

Segurança Informática e nas Organizações - Resumos 2

José Mendes 107188

2023/2024



universidade
de aveiro

1 Criptografia Assimétrica

1.1 Criptografia Assimétrica (de blocos)

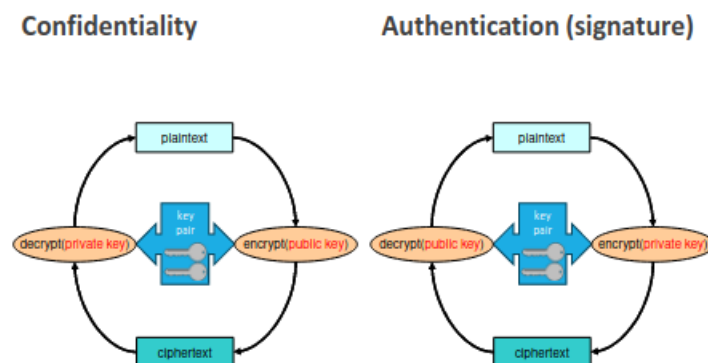
Usa um par de chaves:

- **Chave privada:** pessoal, não transmissível;
- **Chave pública:** disponível a todos;

Permite:

- Confidencialidade sem qualquer exchange of secrets prévia;
- Autenticação
 - De conteúdos (integridade dos dados);
 - De origem (atenticação da source, ou assinatura digital);

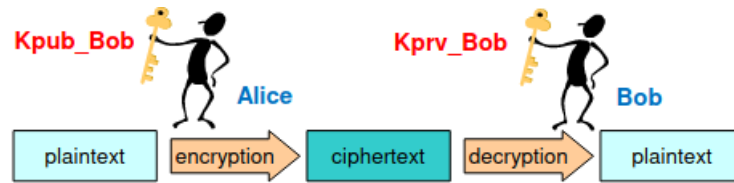
1.2 Operações de uma Cifra Assimétrica



1.3 Use Cases: Comunicação Segura

Comunicação segura com um target (Bob)

- A Alice encripta o plaintext **P** com a chave pública do Bob, **Kpub_Bob**
 - **Alice:** $C = \{P\}_{k_{pub_bob}}$
- O Bob decifra o ciphertext **C** com a sua chave privada, **Kpriv_Bob**
 - **Bob:** $P' = \{C\}_{k_{priv_bob}}$
- P' deve ser igual a **P** (é necessário verificar)
- **Kpub_Bob** precisa de ser conhecida pela Alice



1.4 Cifras Assimétricas

Vantagens:

- São um mecanismo de autenticação fundamental;
- Permitem explorar características que não são possíveis com cifras simétricas;

Desvantagens:

- Performance;
- Normalmente não são muito eficientes e consomem muita memória;

Problemas:

- Distribuição confiável de chaves públicas;
- O lifetime do par de chaves é limitado;

Abordagens: problemas matemáticos complexos

- Logaritmos discretos de números grandes;
- Factorização inteira de números grandes;

Algoritmos mais comuns:

- RSA;
- ElGamal;
- Elliptic Curves (ECC);

Outras técnicas com pares assimétricos de chaves:

- Diffie-Hellman (key agreement);

1.5 RSA (Rivest, Shamir, Adelman, 1978)

Chaves:

- **Privada:** (d, n)
- **Pública:** (e, n)

Encriptação da chave pública (confidencialidade)

- $C = P^e \bmod n$
- $P = C^d \bmod n$

Encriptação da chave privada (assinatura)

- $C = P^d \bmod n$
- $P = C^e \bmod n$

P, C are numbers
 $0 \leq P, C < n$

Complexidade Computacional

- Logaritmo discreto;
- Factorização inteira;

