UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

EXERCÍCIOS DE CRIPTOGRAFIA SIMÉTRICA, HASH, MAC, PBKDF E CRIPTOGRAFIA AUTENTICADA EM JAVA

Ivo Guilherme Kurtz Bohm
Sadi Júnior Domingos Jacinto

Professora orientadora: Carla Merkle Westphall

Florianópolis
2020

1 QUESTÕES

1. Abra o **projeto2CodigoLivro** e teste o seu funcionamento. Responda:

   1.1. Qual algoritmo é usado no código? Em qual modo?
   Resposta: O algoritmo usado é o *AES* no modo *CBC*.

   1.2. Explique o que faz o método *generateKey* da classe `https://docs.oracle.com/javase/7/docs/api/javax/crypto/KeyGenerator.html.KeyGenerator`
   Resposta: Gera uma chave simétrica, podendo a chave ser gerada independente de um algoritmo ou de maneira específica de um algoritmo.

   1.3. Explique como são usados os métodos *init*, *update* e *doFinal* para cifrar e para decifrar os dados nesse código. Leia a documentação e entenda bem o funcionamento desses métodos.
   Resposta:

   - ***init*:** Inicializa a cifra com uma chave e um conjunto de parâmetros de algoritmo, podendo ser inicializada para uma das quatro seguintes operações: criptografia, decodificação, embalagem da chave ou desembrulhamento da chave, dependendo do valor do parâmetro *opmode*.
     No exemplo, a cifra foi inicializada usando **DECRYPT_MODE** (decifrar), com uma chave gerada previamente e um IV aleatório.
   - ***update*:** Usado para continuar uma operação de criptografia ou decriptação de múltiplas partes (dependendo de como a cifra foi inicializada), processando outra parte de dados. Retorna o número de *bytes* armazenados na saída.
   - ***doFinal*:** Finaliza a operação de criptografia ou decriptografia de múltiplas partes, dependendo de como a cifra foi inicializada. Os dados de entrada que podem ter sido armazenados em *buffer* durante uma operação de atualização anterior são processados. Ao terminar, este método reinicia a cifra para o estado em que se encontrava inicialmente através de uma chamada para o init. Ou seja, o objeto é reinicializado e está disponível para criptografar ou decodificar (dependendo do modo de operação que foi especificado na chamada ao *init*) mais dados.

3. Nesse projeto você irá programar dois sistemas de decifragem, um usando o AES em **modo CBC** e outro usando o AES no **modo contador** (*counter mode* – CTR). Em ambos os casos um IV de 16 bytes é escolhido de forma aleatória. Para o modo CBC use o esquema de padding PKCS5. Para o modo CTR use NoPadding.

   Inicialmente iremos testar apenas a função de decifragem. Use o projeto3Aes para auxiliar a responder as questões. Nas questões seguintes você recebe uma chave AES, um IV e um texto cifrado (ambos codificados em hexa) e o seu objetivo é recuperar o texto plano/texto decifrado. A resposta de cada questão é o texto decifrado (frase legível).

   3.1.
   - Chave CBC: 53efb4b1157fccdb9902676329debc52
   - IV: d161fbaa4c64ecf7d2c4abd885751273

- Texto cifrado em modo CBC: 701f7fa45d9bb922c3cb15a519ba40ede1769eb753650886d6e69ebcad9c2816002679896a65a921d25e00793078474e3dbeca9a2838031c490e5ae9d1ea143f

Resposta: Modo CBC (Cipher Block Chaining) do AES - cifragem encadeada.

3.2.
- Chave CTR: a05e2679204241af07f6857d150a1fcd
- IV: 468ce1126a37b07138e78eab48344712
- Texto cifrado em modo CTR: 36466b5fddcfcb1b8a9479eb8c489e7139a3c35020b1e5ee808b39ff18b6abd812afe7dbbca40e15df391a7c07ece1c8e10a49368b86a946c8379cd8fa01a47f1956671144b0ca18a4c812cde8f7b9

Resposta: Modo CTR (counter) - cifra a contagem do IV e faz XOR com bloco de texto plano.

4. Crie um programa que recebe duas strings pelo teclado, calcula o hash (resumo criptográfico) e o MAC de cada uma das strings escrevendo o resultado na tela. Teste e explique o funcionamento do programa com entrada de strings iguais e depois com entrada de strings diferentes.
Resposta: Como é uma função de cáculo de *hash*, entradas iguais sempre geram saídas iguais e entradas diferentes sempre geram saídas diferentes. Isso pode ser observado durante a execução do código de acordo com os *prints*.[1]

OBS: As questões 2, 5 e 6 não se encontram neste relatório por se tratarem de questões práticas de implementação que não possuem nenhuma pergunta.

[1] No cálculo do *hash* e do MAC foi usado o SHA512