Trabalho de segurança computacional - 2023-1

ALUNA ANA CAROLINE DA ROCHA BRAZ - 212008482

ALUNO MATHEUS BARBOSA E SILVA - 190113987

AES (Advanced Encryption Standard)

É um algoritmo de criptografia simétrica amplamente utilizado e considerado uma das principais escolhas para a criptografia de dados. Ele substituiu o antigo algoritmo DES (Data Encryption Standard) devido à sua segurança e desempenho aprimorados. O AES opera no modo de criptografia de bloco, o que significa que ele criptografa e descriptografa dados em blocos fixos de tamanho fixo. O tamanho do bloco padrão do AES é de 128 bits (16 bytes). No entanto, o AES também suporta tamanhos de chave de 128 bits, 192 bits e 256 bits.

PASSO A PASSO

CHAVE

```
É necessário um chave secreta compartilhada entre o remetente e o destinatário para criptografar e descriptografar os dados.
```

EXPANSÃO DE CHAVE

```
A chave original passa por uma etapa de expansão de chave para gerar uma série de chaves de rodada que serão usadas nas iterações d
```

CRIPTOGRAFIA

```
O algoritmo AES executa várias rodadas (10, 12 ou 14, dependendo do tamanho da chave) de substituição de bytes, permutação de linhas
```

DESCRIPTOGRAFIA

```
O processo de descriptografia do AES é semelhante à criptografia, mas envolve a aplicação das operações inversas nas etapas de cript
```

IMPLEMENTAÇÃO

A main do projeto fica no App.java, sendo assim para rodar basta digitar no terminal: `java App.java`

Miller-Rabin

O algoritmo de Miller-Rabin é um teste probabilístico de primalidade utilizado para determinar se um número é composto ou provavelmente primo. O algoritmo de Miller-Rabin é baseado no chamado "testemunho de Fermat". Se um número composto n passar no teste de Miller-Rabin, ele é considerado um "falso testemunho de Fermat"

PASSO A PASSO

Dado um número ímpar n a ser testado, escreve-se n-1 como 2^r * d, onde d é um número ímpar.

```
Para cada iteração i (de 1 a k, onde k é o número de iterações desejadas):
    Escolhe-se um número aleatório a entre 2 e n-2.
    Computa-se x = a^d mod n.
    Se x for igual a 1 ou n-1, passa-se para a próxima iteração.
    Para j variando de 1 a r-1:
        Computa-se x = x^2 mod n.
        Se x for igual a n-1, passa-se para a próxima iteração.
        Se x for igual a 1, o número é considerado composto.
    Se nenhum dos testes anteriores for satisfeito, o número é considerado composto.
```

Se após todas as iterações o número não for considerado composto, ele é provavelmente primo.

RSA (Rivest, Shamir e Adleman)

RSA é um algoritmo de cifração assimétrica. A chave de cifração RSA é gerada aleatoriamente. A cifração RSA é implementada utilizando o algoritmo de exponenciação modular.

PASSO A PASSO

Gere chaves RSA:

```
Escolha dois números primos grandes, p e q.
Calcule o produto n = p * q, que será o módulo para as operações de criptografia e descriptografia.
Calcule a função totiente de Euler ϕ(n) = (p - 1) * (q - 1).
Escolha um número inteiro e relativamente primo a ϕ(n), chamado de e, para ser a chave pública de criptografia.
Calcule o inverso multiplicativo de e módulo ϕ(n), chamado de d, para ser a chave privada de descriptografia.
```

Implemente as funções de criptografia e descriptografia RSA:

```
A função de criptografia recebe a mensagem original como entrada e retorna a mensagem criptografada.
A mensagem criptografada é calculada elevando a mensagem original à potência e módulo n.
A função de descriptografia recebe a mensagem criptografada como entrada e retorna a mensagem original.
A mensagem original é calculada elevando a mensagem criptografada à potência d módulo n.
```

Implemente as funções de assinatura e verificação RSA:

```
A função de assinatura recebe a mensagem original e a chave privada como entrada e retorna a assinatura.
A assinatura é calculada elevando a mensagem original à potência d módulo n.
A função de verificação recebe a mensagem original, a assinatura e a chave pública como entrada e retorna um valor booleano indican
Para verificar a assinatura, a mensagem original é calculada elevando a assinatura à potência e módulo n e comparando o resultado c
```

Implemente funções auxiliares:

```
Função para converter uma string em um número inteiro para a mensagem original.
Função para converter um número inteiro em uma string para a mensagem original.
```

Teste sua implementação:

```
Gere um par de chaves RSA.
Criptografe e descriptografe uma mensagem para verificar se a implementação funciona corretamente.
Assine uma mensagem com a chave privada e verifique a assinatura usando a chave pública.
```

OAEP (Optimal Asymmetric Encryption Padding)

OAEP (Optimal Asymmetric Encryption Padding) é um esquema de preenchimento (padding) assimétrico ótimo utilizado em criptografia assimétrica, especialmente com algoritmos de chave pública como RSA. O objetivo do OAEP é aumentar a segurança da criptografia assimétrica, tornando-a resistente a certos tipos de ataques. O OAEP é usado para preencher os dados antes de criptografá-los usando uma chave pública. Isso é necessário porque os algoritmos de criptografia assimétrica, como o RSA, têm restrições no tamanho máximo dos dados que podem ser criptografados diretamente.

PASSO A PASSO

O OAEP consiste em duas principais etapas de preenchimento: preenchimento de mensagem e máscara aleatória.

```
Preenchimento de mensagem:
    Os dados a serem criptografados são preenchidos com zeros para alcançar um tamanho específico.
    É adicionada uma sequência aleatória de bits (denominada "salmoura" ou "salt") aos dados. Essa sequência serve para tornar a cri
    Os dados preenchidos são combinados com a salmoura usando uma função de mistura (normalmente uma função de hash segura, como SHA

Máscara aleatória:
    Uma sequência aleatória de bits é gerada.
    A combinação dos dados preenchidos e a salmoura da etapa anterior é combinada com a sequência aleatória de bits usando uma opera
    O resultado final é o dado de entrada para o algoritmo de criptografia assimétrica (por exemplo, RSA).
```

Ao descriptografar os dados criptografados com OAEP, as etapas são executadas em ordem inversa. O preenchimento é removido e a salmoura é separada dos dados antes de prosseguir com a descriptografia.

IMPLEMENTAÇÃO

O algoritmo não implementado com o uso de arquivos, apenas print na tela, portanto utilizando o terminal digite: `./main.py`

No terminal irá pedir para que escreva a mensagem que deseja ser encriptada e decriptada, sendo esses dois feito de forma automático Além disso o código verifica e faz assinatura do RSA.