INSTITUTO FEDERAL DE MATO GROSSO - IFMT

BEATRIZ APARECIDA DUTRA
NAMEM RACHID JAUDY

RELATÓRIO TÉCNICO: IMPLEMENTAÇÃO DO ALGORITMO RSA

Trabalho avaliativo da disciplina de Segurança em Sistemas de Computação

CUIABÁ - MT
2025

Sumário

1 Introdução

A segurança da informação é um pilar fundamental da computação moderna, garantindo a confidencialidade, integridade e disponibilidade dos dados. Um dos maiores desafios históricos da criptografia foi o problema da distribuição de chaves: como duas partes podem trocar informações de forma segura sem terem previamente acordado uma chave secreta? A solução surgiu com a **Criptografia Assimétrica**, também conhecida como criptografia de chave pública.

Diferente da criptografia simétrica, que utiliza uma única chave para cifrar e decifrar, a criptografia assimétrica utiliza um par de chaves matematicamente relacionadas: uma pública (para cifrar) e uma privada (para decifrar). O algoritmo **RSA** (Rivest-Shamir-Adleman) é o exemplo mais clássico e amplamente utilizado desse modelo.

Este trabalho tem como objetivo a implementação prática do algoritmo RSA, conforme especificado nos requisitos da avaliação. O projeto abrange desde a base matemática, com a geração de números primos através do método do **Crivo de Eratóstenes**, até a aplicação das fórmulas de cifragem e decifragem em mensagens textuais, convertidas previamente para representação hexadecimal.

O código-fonte completo desta implementação está disponível publicamente no GitHub através do link: https://github.com/Namem/atividade_criptografia_assimetrica_rsa.

2 Fundamentação Teórica

2.1 O Algoritmo RSA

O RSA baseia sua segurança na dificuldade computacional de fatorar o produto de dois grandes números primos. O processo matemático do RSA ocorre nas seguintes etapas:

1. **Geração de Chaves:**
    - Escolhem-se dois números primos distintos, p e q.
    - Calcula-se o módulo $n = p \times q$. O valor de n é utilizado em ambas as chaves.
    - Calcula-se a Função Totiente de Euler: $\phi(n) = (p-1)(q-1)$.

- Escolhe-se um inteiro e (chave pública) tal que $1 < e < \phi(n)$ e que e seja coprimo de $\phi(n)$.
- Calcula-se d (chave privada), tal que d seja o inverso multiplicativo modular de e:

$$d \times e \equiv 1 \pmod{\phi(n)}$$

2. **Cifragem e Decifragem:** Para uma mensagem M (onde $M < n$), a cifragem resulta no texto cifrado C [3]:

$$C = M^e \pmod{n}$$

A decifragem recupera a mensagem original M [3]:

$$M = C^d \pmod{n}$$

2.2 O Crivo de Eratóstenes

Para a implementação do RSA, é pré-requisito a capacidade de gerar números primos. O Crivo de Eratóstenes é um algoritmo eficiente para encontrar todos os números primos até um limite n.

O método consiste em criar uma lista de inteiros de 2 até n. Iterativamente, seleciona-se o primeiro número não marcado (primo) e eliminam-se todos os seus múltiplos da lista. A eficiência deste algoritmo é de aproximadamente $O(n \log \log n)$, tornando-o ideal para a geração das chaves neste projeto.

2.3 Representação Hexadecimal

A notação hexadecimal (base 16) é padrão em segurança da informação. O trabalho exige a conversão da mensagem original para valores hexadecimais antes da cifragem. Isso permite visualizar os dados brutos (bytes) que serão submetidos às operações matemáticas.

3 Descrição da Implementação

A solução foi desenvolvida na linguagem **Python**, escolhida devido ao suporte nativo a inteiros de precisão arbitrária, essencial para as exponenciações modulares.

3.1 Estrutura do Código

O software foi estruturado de forma modular:

- `gerar_crivo_eratostenes(limite)`: Implementa a eliminação de múltiplos para retornar uma lista de primos válidos.
- `selecionar_chaves()`: Seleciona aleatoriamente p e q (maiores que 70) da lista do Crivo, calcula $\phi(n)$, escolhe e aleatoriamente entre os coprimos e calcula d pelo inverso modular.
- `converter_para_hex()`: Transforma a string de entrada em inteiros e exibe a representação visual (ex: `0x4A`).
- `cifrar_rsa` e `decifrar_rsa`: Aplicam as funções de potência modular (`pow(base, exp, mod)`).

4 Resultados Experimentais

Abaixo são apresentados os logs de execução do sistema, demonstrando o funcionamento correto das etapas de geração, cifragem e decifragem.

4.1 Cenário de Teste 1: Geração de Chaves

O sistema utilizou o Crivo para filtrar primos e selecionou dois candidatos. O módulo n resultante (31373) é suficiente para cobrir a tabela ASCII.

```
[Sistema] Iniciando geracao de chaves criptograficas...
 -> Primos escolhidos (via Crivo): p=137, q=229
 -> Modulo n: 31373
 -> Phi(n): 31008

[Sucesso] Chaves Geradas:
```

```
>> Publica (e, n): (53, 31373)
>> Privada (d, n): (19301, 31373)
```

Listing 1: Log de Geração das Chaves

4.2 Cenário de Teste 2: Cifragem de Mensagem

Foi utilizada a mensagem "Ola". O sistema converteu para hexadecimal e gerou o vetor cifrado. Nota-se que o vetor numérico não guarda semelhança visual com a mensagem original.

```
[Entrada do Usuario] Digite a mensagem de texto: Ola

[Conversao] Mensagem Original: 'Ola'
[Conversao] Representacao Hex:  0x4F 0x6C 0x61

[Criptografia] Resultado (Vetor Cifrado): [28660, 29013, 30372]
```

Listing 2: Log de Cifragem

4.3 Cenário de Teste 3: Descriptografia

Utilizando a chave privada ($d = 19301$), o sistema reverteu a operação matemática, recuperando a integridade da mensagem.

```
[Descriptografia] Iniciando processo...
[Descriptografia] Mensagem Original Restaurada: 'Ola'
```

Listing 3: Log de Descriptografia

5 Conclusão

O desenvolvimento deste trabalho permitiu uma compreensão prática dos mecanismos da criptografia assimétrica. A implementação do algoritmo RSA evidenciou a elegância da aritmética modular e a importância de algoritmos auxiliares, como o Crivo de Eratóstenes, para a geração de parâmetros de segurança.

Conclui-se que o software atendeu a todos os requisitos propostos: geração autônoma de primos, cálculo correto de chaves e a cifragem/decifragem bem-sucedida com visualização hexadecimal.

6 Referências Bibliográficas

1. STALLINGS, William. *Cryptography and Network Security: Principles and Practice*. 7ª Ed. Pearson, 2017.

2. RIVEST, R.; SHAMIR, A.; ADLEMAN, L. *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems*. Communications of the ACM, Vol. 21, No. 2, 1978.

3. Material da Disciplina: *Trabalho de Criptografia Assimétrica com RSA*. PDF Disponibilizado no AVA. 2025.