UNIVERSIDADE FEDERAL DE SERGIPE

DEPARTAMENTO DE COMPUTAÇÃO

Criptografia

Projeto e Análise de Algoritmos

Bruno Prado

Departamento de Computação / UFS

Introdução

- Contexto e história
  - A proteção de segredos de estratégicos e militares remonta à civilização egípcia a mais de 4.000 anos

Introdução

- Contexto e história
  - A proteção de segredos de estratégicos e militares remonta à civilização egípcia a mais de 4.000 anos
  - Na era da informação, com o uso massivo de sistemas computacionais e de redes de comunicação, é preciso ainda mais proteção das informações digitais

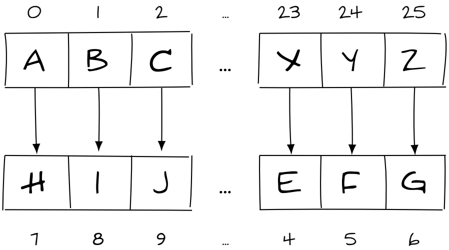
Introdução

- Contexto e história
  - A proteção de segredos de estratégicos e militares remonta à civilização egípcia a mais de 4.000 anos
  - Na era da informação, com o uso massivo de sistemas computacionais e de redes de comunicação, é preciso ainda mais proteção das informações digitais

$$\textit{Cripto} \longleftrightarrow \textit{Escondido}$$
$$\textit{Grafia} \longleftrightarrow \textit{Escrita}$$

Introdução

- Criptografia do imperador Júlio César

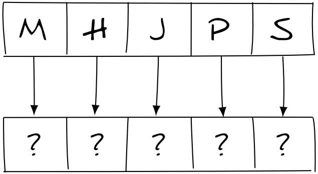


$$E_d(m) = (m_i + d) \bmod 26 = c$$

$$D_d(c) = (c_i - d) \bmod 26 = m$$

Introdução

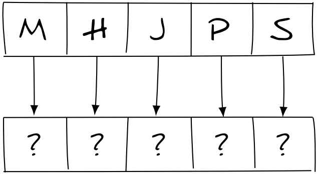
- Criptografia do imperador Júlio César



Quantos passos são necessários
para decifrar este código por força bruta?

Introdução

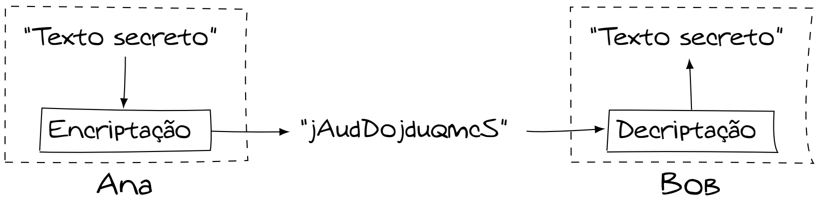
- Criptografia do imperador Júlio César



E se os mapeamentos fossem
gerados a partir de uma permutação?

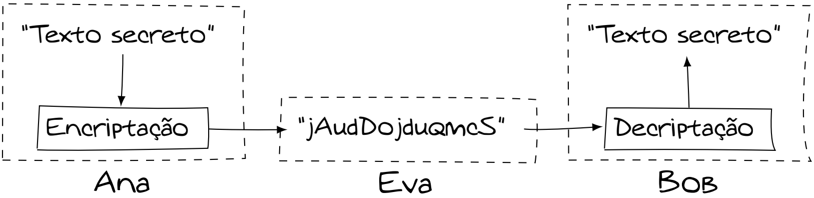
Introdução

- Cenário de funcionamento
  - Comunicação entre terceiros
  - Canal de comunicação inseguro



Introdução

- Cenário de funcionamento
  - Comunicação entre terceiros
  - Canal de comunicação inseguro



Introdução

- O que é criptografia em sistemas?
  - É a aplicação de técnicas matemáticas para proporcionar segurança da informação

Confidencialidade ↔ Sigilo das informações

Introdução

- O que é criptografia em sistemas?
  - É a aplicação de técnicas matemáticas para proporcionar segurança da informação

Confidencialidade ⟷ Sigilo das informações

+

Integridade ⟷ Proteção do conteúdo

Introdução

- O que é criptografia em sistemas?
  - É a aplicação de técnicas matemáticas para proporcionar segurança da informação

Confidencialidade ↔ Sigilo das informações

+

Integridade ↔ Proteção do conteúdo

+

Autenticação ↔ Identificação das partes

Introdução

- O que é criptografia em sistemas?
  - É a aplicação de técnicas matemáticas para proporcionar segurança da informação

Confidencialidade ⟷ Sigilo das informações

\+

Integridade ⟷ Proteção do conteúdo

\+

Autenticação ⟷ Identificação das partes

\+

Não repúdio ⟷ Garantia de acesso legítimo

Introdução

- Função de mão única: é uma função bijetora com baixo custo computacional para gerar os resultados, entretanto, é difícil reverter o resultado, com o objetivo de determinar as entradas utilizadas
  - Multiplicação dos números primos $z = x \times y$ é $O\left(n^2\right)$
  - Fatoração do número z para obter x e y é $O\left(2^n\right)$

Introdução

- Função de mão única: é uma função bijetora com baixo custo computacional para gerar os resultados, entretanto, é difícil reverter o resultado, com o objetivo de determinar as entradas utilizadas
  - Multiplicação dos números primos $z = x \times y$ é $O\left(n^2\right)$
  - Fatoração do número z para obter x e y é $O\left(2^n\right)$

$$
\begin{aligned}
f(x, y) = &\quad \textit{Multiplicação}(x, y) &= z \\
f^{-1}(z) = &\quad \textit{Fatoração}(z) &= x, y
\end{aligned}
$$

Introdução

- Função de mão única: é uma função bijetora com baixo custo computacional para gerar os resultados, entretanto, é difícil reverter o resultado, com o objetivo de determinar as entradas utilizadas
  - Multiplicação dos números primos $z = x \times y$ é $O\left(n^2\right)$
  - Fatoração do número z para obter x e y é $O\left(2^n\right)$

$$\begin{aligned} f(x, y) = &\quad \mathit{Multiplicação}(x, y) &= z \\ f^{-1}(z) = &\quad \mathit{Fatoração}(z) &= x, y \end{aligned}$$

$$\uparrow \mathit{Custo\ problema} \quad \longleftrightarrow \quad \uparrow \mathit{Segurança\ da\ criptografia}$$

