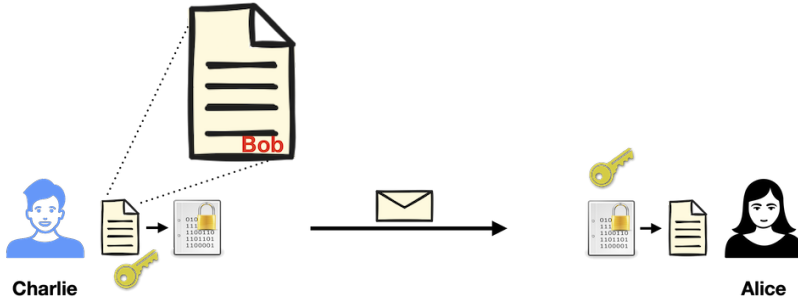
Dificuldades com criptografia simétrica



- Não é possível garantir a confidencialidade entre duas partes
- Todos que conhecem a chave podem decifrar a mensagem

E se for necessário conversar com mais de uma parte?

Dificuldades com criptografia simétrica



- Não é possível garantir a autenticidade da origem da mensagem
- Qualquer parte pode se passar por outra parte

Comunicação segura entre duas partes

Criptografia assimétrica

Garante a confidencialidade, a autenticidade e a integridade das mensagens

- **Cada parte possui um par de chaves: pública e privada**

- A chave pública é compartilhada com todos
- A chave privada é mantida em segredo

- **Para garantir a confidencialidade**

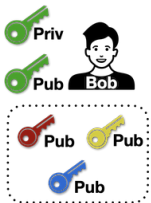
- A chave pública é usada para cifrar a mensagem
- A chave privada é usada para decifrar a mensagem

- **Para garantir a autenticidade**

- A chave privada é usada para assinar a mensagem
- A chave pública é usada para verificar a assinatura

Comunicação segura entre duas partes

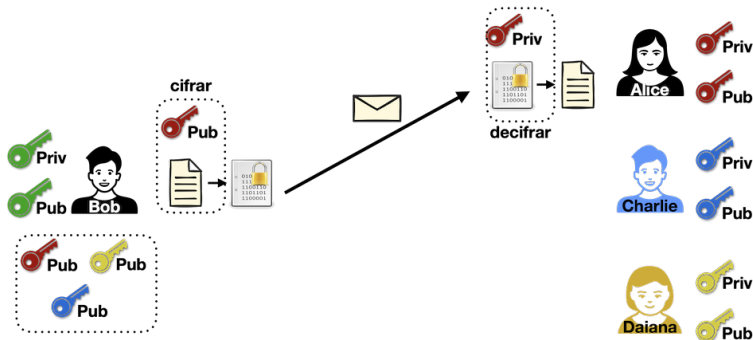
Criptografia assimétrica



- Bob mantém um chaveiro com as chaves públicas de seus contatos
- Todos os demais contatos podem manter seu próprio chaveiro

Comunicação segura entre duas partes

Criptografia assimétrica



- Bob cifra a mensagem com a chave pública de Alice
- Somente Alice pode decifrar a mensagem com sua chave privada

Comunicação segura entre duas partes

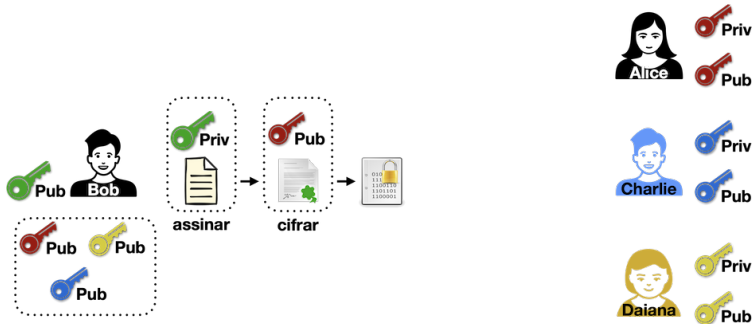
Criptografia assimétrica



- Alice assina a mensagem com sua chave privada
- Bob verifica a assinatura com a chave pública de Alice

