CIC0201 - Segurança Computacional – 2022/1

Prof. João Gondim

Trabalho de Implementação 2 - Gerador/Verificador de Assinaturas

Neste trabalho, deve-se implementar um gerador e verificador de assinaturas RSA em arquivos. Assim, deve-se implementar um programa com as seguintes funcionalidades:

- Parte I: Geração de chaves
  1. Geração de chaves (p e q primos com no mínimo de 1024 bits)
- Parte II: Cifra simétrica
  1. Geração de chaves simétrica
  2. Cifração simétrica de mensagem (AES modo CTR),
- Parte III: Geração da assinatura
  1. Cálculo de hashes da mensagem em claro (função de hash SHA-3)
  2. Assinatura da mensagem (cifração do hash da mensagem usando OAEP)
  3. Formatação do resultado (caracteres especiais e informações para verificação em BASE64)
- Parte IV: Verificação:
  1. Parsing do documento assinado e decifração da mensagem (de acordo com a formatação usada, no caso BASE64)
  2. Decifração da assinatura (decifração do hash)
  3. Verificação (cálculo e comparação do hash do arquivo)

Observações:

1. Permite-se a utilização de bibliotecas públicas para aritmética modular e função de hash.
2. Não é permitida a utilização de bibliotecas públicas, como OpenSSL, para primitivas de criptográficas de cifração e decifração simétrica e assimétrica, bem como de geração de chaves.
3. A pontuação máxima será conferida aos trabalhos que realmente implementarem as seguintes primitivas:
   a. geração de chaves com teste de primalidade (Miller-Rabin)
   b. geração de chave e cifração AES
   c. cifração e decifração RSA com OAEP
   d. formatação/parsing
   e. AES modo CTR
4. A avaliação será mediante apresentação do trabalho, com a verificação das funcionalidades e inspeção do código.
5. Implementação preferencialmente individual, podendo ser em dupla. Linguagens preferenciais C, C++, Java e Python.

O que deve ser entregue: o código fonte e seu executável, descritivo (4 pg max), do AES-CTR, da assinatura RSA e do programa. Data de Entrega: 19/09/2022, por email até 09:59h. Apresentações: 19 e 21/09/2022

Links úteis:

https://crypto.stanford.edu/pbc/notes/numbertheory/millerrabin.html

https://www.ijser.org/researchpaper/Implementation-of-Advanced-Encryption-Standard-Algorithm.pdf;

https://cseweb.ucsd.edu/~mihir/papers/oaep.pdf