Projeto 2 - Segurança Computacional

Gerador e **verificador de assinaturas RSA** em **ARQUIVOS**

Matricula - **180107992**
Nome: **Pedro Braga**

AES-CTR conhecido por seu nome original **Rijndael.**

É utilizado para criptografia de dados eletrônicos e foi estabelecida pelo instituto nacional de padrões e tecnologia dos E.U.A. (*NIST*) em 2001.

Parte I: Geração de chaves e cifra simétrica

a) Geração de chaves (p e q primos com no mínimo de 1024 bits)

Seleciona dois números primos aleatórios grandes para iniciação do RSA.

b) Geração de chave simétrica de sessão

Gera chave de sessão com 128-bits(16-bytes) e é utilizada para a cifra com o AES.

c) Cifração simétrica de mensagem (AES modo CTR)

Leia-se o arquivo texto e é criptografado sua mensagem.

Gera a criptografia simétrica descrita na parte 2.2 utilizando como chave a chave de sessão.

Além da criptografia ele é convertido em BASE64 e seu output é inserido no arquivo file.txt.aes.

d) Cifração assimétrica da chave de sessão, usando OAEP.

Utiliza a chave de sessão para criação das chaves públicas e privadas, criptografando a chave de sessão com a chave pública.

Parte II: Assinatura

a) Cálculo de hashes da mensagem em claro (função de hash SHA-3)

É gerado o hash baseado na mensagem do arquivo texto.

Para verificar-se integridade.

b) Assinatura da mensagem (cifração do hash da mensagem) - com a chave privada

Criptografa-se o hash gerado com uma chave privada.

c) Formatação do resultado (caracteres especiais e informações para verificação em BASE64)

É convertido para a BASE64 a assinatura.

Parte III: Verificação:

a) Parsing do documento assinado(de acordo com a formatação usada, no caso BASE64)

A assinatura em BASE64 é convertido novamente para a mensagem normal.

Durante a decifração da mensagem na letra b), a função decrypt faz o parsing da BASE64.

b) Decifração da assinatura (decifração do hash)

É feito a decifração da assinatura, e a chave de sessão é utilizada para decifrar o arquivo file.txt.aes que foi criptografado.

c) Verificação (cálculo e comparação do hash do arquivo)

É feito a verificação de igualdade para verificar a integridade do arquivo, se o hash for igual ao hash feito da mensagem pelo receptor, isso garante a integridade da troca de mensagens.

Funções:

- key_expansion - Gera 44 palavras baseado na chave inicial.
- add_round_key -
- shift_rows -
- mix_columns -

- encrypt - Aplica todas as funções descritas em 10 rounds iniciando com o add_round_key seguindo de
  - sub_bytes
  - shift_rows
  - mix_columns
  - add_round_key
- e em seu último round não é utilizado o mix_columns.

- decrypt - Aplica todas as funções descritas em 10 rounds e em seu último round não é utilizado o mix_columns.
  - inv_sub_bytes.
  - inv_shift_rows.
  - inv_mix_columns.
  - add_round_key.

  E no último round não é utilizado o mix_columns

RSA

Utiliza o algoritmo de Diffie-Helman.

1) Se escolhe dois números primos grandes. (p, q)

2) É calculado o produto dos dois números do passo anterior como (n=pq).

3) Calcular a função totiente de Euler (totient = (p-1)*(q-1))

4) É escolhido um número e que é coprimo de N, utilizando o algoritmo de euclides.

5) É escolhido d baseado ed = 1 (mod totient) na função modular.

   d é um coprimo positivo menor que o totient e é utilizado como uma chave privada. Utilizado o algoritmo de euclides extendido.

6) É criptografado (m = message) utilizando (c = m ^ e mod n).

7) É feito a decifração (m = message) utilizando decrypt = c ^ d (mod n).