Introdução

- Métricas de avaliação da criptografia
  - Nível de segurança: como é de difícil quantificação, uma boa estratégia é medir a quantidade de passos necessários para resolver o problema em questão

Introdução

- Métricas de avaliação da criptografia
  - Nível de segurança: como é de difícil quantificação, uma boa estratégia é medir a quantidade de passos necessários para resolver o problema em questão
  - Desempenho: avalia a eficiência de espaço e de tempo para encriptação dos dados originais, verificando o aumento do volume de dados e a taxa de processamento atingida pelo algoritmo

Introdução

- Métricas de avaliação da criptografia
  - Nível de segurança: como é de difícil quantificação, uma boa estratégia é medir a quantidade de passos necessários para resolver o problema em questão
  - Desempenho: avalia a eficiência de espaço e de tempo para encriptação dos dados originais, verificando o aumento do volume de dados e a taxa de processamento atingida pelo algoritmo
  - Implementação: consiste em verificar a viabilidade prática de implementação de uma solução, utilizando componentes de hardware ou software

Introdução

- Criptografia perfeita (*one-time pad*)
  - Durante a segunda guerra mundial, Claude E. Shannon formalizou o conceito de segredo perfeito, demonstrando que é impossível decifrar uma mensagem cifrada c que não oferece nenhuma informação sobre a mensagem original m

Introdução

- Criptografia perfeita (*one-time pad*)
  - Durante a segunda guerra mundial, Claude E. Shannon formalizou o conceito de segredo perfeito, demonstrando que é impossível decifrar uma mensagem cifrada c que não oferece nenhuma informação sobre a mensagem original m
  - Esta definição requer que para um conjunto de mensagens M, as probabilidades de se obterem a mesma mensagem cifrada c, utilizando um conjunto de chaves perfeitamente aleatórias K, sejam iguais

Introdução

- Criptografia perfeita (*one-time pad*)
  - Durante a segunda guerra mundial, Claude E. Shannon formalizou o conceito de segredo perfeito, demonstrando que é impossível decifrar uma mensagem cifrada c que não oferece nenhuma informação sobre a mensagem original m
  - Esta definição requer que para um conjunto de mensagens M, as probabilidades de se obterem a mesma mensagem cifrada c, utilizando um conjunto de chaves perfeitamente aleatórias K, sejam iguais

$$
\begin{aligned}
M &= \{0, 1\}^n \\
K &= \{0, 1\}^n
\end{aligned}
$$

Introdução

- Criptografia perfeita (*one-time pad*)
  - Durante a segunda guerra mundial, Claude E. Shannon formalizou o conceito de segredo perfeito, demonstrando que é impossível decifrar uma mensagem cifrada c que não oferece nenhuma informação sobre a mensagem original m
  - Esta definição requer que para um conjunto de mensagens M, as probabilidades de se obterem a mesma mensagem cifrada c, utilizando um conjunto de chaves perfeitamente aleatórias K, sejam iguais

$$
\begin{aligned}
M &= \{0, 1\}^n \\
K &= \{0, 1\}^n \\
E_k\left(m = m_0 m_1 \dots m_{n-1}\right) &= c_0 c_1 \dots c_{n-1}, \quad c_i = m_i \oplus k_i \\
D_k\left(c = c_0 c_1 \dots c_{n-1}\right) &= m_0 m_1 \dots m_{n-1}, \quad m_i = c_i \oplus k_i
\end{aligned}
$$

Introdução

- Criptografia perfeita (*one-time pad*)
  - Durante a segunda guerra mundial, Claude E. Shannon formalizou o conceito de segredo perfeito, demonstrando que é impossível decifrar uma mensagem cifrada c que não oferece nenhuma informação sobre a mensagem original m
  - Esta definição requer que para um conjunto de mensagens M, as probabilidades de se obterem a mesma mensagem cifrada c, utilizando um conjunto de chaves perfeitamente aleatórias K, sejam iguais

$$
\begin{aligned}
M &= \{0,1\}^n \\
K &= \{0,1\}^n \\
E_k(m = m_0 m_1 \ldots m_{n-1}) &= c_0 c_1 \ldots c_{n-1}, \quad c_i = m_i \oplus k_i \\
D_k(c = c_0 c_1 \ldots c_{n-1}) &= m_0 m_1 \ldots m_{n-1}, \quad m_i = c_i \oplus k_i \\
&\downarrow \\
|K| &\geq |M|
\end{aligned}
$$

Introdução

- Por que uma técnica de criptografia com segurança perfeita não é amplamente utilizada?

Introdução

- Por que uma técnica de criptografia com segurança perfeita não é amplamente utilizada?
  - É exigida a geração de chaves perfeitamente aleatórias, ou seja, sem padrões ou repetições para cada mensagem encriptada

Introdução

- Por que uma técnica de criptografia com segurança perfeita não é amplamente utilizada?
  - É exigida a geração de chaves perfeitamente aleatórias, ou seja, sem padrões ou repetições para cada mensagem encriptada
  - As chaves geradas precisam ter pelo menos o tamanho da mensagem original, para atender o requisito de não repetição da chave

Introdução

- Por que uma técnica de criptografia com segurança perfeita não é amplamente utilizada?
  - É exigida a geração de chaves perfeitamente aleatórias, ou seja, sem padrões ou repetições para cada mensagem encriptada
  - As chaves geradas precisam ter pelo menos o tamanho da mensagem original, para atender o requisito de não repetição da chave
  - Todas as informações precisam ser protegidas e previamente distribuídas entre as partes, evitando reuso parcial ou total das informações