Comunicação segura entre duas partes

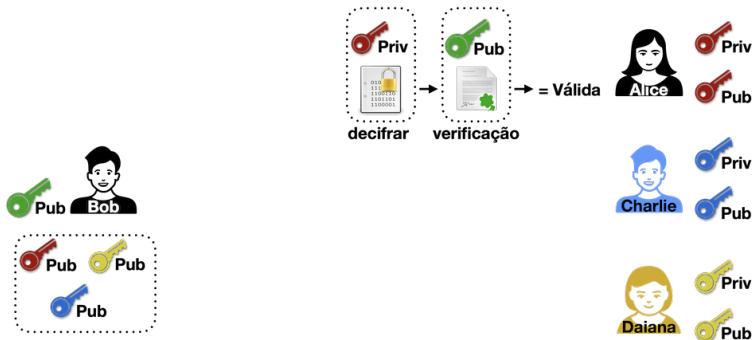
Criptografia assimétrica



- Bob assina a mensagem com sua chave privada e cifra com a chave pública de Alice

Comunicação segura entre duas partes

Criptografia assimétrica



- Alice decifra a mensagem com sua chave privada e verifica a assinatura com a chave pública de Bob

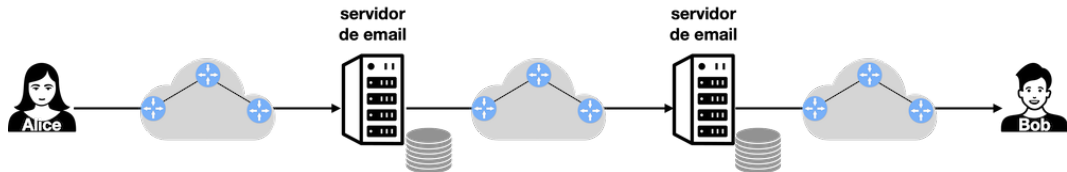
Criptografia fim-a-fim (E2EE, *end-to-end encryption*)

Garante que o conteúdo só possa ser acessado pelo remetente e o destinatário



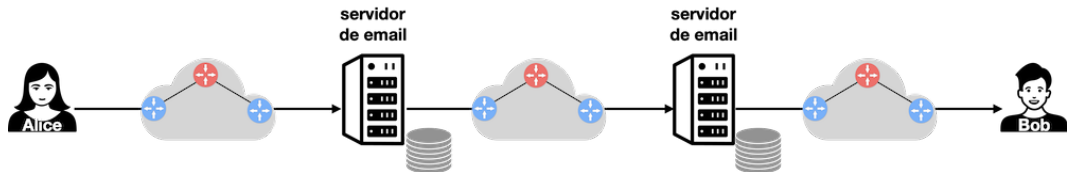
Criptografia fim-a-fim (E2EE, *end-to-end encryption*)

Garante que o conteúdo só possa ser acessado pelo remetente e o destinatário



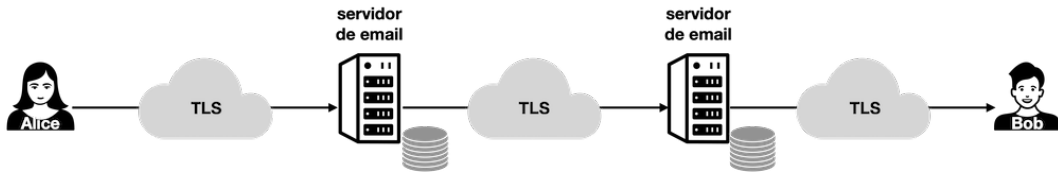
Criptografia fim-a-fim (E2EE, *end-to-end encryption*)

Garante que o conteúdo só possa ser acessado pelo remetente e o destinatário



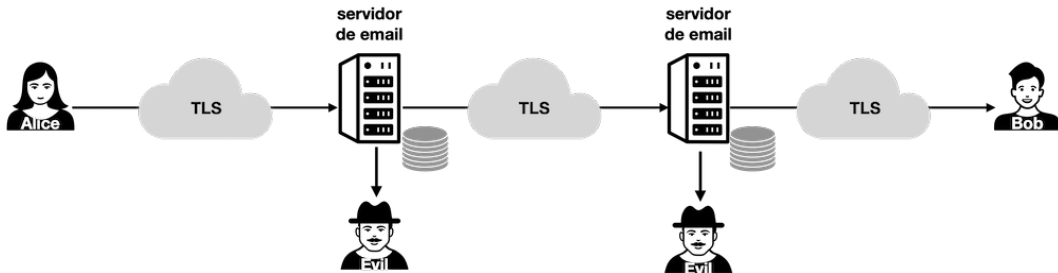
Criptografia fim-a-fim (E2EE, *end-to-end encryption*)

