

# Teoria da Informação

Ney Lemke

lemke@exatas.unisinos.br

**Ciência da Computação**  
**Mestrado em Computação Aplicada**  
**Unisinos**

# Criptografia

# Plano da Aula

- Noções Básicas de Segurança
- Elementos de Criptografia
- Criptografia de Chave Pública
- Algoritmo RSA

# Noções Básicas

- Ataques à segurança
- Mecanismos de Segurança
- Serviços de Segurança

# Ataques

- Interrupção
- Interceptação
- Modificação
- Fabricação

# Tipos de Ataque

**Passivos:** São aqueles que visam monitorar o sistema ou obter informações confidenciais

**Ativos:** São aqueles que visam alterar informações ou o seu funcionamento.

# Elementos de Criptografia

# Ingredientes Fundamentais

- Texto
- Algoritmo de Codificação
- Texto Codificado
- Algoritmo de Decodificação

# Algoritmo de Codificação

- Mesmo se o inimigo dispuser do algoritmo e conhecer um ou vários textos cifrados se ele não dispuser da chave, a codificação não poderá ser quebrada.
- o emissor deve ter obtido a chave de segurança de uma forma segura e deve mantê-la
- a chave é secreta e não o algoritmo

# Classificação Sistemas

- Tipo de operação usada para codificar o texto original
  - substituição
  - embaralhamento
  - mixto

# Classificação Sistemas

- Número de chaves utilizada
- Forma de processamento do texto
  - em blocos
  - de forma ininterrupta

# Criptanálise

O processo de descoberta da chave ou do texto original é chama de criptanálise.

O criptoanalista sempre possui acesso ao código de criptografia.

# Tipos de Análise

- Apenas texto cifrado.
- Texto original e texto cifrado conhecido.
- Texto original e cifrado escolhido.

Obs. Os algoritmos são desenhados para suportar a categoria texto original e texto cifrado conhecidos

# Esquema Seguro

Um esquema de criptografia computacionalmente seguro satisfaz pelo menos um dos critérios abaixo:

- o custo de quebrar o código excede o custo da informação.
- o tempo para quebrar o código excede o tempo de vida útil da informação.

# Chave Pública

# Ingredientes Fundamentais

- Texto Original
- Algoritmo de codificação
- Chave pública e chave privada
- Texto codificado
- Algoritmo de decodificação

# Passos Básicos

1. cada usuário usa um par de chaves para ser usada na codificação e decodificação.
2. o usuário disponibiliza a chave pública para os outros usuários e mantém uma coleção das chaves dos demais usuários.
3. Quando Bob quer mandar uma mensagem para Alice. Bob codifica a mensagem usando a chave pública de Alice.
4. Quando Alice recebe a mensagem de Bob ela a decodifica usando sua chave privada.

# Requisitos

- É computacionalmente barato para  $B$  gerar o par de chaves ( $KU_p$  pública -  $KR_b$  privada)
- É computacionalmente barato para o usuário  $A$ , conhecendo a chave pública e a mensagem  $M$  gerar o texto codificado:

$$C = E_{KU_b}(M)$$

# Requisitos

- É computacionalmente barato para o receptor  $B$  decodificar o texto codificado usando a chave privada e reobter a mensagem original

$$M = D_{KR_b}(C) = D_{KR_b}[E_{KU_b}(M)]$$

- É computacionalmente impossível para o inimigo conhecendo a chave pública,  $KU_b$ , determinar a chave privada  $KR_b$ .

# Requisitos

- É computacionalmente impossível para o inimigo conhecendo a chave pública  $KU_b$  e um texto cifrado  $C$  reobter  $M$ .
- Qualquer uma das chaves pode ser utilizada na codificação ou decodificação.

# Algoritmo RSA

# Histórico

Proposta por Ron Rivest, Adi Shamir e Len Adleman em 1977.

# Conceitos Básicos

Tanto o texto cifrado  $C$  como o texto original  $M$  são inteiros entre 0 e  $n - 1$ , para algum  $n$ .

$$C = M^e \pmod{n}$$
$$M = C^d \pmod{n} = (M^e)^d \pmod{n}$$

# Conceitos Básicos

O emissor e o receptor devem conhecer  $n$  e  $e$ . Mas apenas o receptor sabe o valor de  $d$

$$KU = \{e, n\}$$

$$KR = \{d, n\}$$

# Condições

1. É possível encontrar  $e$ ,  $d$  e  $n$  tais que  $M^{ed} = M \pmod n$  para todo  $M < n$ .
2. É relativamente fácil calcular  $M^e$  e  $C$  para todos os valores de  $M < n$ .
3. Não é factível determinar  $d$  dado  $e$  e  $n$ .

# Algoritmo RSA

- Selecione dois números primos  $p$  e  $q$ .
- Calcule  $pq = n$
- Calcule  $\phi(n) = (p - 1)(q - 1)$  o número de inteiros menores que  $n$  e primos relativos em relação a  $n$ .
- selecione um número  $e$  que é primo relativo a  $\phi(n)$
- $d = e^{-1} \pmod{\phi(n)}$
- $KU = \{e, n\}$
- $KP = \{d, n\}$

# Comentário Final

Para quebrar o código basta encontrar  $p$  e  $q$  tais que  $pq = n$ .  
Mas o problema é que este problema é computacionalmente intensivo.