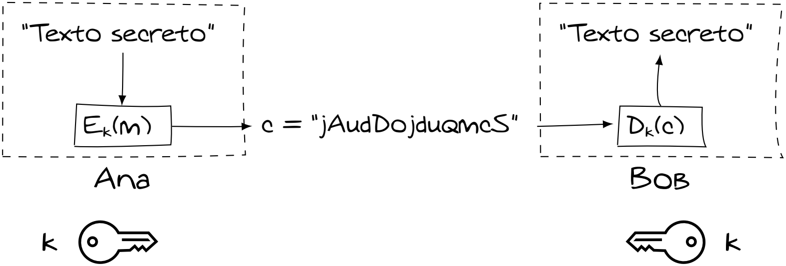
Introdução

- Por que uma técnica de criptografia com segurança perfeita não é amplamente utilizada?
  - É exigida a geração de chaves perfeitamente aleatórias, ou seja, sem padrões ou repetições para cada mensagem encriptada
  - As chaves geradas precisam ter pelo menos o tamanho da mensagem original, para atender o requisito de não repetição da chave
  - Todas as informações precisam ser protegidas e previamente distribuídas entre as partes, evitando reuso parcial ou total das informações

Só é utilizada em aplicações onde
a segurança da informação é muito crítica

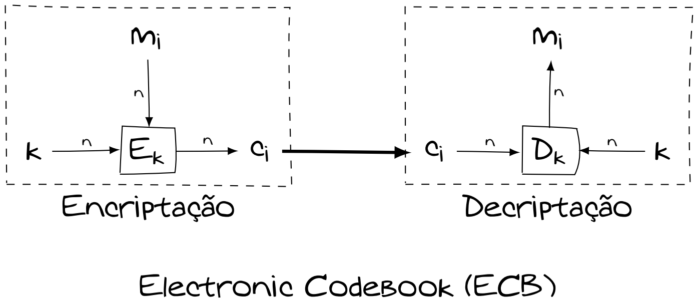
Criptografia

- Criptografia simétrica
  - É um esquema de criptografia que utiliza um conjunto de chaves privadas $k \in K$, tal que $E_k(m) = c$ e $D_k(c) = m$, que precisam ser conhecidas pelas partes envolvidas na comunicação



Criptografia

- Criptografia simétrica
  - Os dados são encriptados ou decriptados utilizando blocos de tamanho fixo com *n* bits (128, 192 ou 256)



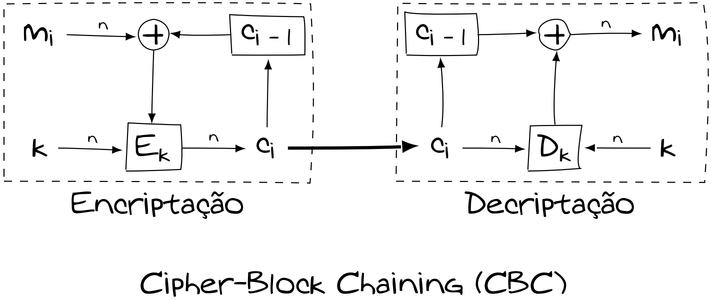
Criptografia

- Criptografia simétrica

```c
// Procedimento de decriptação (AES-ECB)
void aes_d_ecb(uint8_t* m, const uint8_t* c, aes_t*
    aes) {
    // Nr = Número de rodadas
    const uint8_t Nr = aes->Nk + 6;
    // Decriptação AES-EBC
    Decipher(m, c, aes->ke, Nr);
}
// Procedimento de encriptação (AES-ECB)
void aes_e_ecb(uint8_t* c, const uint8_t* m, aes_t*
    aes) {
    // Nr = Número de rodadas
    const uint8_t Nr = aes->Nk + 6;
    // Encriptação AES-EBC
    Cipher(c, m, aes->ke, Nr);
}
```

Criptografia

- Criptografia simétrica
  - Os dados são encriptados ou decriptados utilizando blocos de tamanho fixo com *n* bits (128, 192 ou 256)



Criptografia

- Criptografia simétrica

```
// Procedimento de decriptação (AES-CBC)
void aes_d_cbc(uint8_t* m, const uint8_t* c, size_t
    l, aes_t* aes) {
    // Nr = Número de rodadas
    const uint8_t Nr = aes->Nk + 6;
    // Ponteiro para valor anterior
    const uint8_t* ci1 = aes->c0;
    // Decriptação AES-CBC
    for(size_t i = 0; i < l; i = i + 16) {
        Decipher(m + i, c + i, aes->ke, Nr);
        Xor(m + i, m + i, ci1);
        ci1 = c + i;
    }
    // Salvando c[i - 1]
    memcpy(aes->c0, ci1, 16);
}
```

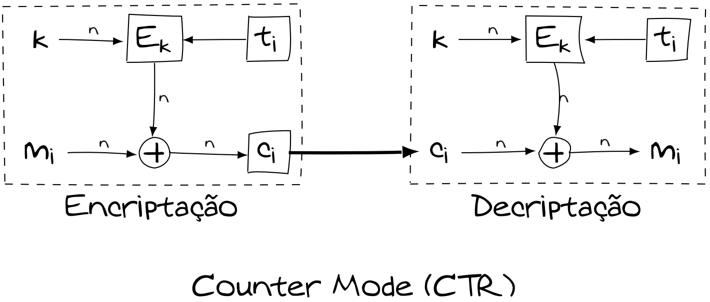
Criptografia

- Criptografia simétrica

```c
// Procedimento de encriptação (AES-CBC)
void aes_e_cbc(uint8_t* c, const uint8_t* m, size_t
    l, aes_t* aes) {
    // Nr = Número de rodadas
    const uint8_t Nr = aes->Nk + 6;
    // Armazenamento de m[i] xor c[i - 1]
    static uint8_t* mxci1 = (uint8_t*)(malloc(16));
    // Ponteiro para valor anterior
    uint8_t* ci1 = aes->c0;
    // Encriptação AES-CBC
    for(size_t i = 0; i < l; i = i + 16) {
        Xor(mxci1, m + i, ci1);
        Cipher(c + i, mxci1, aes->ke, Nr);
        ci1 = c + i;
    }
    // Salvando c[i - 1]
    memcpy(aes->c0, ci1, 16);
}
```

Criptografia

- Criptografia simétrica
  - Os dados são encriptados ou decriptados utilizando blocos de tamanho fixo com *n* bits (128, 192 ou 256)



Criptografia

- Criptografia simétrica

```
// Procedimento de decriptação/encriptação (AES-CTR)
void aes_x_ctr(uint8_t* out, uint8_t* in, size_t l,
    aes_t* aes) {
    // Nr = Número de rodadas
    const uint8_t Nr = aes->Nk + 6;
    // Ponteiro para contador
    uint8_t* ti = aes->c0;
    // Encriptação AES-CTR
    for(size_t i = 0; i < l; i = i + 16) {
        Cipher(out + i, ti, aes->ke, Nr);
        Xor(out + i, out + i, in + i);
        AddCounter(ti);
    }
}
```

Criptografia

- Criptografia simétrica
  - Possui implementação eficiente das operações em hardware e software, por utilizar de iterações, substituições, permutações e operações binárias
  - Sua construção permite que o uso repetitivo de chaves sem comprometer a segurança do sistema

Criptografia

- Criptografia simétrica
  - Possui implementação eficiente das operações em hardware e software, por utilizar de iterações, substituições, permutações e operações binárias
  - Sua construção permite que o uso repetitivo de chaves sem comprometer a segurança do sistema

Algoritmos: AES, DES, RC5, ...