Garante que o conteúdo só possa ser acessado pelo remetente e o destinatário



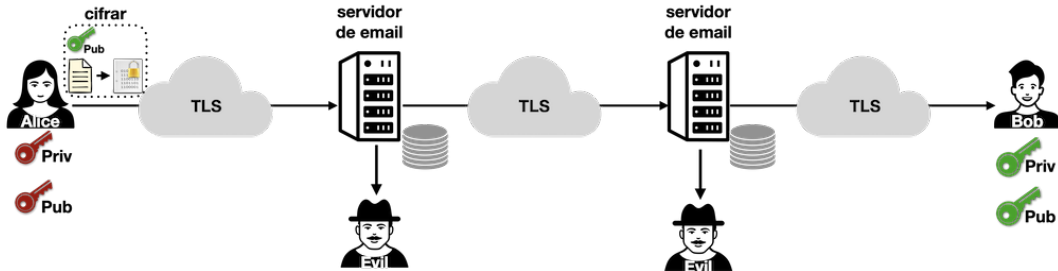
Criptografia fim-a-fim (E2EE, *end-to-end encryption*)

Garante que o conteúdo só possa ser acessado pelo remetente e o destinatário



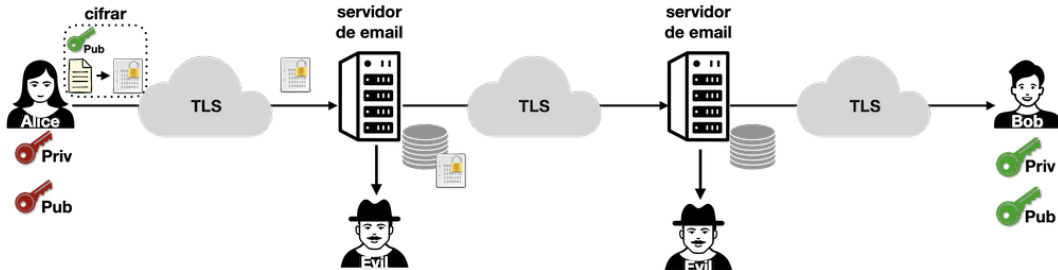
Criptografia fim-a-fim (E2EE, *end-to-end encryption*)

Garante que o conteúdo só possa ser acessado pelo remetente e o destinatário



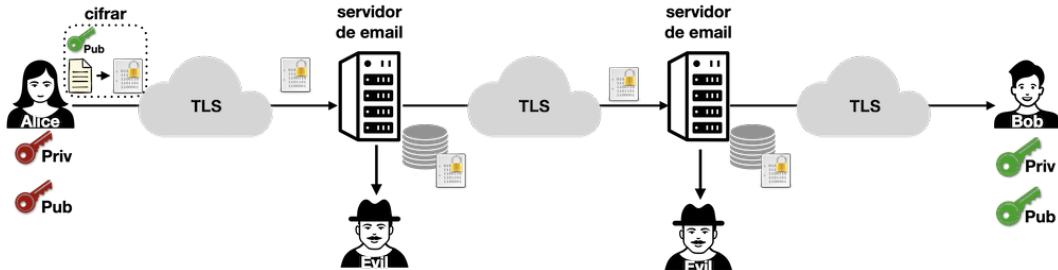
Criptografia fim-a-fim (E2EE, *end-to-end encryption*)

Garante que o conteúdo só possa ser acessado pelo remetente e o destinatário



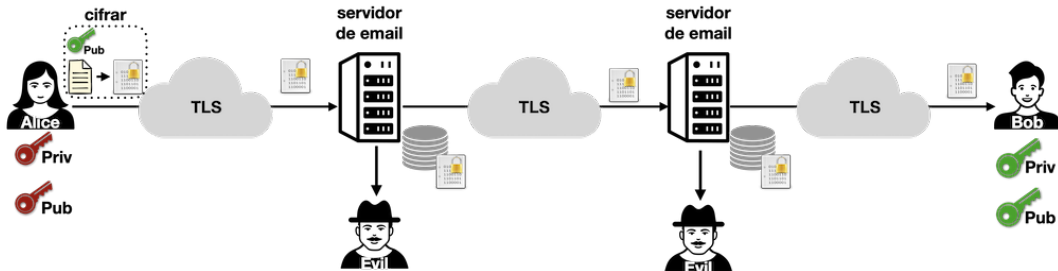
Criptografia fim-a-fim (E2EE, *end-to-end encryption*)

Garante que o conteúdo só possa ser acessado pelo remetente e o destinatário



Criptografia fim-a-fim (E2EE, *end-to-end encryption*)

Garante que o conteúdo só possa ser acessado pelo remetente e o destinatário



Função de dispersão criptográfica

Integridade

Garantia de que a informação não foi alterada de forma não autorizada

- Como detectar que um arquivo foi corrompido?
- Como detectar que dados foram alterados de forma não autorizada?

