

# Criptografia Simétrica

Prof. Vilmar Abreu Junior  
vilmar.abreu@pucpr.br

# Agenda

- Termos
- Conceitos básicos
- Elementos
- Tipo de ataques
- Algoritmos

# Termos

- **Cifração/Encriptação:** Processo de transformar um texto às claras (original) utilizando um algoritmo em um texto codificado/embaralhado;
- **Decifração/Decriptação:** Processo de restaurar um texto cifrado em um texto às claras (original);
- **Criptografia:** Conjunto de técnicas de escrita de texto utilizando cifração.

# Antes de mais nada, é bom saber:

- Nenhuma técnica de criptografia é **completamente segura**;
- Porém dois **critérios** são considerados:
  - O **custo** para quebrar a cifração excede o valor da informação;
  - O **tempo** para quebrar a cifração excede a vida útil da informação.

# Criptografia Simétrica

- Também conhecida como: Criptografia de **chave única** ou criptografia **convencional**;
- Técnica elementar para prover **confidencialidade** para dados **transmitidos** ou **armazenados**;
- A cifração e decifração da informação é realizada utilizando a **mesma chave**.

# Criptografia Simétrica (cont.)

- **Único** tipo de criptografia existente até o final da **década de 70**;
- Conceito existe desde a **época** de Júlio Cesar até os dias de hoje;
- Continua sendo **amplamente** a técnica mais utilizada.

# Mas o que é uma chave?

- String que determina a **saída** de um algoritmo de cifração;
- Tamanho pode ser **tão** grande quanto a mensagem a ser cifrada;
- Chaves de tamanho a partir de 80 bits são consideradas **apropriadas**;
- Chave **não** é sinônimo de senha.

# Elementos

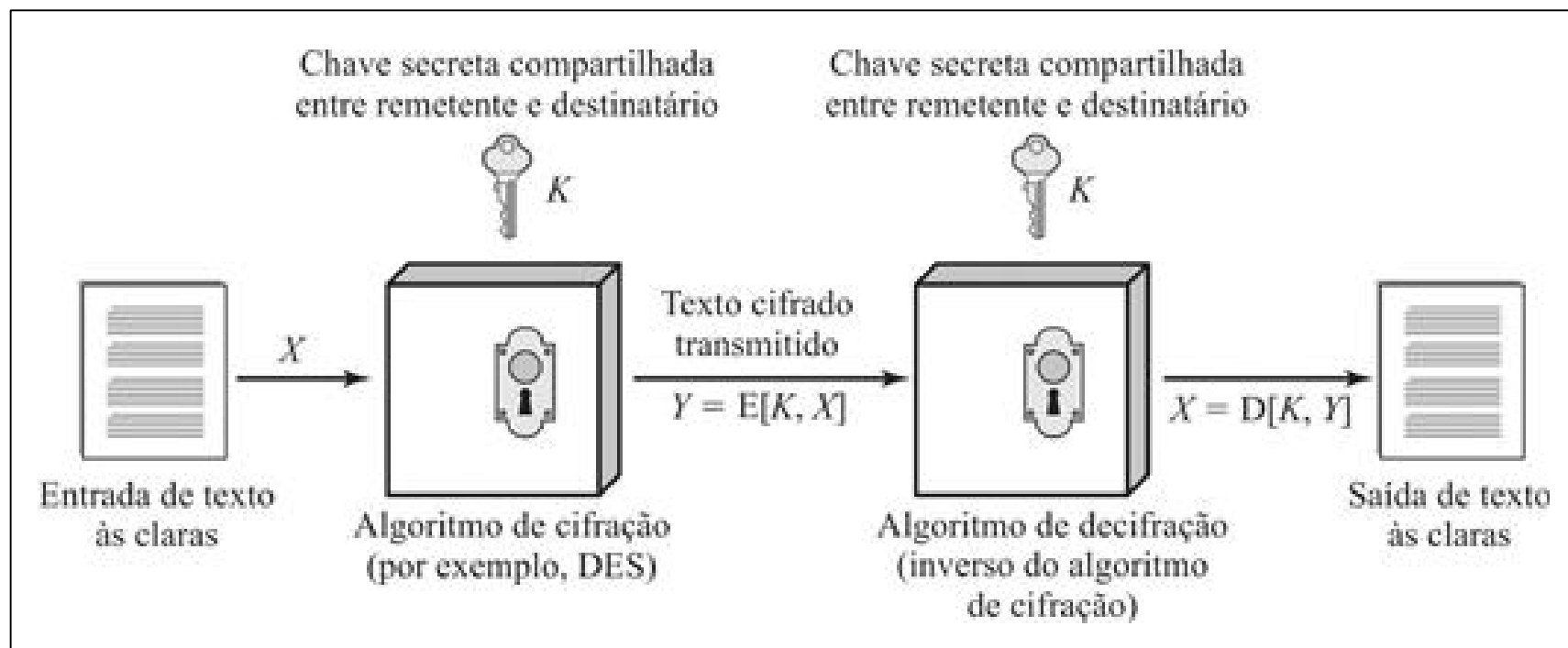
- **Texto às claras:** Mensagem/dado original;
- **Algoritmo de cifração:** Executa substituições e transformações no texto às claras;
- **Chave secreta:** As substituições/transformações do algoritmo dependem da chave.