- Criptografia simétrica
  - O padrão NIST FIPS 197[1], criado em 2001 para o *Advanced Encryption Standard* (AES), utiliza o algoritmo Rijndael que foi desenvolvido pelos criptólogos belgas Joan Daemen e Vincent Rijmen

[1] https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf

Criptografia

- Criptografia simétrica
  - O padrão NIST FIPS 197[1], criado em 2001 para o *Advanced Encryption Standard* (AES), utiliza o algoritmo Rijndael que foi desenvolvido pelos criptólogos belgas Joan Daemen e Vincent Rijmen
  - Neste padrão de criptografia simétrica são utilizados blocos de dados com tamanho fixo de 128 bits e chaves privadas com 128, 192 ou 256 bits

[1] https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf

Criptografia

- Criptografia simétrica
  - O padrão NIST FIPS 197[1], criado em 2001 para o *Advanced Encryption Standard* (AES), utiliza o algoritmo Rijndael que foi desenvolvido pelos criptólogos belgas Joan Daemen e Vincent Rijmen
  - Neste padrão de criptografia simétrica são utilizados blocos de dados com tamanho fixo de 128 bits e chaves privadas com 128, 192 ou 256 bits

Nenhum ataque se mostrou viável ainda

[1] https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf

Criptografia

- Criptografia simétrica

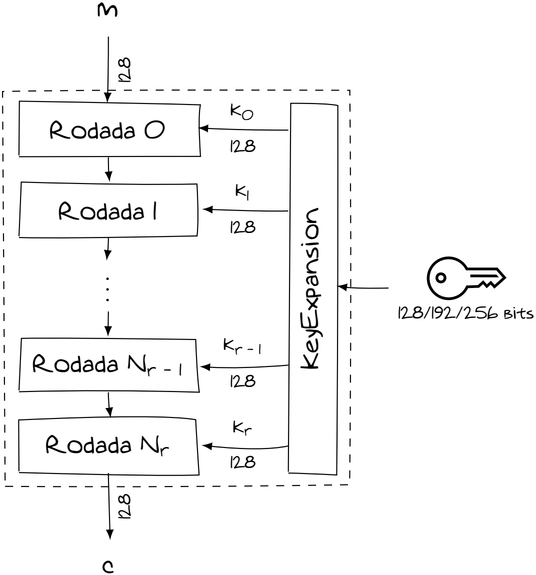
Tamanho da chave → Nk
Tamanho do bloco → NB
Número de rodadas → Nr

	Nk	NB	Nr
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

Palavras de 32 bits

Criptografia

- Criptografia simétrica



Criptografia

- Criptografia simétrica

```c
// Procedimento KeyExpansion
void KeyExpansion(uint8_t* out, uint8_t* in, uint8_t Nk) {
    const uint8_t Nr = Nk + 6; uint8_t temp[4];
    // Primeira rodada é a própria chave
    for(uint8_t i = 0; i < Nk; i++)
        WriteWord(out, i << 2, in, i << 2);
    // Gerando as rodadas a partir das anteriores
    for(uint8_t i = Nk; i < (Nr + 1) << 2; i++) {
        WriteWord(temp, 0, out, (i - 1) << 2);
        if(i % Nk == 0) {
            RotWord(temp); SubWord(temp);
            temp[0] = temp[0] ^ rcon[i / Nk];
        }
        else if(Nk > 6 && i % Nk == 4) SubWord(temp);
        WriteWordXor(out, i << 2, out, (i - Nk) << 2, temp);
    }
}
```

Criptografia

- Criptografia simétrica

```
// Procedimento de encriptação
void Cipher(uint8_t* c, const uint8_t* m, uint8_t* k,
    uint8_t Nr) {
    uint8_t state[4][4];
    ReadInput(state, m);
    AddRoundKey(state, k, 0);
    for(uint8_t i = 1; i < Nr; i++) {
        SubBytes(state);
        ShiftRows(state);
        MixColumns(state);
        AddRoundKey(state, k, i);
    }
    SubBytes(state);
    ShiftRows(state);
    AddRoundKey(state, k, Nr);
    WriteOutput(c, state);
}
```

Criptografia

- Criptografia simétrica

```
// Procedimento de decriptação
void Decipher(uint8_t* m, const uint8_t* c, uint8_t* k, uint8_t Nr) {
    uint8_t state[4][4];
    ReadInput(state, c);
    AddRoundKey(state, k, Nr);
    for(int8_t i = Nr - 1; i >= 1; i--) {
        InvShiftRows(state);
        InvSubBytes(state);
        AddRoundKey(state, k, i);
        InvMixColumns(state);
    }
    InvShiftRows(state);
    InvSubBytes(state);
    AddRoundKey(state, k, 0);
    WriteOutput(m, state);
}
```

Criptografia

- Criptografia simétrica
  - AddRoundKey

$$
\begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix}
=
\begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix}
+
\begin{bmatrix} ki_{0,0} & ki_{0,1} & ki_{0,2} & ki_{0,3} \\ ki_{1,0} & ki_{1,1} & ki_{1,2} & ki_{1,3} \\ ki_{2,0} & ki_{2,1} & ki_{2,2} & ki_{2,3} \\ ki_{2,0} & ki_{3,1} & ki_{3,2} & ki_{3,3} \end{bmatrix}
$$