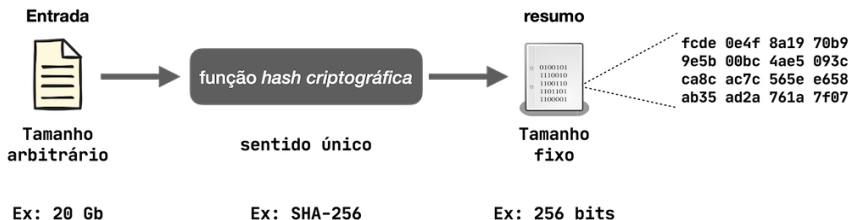
Funções de dispersão criptográfica

- Autenticação de usuários (senhas)
- Autenticação de mensagens (HMAC)
- Integridade ou identificador de arquivos (imagens ISO, arquivos no git)
- Assinaturas digitais
- Prova de trabalho (mineração de Bitcoins)

Função de dispersão criptográfica

Cryptographic hash function

- Mapeia entrada de tamanho arbitrário para uma saída de tamanho fixo
 - A saída é chamada de *hash* ou *resumo criptográfico*
- Função unidirecional, não é possível recuperar a entrada a partir do *hash*



Funções de dispersão criptográfica

Propriedades

■ Determinismo

- A mesma entrada sempre produz o mesmo *hash*

■ Resistência a pré-imagem (sentido único)

- Dado um *hash* h , deve ser computacionalmente inviável encontrar uma entrada m tal que $h = \text{hash}(m)$

■ Resistência a segunda pré-imagem (resistência a colisão fraca)

- Dado uma entrada m_1 , deve ser computacionalmente inviável encontrar uma entrada m_2 diferente de m_1 tal que $\text{hash}(m_1) = \text{hash}(m_2)$

■ Resistência a colisões

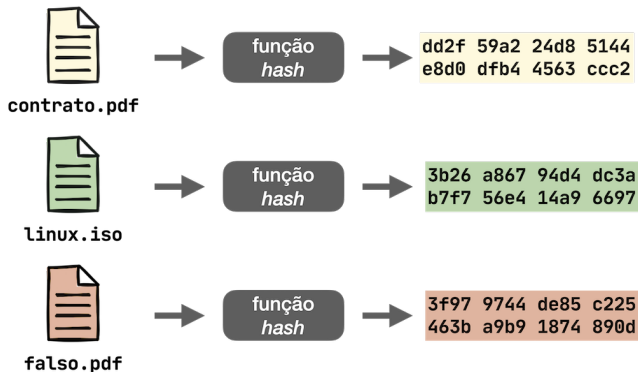
- Deve ser computacionalmente inviável encontrar duas entradas diferentes, m_1 e m_2 , tal que $\text{hash}(m_1) = \text{hash}(m_2)$

■ Difusão

- Uma pequena alteração na entrada deve produzir um *hash* completamente diferente

Funções de dispersão criptográfica

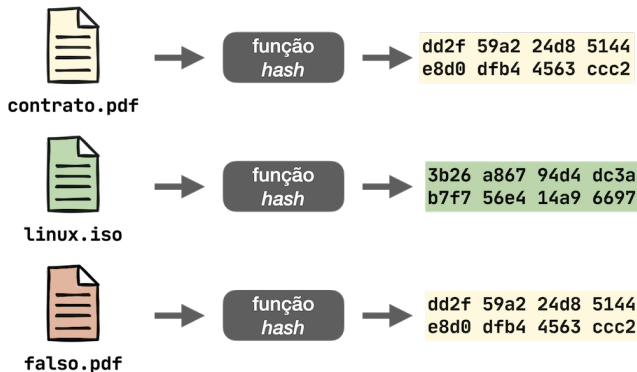
Colisões permitiriam um adversário substituir ou modificar uma informação sem ser detectada



- Entradas diferentes devem produzir *hashes* diferentes
- Colisões são possíveis, mas devem ser computacionalmente inviáveis de serem encontradas

Funções de dispersão criptográfica

Colisões permitiriam um adversário substituir ou modificar uma informação sem ser detectada



- Entradas diferentes devem produzir *hashes* diferentes
- Colisões são possíveis, mas devem ser computacionalmente inviáveis de serem encontradas