# Elementos (cont.)

- **Texto cifrado:** Mensagem/dado embaralhado produzido pelo algoritmo de cifração, utilizando o texto às claras e a chave secreta;
- **Algoritmo de decifração:** É, essencialmente, o algoritmo de criptografia executado ao contrário. Recebe o texto criptografado e a chave secreta, produzindo o texto às claras original.

# Modelo Simplificado



# Premissas

- **Distribuição das chaves:** Remetente e destinatário devem **obter** a chave de maneira segura e **manter** em segurança;
- **Algoritmo de cifração robusto:** Caso o oponente tenha acesso a um ou mais textos cifrados, ele não deve ser capaz de **decifrar** o texto ou **descobrir** a chave;
  - Idealmente, mesmo que o oponente tenha o **texto cifrado** e o **texto às claras**, ele é incapaz de **descobrir** a chave.

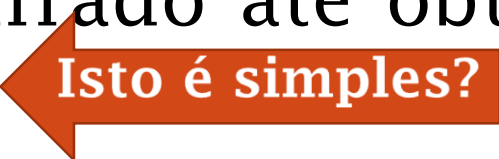
# Como atacar?

- Há duas abordagens gerais para atacar um esquema de cifração simétrica:
  - **Criptanálise;**
  - **Ataque de força bruta.**

# Criptanálise

- **Descrição:** Recorrem a natureza do algoritmo e amostras de textos cifrados e o texto às claras correspondente;
- **Funcionamento:** Explora as características do algoritmo para tentar deduzir o texto às claras ou a chave que está sendo usado;
- **Caso tenha sucesso:** Efeito catastrófico, todas as mensagens futuras e passadas cifradas com aquela chave são comprometidas.

# Ataque de força bruta

- **Descrição:** Tentar todas as chaves possíveis em um texto cifrado até obter uma tradução inteligível;  Isto é simples?
- **Funcionamento:** Em média, deve ser tentada metade de todas as chaves possíveis para obter sucesso;
- **Caso tenha sucesso:** Efeito catastrófico, todas as mensagens futuras e passadas cifradas com aquela chave são comprometidas.