Criptografia

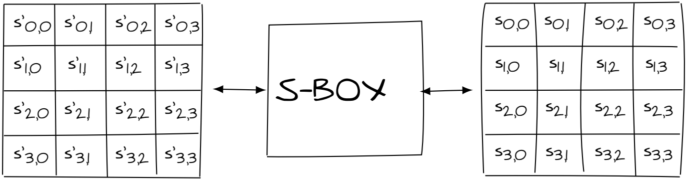
- Criptografia simétrica
  - AddRoundKey

$$\begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix} = \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} + \begin{bmatrix} ki_{0,0} & ki_{0,1} & ki_{0,2} & ki_{0,3} \\ ki_{1,0} & ki_{1,1} & ki_{1,2} & ki_{1,3} \\ ki_{2,0} & ki_{2,1} & ki_{2,2} & ki_{2,3} \\ ki_{2,0} & ki_{3,1} & ki_{3,2} & ki_{3,3} \end{bmatrix}$$

Na aritmética $GF(2^8)$, a adição
é equivalente ao ou-exclusivo bit a bit

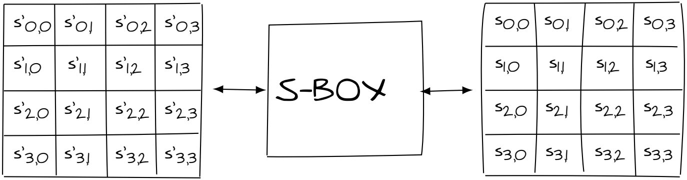
Criptografia

- Criptografia simétrica
  - SubBytes/InvSubBytes



Criptografia

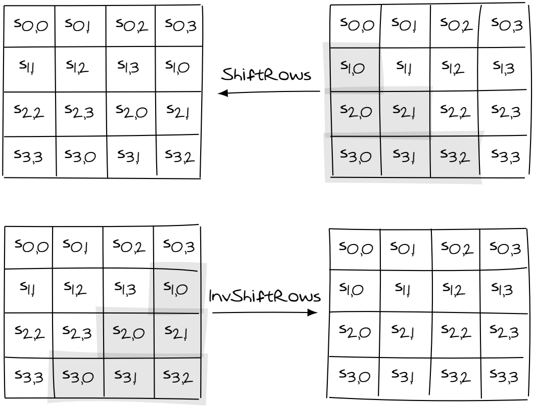
- Criptografia simétrica
  - SubBytes/InvSubBytes



$$s'[i,j] = s\text{-}BOX[s[i,j]]$$
$$s[i,j] = s\text{-}BOX^{-1}[s'[i,j]]$$

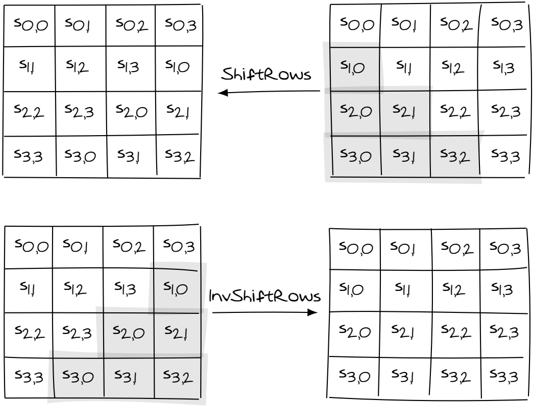
Criptografia

- Criptografia simétrica
  - ShiftRows/InvShiftRows



Criptografia

- Criptografia simétrica
  - ShiftRows/InvShiftRows



$$s = \mathrm{InvShiftRows}(\mathrm{ShiftRows}(s))$$

Criptografia

- Criptografia simétrica
  - MixColumns/InvMixColumns

$$
\overset{s'}{\begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,1} & s'_{1,2} & s'_{1,3} & s'_{1,0} \\ s'_{2,2} & s'_{2,3} & s'_{2,0} & s'_{2,1} \\ s'_{3,3} & s'_{3,0} & s'_{3,1} & s'_{3,2} \end{bmatrix}}
=
\overset{a}{\begin{bmatrix} 0x02 & 0x03 & 0x01 & 0x01 \\ 0x01 & 0x02 & 0x03 & 0x01 \\ 0x01 & 0x01 & 0x02 & 0x03 \\ 0x03 & 0x01 & 0x01 & 0x02 \end{bmatrix}}
\times
\overset{s}{\begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,1} & s_{1,2} & s_{1,3} & s_{1,0} \\ s_{2,2} & s_{2,3} & s_{2,0} & s_{2,1} \\ s_{3,3} & s_{3,0} & s_{3,1} & s_{3,2} \end{bmatrix}}
$$

$$
\overset{s}{\begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix}}
=
\overset{a^{-1}}{\begin{bmatrix} 0x0E & 0x0B & 0x0D & 0x09 \\ 0x09 & 0x0E & 0x0B & 0x0D \\ 0x0D & 0x09 & 0x0E & 0x0B \\ 0x0B & 0x0D & 0x09 & 0x0E \end{bmatrix}}
\times
\overset{s'}{\begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,1} & s'_{1,2} & s'_{1,3} & s'_{1,0} \\ s'_{2,2} & s'_{2,3} & s'_{2,0} & s'_{2,1} \\ s'_{3,3} & s'_{3,0} & s'_{3,1} & s'_{3,2} \end{bmatrix}}
$$

Criptografia

- Criptografia simétrica
  - MixColumns/InvMixColumns

$$\overset{s'}{\begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,1} & s'_{1,2} & s'_{1,3} & s'_{1,0} \\ s'_{2,2} & s'_{2,3} & s'_{2,0} & s'_{2,1} \\ s'_{3,3} & s'_{3,0} & s'_{3,1} & s'_{3,2} \end{bmatrix}} = \overset{a}{\begin{bmatrix} 0x02 & 0x03 & 0x01 & 0x01 \\ 0x01 & 0x02 & 0x03 & 0x01 \\ 0x01 & 0x01 & 0x02 & 0x03 \\ 0x03 & 0x01 & 0x01 & 0x02 \end{bmatrix}} \times \overset{s}{\begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,1} & s_{1,2} & s_{1,3} & s_{1,0} \\ s_{2,2} & s_{2,3} & s_{2,0} & s_{2,1} \\ s_{3,3} & s_{3,0} & s_{3,1} & s_{3,2} \end{bmatrix}}$$

$$\overset{s}{\begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix}} = \overset{a^{-1}}{\begin{bmatrix} 0x0E & 0x0B & 0x0D & 0x09 \\ 0x09 & 0x0E & 0x0B & 0x0D \\ 0x0D & 0x09 & 0x0E & 0x0B \\ 0x0B & 0x0D & 0x09 & 0x0E \end{bmatrix}} \times \overset{s'}{\begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,1} & s'_{1,2} & s'_{1,3} & s'_{1,0} \\ s'_{2,2} & s'_{2,3} & s'_{2,0} & s'_{2,1} \\ s'_{3,3} & s'_{3,0} & s'_{3,1} & s'_{3,2} \end{bmatrix}}$$