Seleção de Chaves

- **n** grande (centenas ou milhares de bits);
- $n = p \times q$ com **p** e **q** sendo números primos grandes (secrets);
- Escolher um **e** co-primo de $(p - 1) \times (q - 1)$;
- Computar **d** tal que $e \times d \equiv 1 \pmod{(p - 1) \times (q - 1)}$;
- Descartar **p** e **q**;
- O valor de **d** não pode ser facilmente computado a partir de **e** e **n** (apenas de **p** e **q**);

1.5.1 RSA - Exemplo

p = 5 q = 11 (prime numbers)

- $n = p \times q = 55$
- $(p-1) \times (q-1) = 40$

e = 3 (public key = e, n)

- Coprime of 40

d = 27 (private key = d, n)

- $e \times d \equiv 1 \pmod{40} \rightarrow d \times e \pmod{40} = 1, (27 \times 3) \pmod{40} = 1$

For P = 26 (notice that P, C ∈ [0, n-1])

- $C = P^e \pmod{n} = 26^3 \pmod{55} = 31$
- $P = C^d \pmod{n} = 31^{27} \pmod{55} = 26$

1.6 Encriptação Híbrida

Mistura criptografia simétrica com assimétrica

- Usa o melhor dos dois mundos, evitando os problemas;
- Cifra assimétrica: usa chaves públicas (mas é lenta);
- Cifra simétrica: Rápida (mas com métodos fracos de troca de chaves);

Método

- Obtém K_{pub} do destinatário;
- Gera uma chave simétrica aleatória K_{sym} ;
- Calcula $C1 = E_{sym}(K_{sym}, P)$;
- Calcula $C2 = E_{asym}(K_{pub}, K_{sym})$;
- Envia $C1 + C2$;
 - $C1$ é o texto encriptado com a chave simétrica;
 - $C2$ é a chave simétrica encriptada com a chave pública do destinatário (pode também conter um IV);

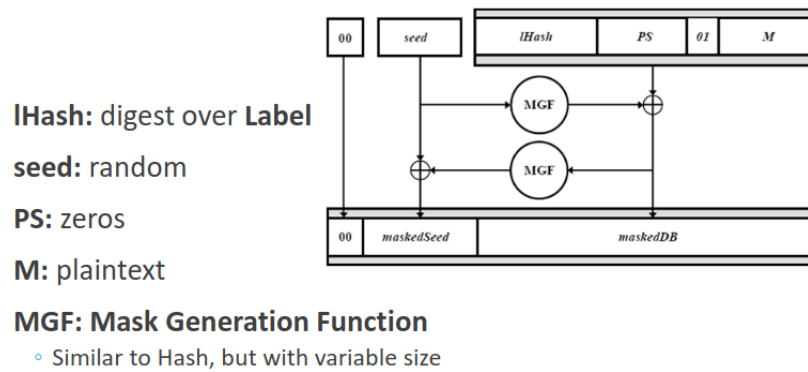
1.7 Randomização de encriptações assimétricas

Resultado de encriptações assimétricas não determinístico (não é previsível)

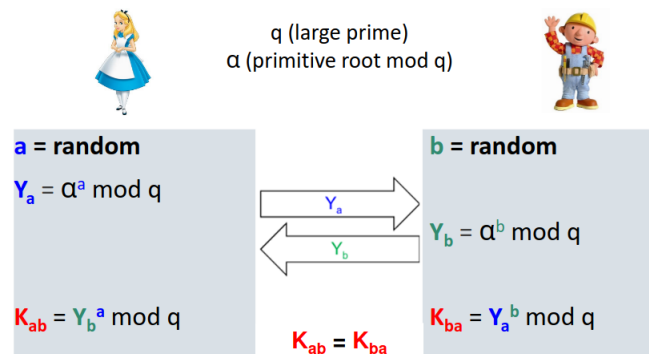
- **N** encriptações do mesmo valor, com a mesma chave, deve produzir **N** resultados diferentes;
- **Objetivo:** Prevenir a descoberta de valores encriptados através de tentativa e erro;