# E se o texto às claras for um arquivo ZIP?

~+Wu"- Ω-0)≤4{∞‡, ë~Ω%ràu.-í Ø-z-  
Ú≠2Ò#Åæð æ«q7,Ωn.®3NÔÚ Ez'Y-f∞Í[±Û\_ èΩ,<NO-±«~xă Åăfèü3Å  
x)ö§k°Å  
\_yÍ ^ΔÉ] ,¤ J/\*iTê&1 'c<uΩ-  
ÄD(G WÄC~y\_ïöÄW PÔ1«ÎÛ†ç],¤;~î^üÑπ~≈~L~9OgflO~&Æ≤ ~≤ ØÔ§":  
~Æ!SGqèvo^ ú\,S>h<-\*6ø‡%x'"|fiÓ#≈~my%~≥ñP<,fi Áj ÅÔ¿"Zù-  
Ω"Ö-6Ëÿ{% „ΩÊó ,i π+Áî\*ú02çSÿ'O-  
2Äfißi /@^"ΠK²\*PÆπ,úé^'3Σ~ö~ÔZî"Y-ÿΩæY> Ω+eô/' <KÆ¿\*+~"≤Ü~  
B ZøK~Qßÿüf,!òfiîzssS/]>ÈQ ü



# Tempo médio requerido para busca exaustiva de chave

Tamanho da chave (bits)	Número de chaves possíveis	Tempo requerido para um PC	Tempo requerido para um super PC
32	$2^{32}$	35,8 minutos	2,15 milisegundos
56	$2^{56}$	1.142 anos	10,01 horas
128	$2^{128}$	$5,4 \times 10^{24}$ anos	$5,4 \times 10^{18}$ anos
168	$2^{168}$	$5,4 \times 10^{36}$ anos	$5,4 \times 10^{30}$ anos

**Observação:** Considerando que cada decifração leva 1  $\mu$ s para PC, e um super PC realiza 1 milhão de decifrações por  $\mu$ s.



# O Algoritmo precisa ser secreto?

- Não, a criptografia simétrica assume que é impraticável decifrar uma informação apenas com a **informação cifrada** e **conhecimento do algoritmo**!
- O que isso implica?
  - Não é necessário investimentos no desenvolvimento de algoritmos de criptografia, eles são **consolidados** e **disponibilizados** na literatura!

# Tipos de Algoritmos

## **Algoritmos baseados em substituição:**

- Cada elemento (letra, bit, etc) do texto às claras é mapeado em outro elemento;

## **Algoritmos baseados em transposição:**

- Cada elemento do texto às claras é rearranjado;

**Nenhuma informação é perdida!**

# Cifra de Cesar

- É o algoritmo mais simples e antigo conhecido de substituição;
- Cada letra é substituída pela letra que está a X índices a direita;
- **Texto às claras:** Professor gente boa
- **Texto cifrado:** Surihvvrj jhqwh erd
- Como realizar o ataque de força bruta?

# Exercício 01

Implemente a Cifra de Cesar;

# Exercício 2

Crie um algoritmo de força bruta para quebrar a chave da Cifra de Cesar.

# Distribuição de chave simétrica

Prof. Vilmar Abreu Junior  
vilmar.abreu@pucpr.br

# Distribuição de chaves

- Para a criptografia de chave simétrica funcionar, é preciso que as duas partes tenham a **mesma chave** e que a mantenham **protegida**;
- Além disso, é desejável que frequentemente exista **troca de chaves** para diminuir a possibilidade de comprometimento do sistema;
- A distribuição de chaves é uma técnica para **entregar** chaves para entidades que desejam conversar, sem que outras entidades vejam a chave.



# Como Bob pode distribuir a chave para Alice?

1. Bob seleciona a chave e entrega fisicamente para Alice;
2. Uma terceira entidade seleciona a chave e entrega fisicamente para Bob e Alice;
3. Se Bob e Alice já possuem previamente uma chave, Bob pode transmitir a nova chave cifrando ela na chave antiga;
4. Se Bob e Alice tiverem uma conexão criptografada com uma terceira entidade, essa pode transmitir a chave cifrada.



# Centro de Distribuição de Chaves (KDC)

- O KDC (*Key Distribution Center*) é uma entidade terceira responsável por **distribuir chaves**;
- Esquema de distribuição amplamente utilizado;
- Cada usuário/processo **compartilha** uma chave única com o KDC;
- Baseado no conceito de hierarquia de chaves.