Algoritmos de funções de dispersão criptográfica

Algoritmo	Resumo (bits)	Comentários
MD5	128	Inseguro, usado na NF-e no Brasil
SHA-1	160	Inseguro, usado no git e torrents
SHA-2	224 a 512	Bitcoin usa SHA-256
SHA-3	256 e 512	Padrão NIST

- Em 2017 a Google apresentou um ataque de colisão prático para o SHA-1
 - É possível criar dois PDFs diferentes com o mesmo *hash*



MD5

1 smartphone
30 sec



SHA-1 Shattered

110 GPU
1 year



SHA-1 Bruteforce

12.000.000 GPU
1 year

Fonte: <https://shattered.io>

HMAC (*Hash-based Message Authentication Code*) I

- Forma de autenticação de mensagens que combina uma chave secreta com uma função de resumo criptográfico
- Permite verificar a autenticidade e a integridade de uma mensagem, mas não garante a confidencialidade

HMAC (*Hash-based Message Authentication Code*) II

- Emissor e receptor compartilham uma chave secreta, que é usada para calcular o HMAC, o qual é enviado junto com a mensagem

$$\text{HMAC}(K, M) = \text{hash}((K \oplus \text{opad}) \parallel \text{hash}((K \oplus \text{ipad}) \parallel M)) \quad (3)$$

, sendo *hash* a função de resumo criptográfico⁴, *K* a chave secreta, *M* a mensagem, *opad*⁵ o preenchimento externo, *ipad* o preenchimento interno, \oplus a operação XOR e \parallel a concatenação

⁴Exemplo: HMAC-SHA256

⁵Necessário para adequar o tamanho da chave com o tamanho do bloco da função de resumo criptográfico

Exemplo em Java para gerar um HMAC

```
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import java.nio.charset.StandardCharsets;
import java.util.Base64;

public class ExemploHmac {

    public static void main(String[] args) throws Exception {
        String chave = "sua chave secreta";
        String dados = "Aula de Segurança no ADS";
        SecretKeySpec keySpec = new SecretKeySpec(chave.getBytes(StandardCharsets.UTF_8), "HmacSHA256");
        Mac mac = Mac.getInstance("HmacSHA256");
        mac.init(keySpec);
        byte[] hmacBytes = mac.doFinal(dados.getBytes(StandardCharsets.UTF_8));
        System.out.println("HMAC: " + Base64.getEncoder().encodeToString(hmacBytes));
    }
}

// Saída: HMAC: uxgrTn9UW+RyYl1VuV/7lqaKbC04xRPwYRv+vv7cvIg=
```

Curiosidades

Codificação em BASE64

- Transformar um texto em BASE64⁶ não é uma forma de criptografia
- BASE64 é uma forma de codificar dados binários em ASCII
- A codificação BASE64 é usada para transmitir dados binários em protocolos de comunicação como HTTP, SMTP, POP3, IMAP, FTP, etc.

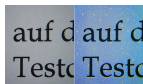
```
echo -n "ADS IFSC" | base64  
# Saída: QURTIELGUOM=
```

⁶<https://www.base64encode.org>

Curiosidades

Esteganografia

- Prática de esconder mensagens dentro de outras mensagens
 - Usada para esconder uma mensagem em pixels menos significativos em uma imagem, em um vídeo, em um áudio, etc.
- Em 2004 foi revelado que todas impressoras laser coloridas produzem um código de identificação invisível a olho nu
 - Matriz de pontos amarelos que identifica a impressora e a data e hora da impressão



Na imagem da árvore, a esteganografia foi usada para esconder a imagem do gato



Fonte: Wikipedia

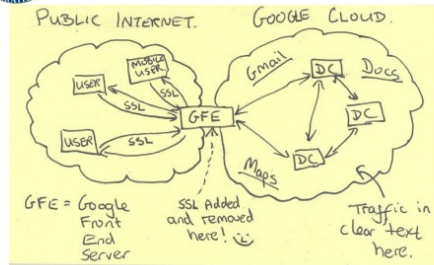
Curiosidades

Programa de vigilância MUSCULAR, parceria entre a NSA e a GCHQ (Reino Unido)

- Em 2013, Edward Snowden revelou que dados eram interceptados entre os centros de dados da Google e Yahoo



Current Efforts - Google



TOP SECRET//SI//NOFORN

Fonte: U.S. National Security Agency

Referências

- Os ícones representando as pessoas foram obtidos de <https://uxwing.com>