As multiplicações são realizadas em $GF(2^8)$

Criptografia

- Criptografia simétrica
  - Multiplicação $a(x) \times b(x) \bmod m(x)$ em $GF(2^8)$, com $m(x) = x^8 + x^4 + x^3 + x + 1$

$$
\begin{aligned}
a(x) \times b(x) &= 0x57 \times 0x83 \\
&= 0b01010111 \times 0b10000011
\end{aligned}
$$

Criptografia

- Criptografia simétrica
  - Multiplicação $a(x) \times b(x) \bmod m(x)$ em $GF(2^8)$, com $m(x) = x^8 + x^4 + x^3 + x + 1$

$$\begin{aligned} a(x) \times b(x) &= 0x57 \times 0x83 \\ &= 0b01010111 \times 0b10000011 \\ &= \left(x^6 + x^4 + x^2 + x + 1\right) \times \left(x^7 + x + 1\right) \end{aligned}$$

Criptografia

- Criptografia simétrica
  - Multiplicação $a(x) \times b(x) \bmod m(x)$ em $GF(2^8)$, com $m(x) = x^8 + x^4 + x^3 + x + 1$

$$
\begin{aligned}
a(x) \times b(x) &= 0x57 \times 0x83 \\
&= 0b01010111 \times 0b10000011 \\
&= \left(x^6 + x^4 + x^2 + x + 1\right) \times \left(x^7 + x + 1\right) \\
&= \left(x^{13} + \cancel{x^7} + x^6\right) + \left(x^{11} + x^5 + x^4\right) + \\
&\quad \left(x^9 + x^3 + \cancel{x^2}\right) + \left(x^8 + \cancel{x^2} + \cancel{x}\right) + \left(\cancel{x^7} + \cancel{x} + 1\right)
\end{aligned}
$$

Criptografia

- Criptografia simétrica
  - Multiplicação $a(x) \times b(x) \bmod m(x)$ em $GF(2^8)$, com $m(x) = x^8 + x^4 + x^3 + x + 1$

$$
\begin{aligned}
a(x) \times b(x) &= 0x57 \times 0x83 \\
&= 0b01010111 \times 0b10000011 \\
&= \left(x^6 + x^4 + x^2 + x + 1\right) \times \left(x^7 + x + 1\right) \\
&= \left(x^{13} + \cancel{x^7} + x^6\right) + \left(x^{11} + x^5 + x^4\right) + \\
&\quad \left(x^9 + x^3 + \cancel{x^2}\right) + \left(x^8 + \cancel{x^2} + \cancel{x}\right) + \left(\cancel{x^7} + \cancel{x} + 1\right) \\
&= x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1
\end{aligned}
$$

Criptografia

- Criptografia simétrica
  - Multiplicação $a(x) \times b(x)$ mod $m(x)$ em $GF(2^8)$, com $m(x) = x^8 + x^4 + x^3 + x + 1$

$$
\begin{aligned}
a(x) \times b(x) &= 0x57 \times 0x83 \\
&= 0b01010111 \times 0b10000011 \\
&= \left(x^6 + x^4 + x^2 + x + 1\right) \times \left(x^7 + x + 1\right) \\
&= \left(x^{13} + \cancel{x^7} + x^6\right) + \left(x^{11} + x^5 + x^4\right) + \\
&\quad \left(x^9 + x^3 + \cancel{x^2}\right) + \left(x^8 + \cancel{x^2} + \cancel{x}\right) + \left(\cancel{x^7} + \cancel{x} + 1\right) \\
&= x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1
\end{aligned}
$$

$$
x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1 \bmod x^8 + x^4 + x^3 + x + 1
$$

$$=$$

$$x^7 + x^6 + 1$$

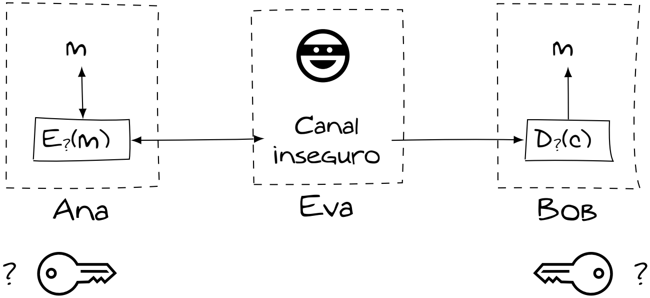
Criptografia

- Criptografia simétrica
  - Multiplicação $a(x) \times b(x) \bmod m(x)$ em $GF(2^8)$, com $m(x) = x^8 + x^4 + x^3 + x + 1$

```c
// Função MultiplyGF
uint8_t MultiplyGF(uint8_t a, uint8_t b) {
    // c(x) = 0, m(x) = x^4 + x^3 + x + 1
    uint8_t c = 0, m = 0x1B;
    // Enquanto b for maior que 0
    while(b > 0) {
        // b é impar (b[0] = 1) -> c(x) = c(x) + a(x)
        c = c ^ ((b & 1) * a);
        // Multiplica a(x) por 2
        // Overflow (a[7] = 1) -> a(x) mod m(x)
        a = (a << 1) ^ ((a >> 7) * m);
        // Divide b por 2
        b = b >> 1;
    }
    // c(x) = (a(x) + b(x)) mod m(x)
    return c;
}
```

Criptografia

- Criptografia simétrica
  - Como compartilhar uma chave será utilizada entre as partes quando não existe um canal seguro?