# Hierarquia de chaves

- A comunicação entre duas entidades é realizada utilizando uma chave **temporária**, chamada de **chave de sessão**;
  - Normalmente a duração/utilização dessa chave está relacionada a uma conexão, depois é descartada;
- Cada chave de sessão é obtida no KDC através de uma conexão criptografada utilizando a **chave mestre**, que é compartilhada entre o KDC e o usuário/processo.

# Hierarquia de chaves (cont.)

- Ou seja, cada usuário/processo **compartilha** uma chave única com o KDC;
- Como é realizado esse compartilhamento de chaves?
  - Normalmente de maneira **física**.

# Cenário de Distribuição de chaves

- Premissas:
  - Bob deseja conversar com Alice utilizando criptografia simétrica;
  - Bob não compartilha uma chave simétrica com Alice
  - Bob compartilha uma chave simétrica com o KDC ( $K_{bob}$ )
  - Alice compartilha uma chave simétrica com o KDC ( $K_{alice}$ )

# Passo a passo (1/5)

Bob requisita ao KDC uma **chave de sessão** ( $K_{sess\tilde{a}o}$ ) para conversar com Alice, esta mensagem contém:

- Identificador de Bob
- Identificador de Alice
- *Nonce* (Identificador único: normalmente um número aleatório ou *timestamp*, tem a finalidade de identificar a requisição)

# Passo a passo (2/5)

KDC responde com uma mensagem cifrada utilizando  $K_{bob}$  , ou seja, apenas Bob consegue decifrar;

A mensagem contém duas informações direcionadas para Bob e duas para Alice.

# Passo a passo (2/5) – (cont.)

Informações direcionadas para Bob:

- Chave de sessão ( $K_{sess\tilde{a}o}$ ), que será utilizada para comunicar com Alice;
- Requisição inicial (Passo 1), com o Nonce incluso;
  - Permite identificar e verificar a integridade da mensagem



# Passo a passo (2/5) – (cont.)

Informações direcionadas para Alice cifradas utilizando a  $K_{alice}$ :

- Chave de sessão ( $K_{sess\tilde{a}o}$ ), que será utilizada para comunicar com Alice;
- Identificador de Bob (por exemplo, IP)



Bob não consegue ler  
as mensagens  
direcionadas para  
Alice!



# Passo a passo (3/5)

- Bob armazena  $K_{sess\tilde{a}o}$  para ser utilizada posteriormente e encaminha as informações que vieram do KDC para Alice ;
- Como a mensagem está cifrada utilizando a  $K_{alice}$  , Alice sabe que a mensagem foi originada em KDC;
- Dessa forma, Alice conhece o identificador de Bob e a chave de sessão.

# Passo a passo (3/5)

- Nesse momento, Bob e Alice possuem a  $K_{sess\tilde{o}a}$  e podem conversar utilizando criptografia simétrica;
- Entretanto, dois passos a mais são desejáveis.

# Passo a passo

- Nesse momento, Bob e Alice possuem a  $K_{sess\tilde{o}a}$  e podem conversar utilizando criptografia simétrica;
- Entretanto, dois passos a mais são desejáveis.

# Passo a passo (4/5)

- Alice gera um nonce e encaminha para Bob, cifrando na  $K_{sess\tilde{o}}$ .

# Passo a passo (5/5)

- Bob responde Alice executando uma função sobre o nonce recebido, cifrando na  $K_{sessão}$ .
  - Essa função pode ser uma operação matemática, por exemplo: incrementar o nonce.
- Observação: Esses dois últimos passos garantem a autenticação!

# Vida útil da chave de sessão

- Considerando que quanto mais trocas de chaves houver, maior será o nível de segurança, por que um possível atacante terá menos amostras de textos cifrados utilizando a mesma chave;
- Por outro lado, a distribuição de chaves atrasa o início da comunicação entre as entidades, além de consumir recursos;
- Qual a vida útil adequada?
  - Um bom administrador deve balancear esses fatores e considerar fatores externos, como o tipo de conexão.

# Exercício 3

Implemente um KDC utilizando a cifra de cesar como algoritmo de cifração.





**PUCPR**  
GRUPO MARISTA

ESCOLA  
**POLITÉCNICA**