Abordagens: Concatenação de um valor a encriptar com dois valores, um fixo (para controlo de integridade) e outro aleatório (para randomização);

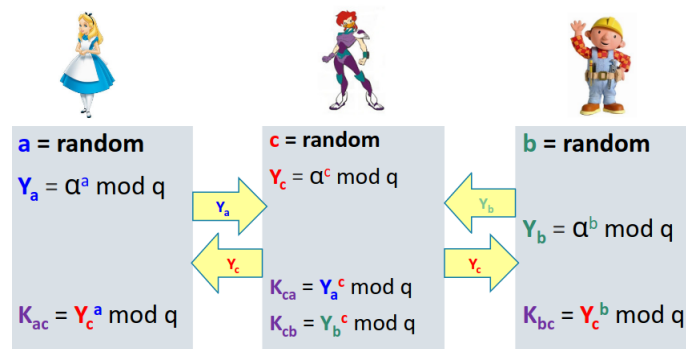
1.7.1 OAEP (Optimal Asymmetric Encryption Padding)



1.8 Diffie-Hellman Key Agreement (1976)



1.8.1 DH Key Agreement: MitM Attack



1.9 Elliptic Curve Cryptography (ECC)

Curvas elípticas são funções específicas

- Têm um gerador G ;
- Uma chave privada K_{priv} , é um inteiro com um máximo de bits permitidos pela curva;
- Uma chave pública K_{pub} , é um ponto $(x, y) = K_{priv} \times G$
- Dada K_{pub} , deve ser computacionalmente difícil determinar K_{priv} ;

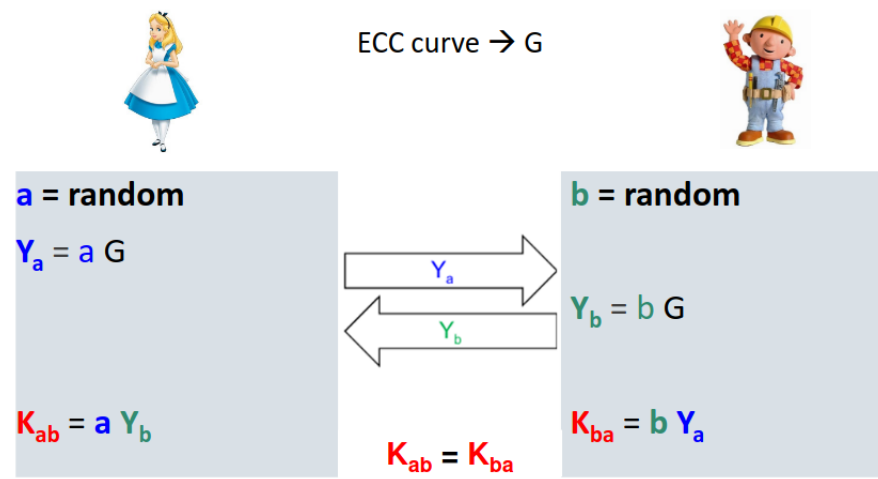
Curves

- NIST curves (15)
 - P-192, P-224, P-256, P-384, P-521
 - B-163, B-233, B-283, B-409, B-571
 - K-163, K-233, K-283, K-409, K-571

Other curves

- Curve25519 (256 bits)
- Curve448 (448 bits)

1.10 ECDH: DH com ECC



1.11 Encriptação de chave pública com ECC

Mistura encriptação híbrida com EDHC

Método

- Obtém K_{pub_recv} do destinatário;
- Gera um random K_{priv_send} com um correspondente K_{pub_send} ;
- Calcula $K_{sym} = K_{priv_send} \times K_{pub_recv}$;
- $C = E(P, K_{sym})$;
- Envia $C + K_{pub_send}$;
- Destinatário calcula $K_{sym} = K_{pub_send} \times K_{priv_recv}$;
- $P = D(C, K_{sym})$;

2 Assinaturas digitais