Criptografia

- Troca de chaves com Diffie-Hellman
  - É baseado no problema intratável do logaritmo discreto, ou seja, ainda não existe nenhum algoritmo computacionalmente eficiente para sua resolução

Criptografia

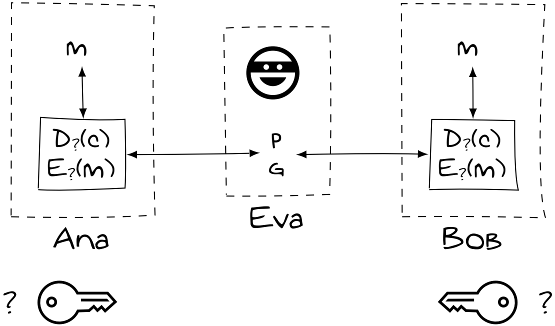
- Troca de chaves com Diffie-Hellman
  - É baseado no problema intratável do logaritmo discreto, ou seja, ainda não existe nenhum algoritmo computacionalmente eficiente para sua resolução
  - O cálculo da exponenciação b^n é rapidamente obtida com complexidade de $O(\log n)$

Criptografia

- Troca de chaves com Diffie-Hellman
  - É baseado no problema intratável do logaritmo discreto, ou seja, ainda não existe nenhum algoritmo computacionalmente eficiente para sua resolução
  - O cálculo da exponenciação b^n é rapidamente obtida com complexidade de $O(\log n)$
  - Entretanto, para se obter o número n a partir de b^n, é preciso calcular o logaritmo discreto $\log_b b^n = n$ que possui complexidade exponencial $O(2^n)$

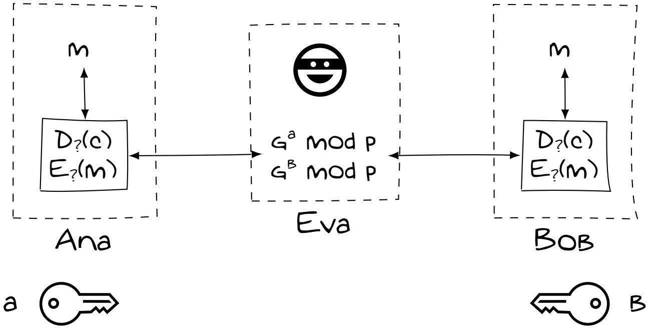
Criptografia

- Troca de chaves com Diffie-Hellman
  - Ana e Bob não se conhecem, mas concordam em utilizar uma base g e o número primo p, ambos públicos e transmitidos pelo canal inseguro



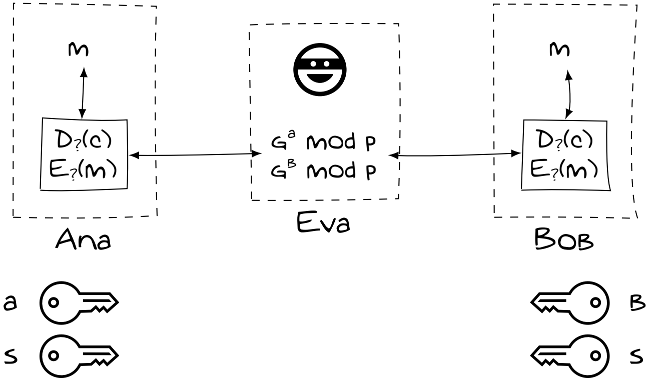
Criptografia

- Troca de chaves com Diffie-Hellman
  - Utilizando suas próprias chaves privadas, Ana envia para Bob $g^a \bmod p$ e Bob envia para Ana $g^b \bmod p$



Criptografia

- Troca de chaves com Diffie-Hellman
  - A chave compartilhada s é gerada por Ana $s = (g^b)^a \bmod p$ e por Bob $s = (g^a)^b \bmod p$



Criptografia

- Troca de chaves com Diffie-Hellman
  - Para que a atacante Eva seja capaz de obter o valor da chave privada compartilhada s, é preciso resolver eficientemente o problema do logaritmo discreto

Criptografia

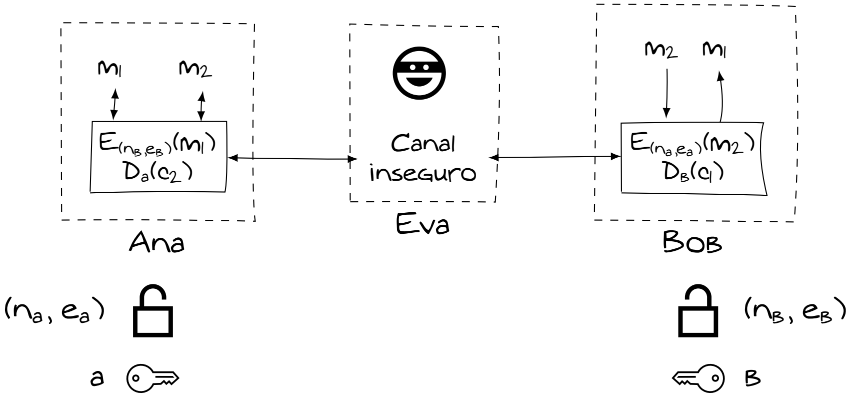
- Troca de chaves com Diffie-Hellman
  - Para que a atacante Eva seja capaz de obter o valor da chave privada compartilhada s, é preciso resolver eficientemente o problema do logaritmo discreto
  - Devem ser utilizados números primos de tamanho grande, com pelo menos 2.048 bits, para dificultar a recuperação das chaves privadas a ou b

Criptografia

- Troca de chaves com Diffie-Hellman
  - Para que a atacante Eva seja capaz de obter o valor da chave privada compartilhada s, é preciso resolver eficientemente o problema do logaritmo discreto
  - Devem ser utilizados números primos de tamanho grande, com pelo menos 2.048 bits, para dificultar a recuperação das chaves privadas a ou b
  - Esta técnica de troca de chave é vulnerável ao <u>ataque do homem do meio</u> e pode ser evitado com autenticação das partes envolvidas

Criptografia

- Criptografia assimétrica
  - Utiliza chaves privadas para decriptação e públicas para encriptação das mensagens



Criptografia

- Criptografia assimétrica
  - A criptografia de chave pública RSA é baseada no problema intratável de fatoração de números, gerando uma chave pública n através da multiplicação de dois números primos p e q, que precisam ser distintos e suficientemente grandes

Criptografia

- Criptografia assimétrica
  - A criptografia de chave pública RSA é baseada no problema intratável de fatoração de números, gerando uma chave pública n através da multiplicação de dois números primos p e q, que precisam ser distintos e suficientemente grandes
  - A chave pública (n, e) é obtida por $n = (p - 1) \times (q - 1)$ e pela geração de um número aleatório e ímpar e tal que $1 < e \leq n$ e $mdc\,(e, n) = 1$

Criptografia

- Criptografia assimétrica
  - A criptografia de chave pública RSA é baseada no problema intratável de fatoração de números, gerando uma chave pública n através da multiplicação de dois números primos p e q, que precisam ser distintos e suficientemente grandes
  - A chave pública (n, e) é obtida por $n = (p - 1) \times (q - 1)$ e pela geração de um número aleatório e ímpar e tal que $1 < e \leq n$ e $mdc\,(e, n) = 1$
  - Através da aplicação do inverso multiplicativo, chave privada d é gerada, onde $1 < d \leq n$ e $e \times d \equiv 1 \bmod n$

Criptografia

- Criptografia assimétrica
  - A encriptação utiliza a chave pública (n, e) e a mensagem m fornecidas, calculando $c = m^e \bmod n$
  - A mensagem c é decriptada aplicando a chave privada d em $m = c^d \bmod n$ para obter m

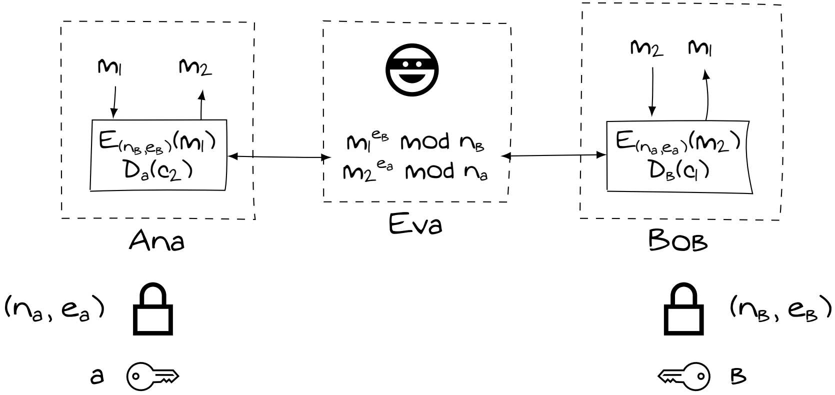
Criptografia

- Criptografia assimétrica
  - A encriptação utiliza a chave pública (n, e) e a mensagem m fornecidas, calculando $c = m^e \bmod n$
  - A mensagem c é decriptada aplicando a chave privada d em $m = c^d \bmod n$ para obter m

$$c^d \equiv \left(m^e\right)^d \equiv m \bmod n$$

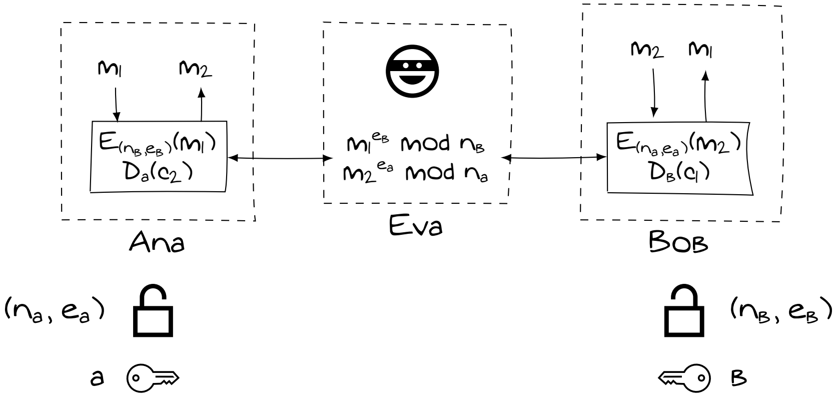
Criptografia

- Criptografia assimétrica



Criptografia

- Criptografia assimétrica



$$m_2 = (m_2^{e_a})^a \bmod n_a \qquad m_1 = (m_1^{e_B})^B \bmod n_B$$

Criptografia

- Criptografia assimétrica
  - Permite a autenticação das partes, uma vez que as chaves públicas são seguras e só podem ser decriptadas com a chave privada do proprietário

Criptografia

- Criptografia assimétrica
  - Permite a autenticação das partes, uma vez que as chaves públicas são seguras e só podem ser decriptadas com a chave privada do proprietário
  - Os algoritmos mais utilizados, como o Diffie-Hellman ou RSA, apresentam um custo computacional mais elevado em comparação aos algoritmos de criptografia simétricos, uma vez que demandam chaves com tamanho bem maiores, com pelo menos 2.048 bits para serem consideradas seguras

Exercício

- A empresa de tecnologia Poxim Tech está aplicando técnicas de criptografia em todos os seus sistemas de transmissão de dados, visando para proteger os dados de acessos não autorizados
  - Os dados transmitidos entre as partes são representados por bytes no formato hexadecimal
  - A criptografia simétrica AES é aplicada no modo ECB com chaves de 128, 192 ou 256 bits
  - No compartilhamento das chaves privadas é utilizado o Diffie-Hellman com parâmetros de até 2.048 bits

Exercício

- Formato do arquivo de entrada
  - *Número de operações* (n)
    - *dh a b g p*
    - *d c*
    - *e m*

```
3
dh␣2F333D84630F102FDA0B594D4FF7CA46␣FBDB83740FB1D83EE4
    15C34725D377FF␣C54B073C6A2B3745AEAC545F8493439A568
    BBF2902BE07D20A359A20A9BBD26E06DAAA7005E2B5B48E091
    3129C57ACF2E26B1BE42923B63358505401OB266F11␣1219C9
    43937D661A8CA99AA1DC0CCBC2D28018D60CAB90A8D9097BC5
    981C99AA3662EEC9DF54E36CFD7D0DD98AD99B5C59B332655F
    C20E38CB89FE63A5970EDB
d␣F0FA40FAF0F0CA
e␣00112233445566778899AABBCCDDEEFF50C0440
```

Exercício

- Formato do arquivo de saída
  - Para cada comando é exibido seu resultado

```
s=E8613C49876806B074535ACF62DD673D
m=C1953BF75E2AA86EA5C94B7C4345FCE7
c=29C9E347235ACEDCC86B6F7A5791438960FC24F37C4A262492C6F79D12719231
```