PUC-Rio – Departamento de Informática
Bacharelado em Informática
Segurança da Informação
Prof.: Anderson Oliveira da Silva

Trabalhos de Laboratório

1. Construir um programa Java utilizando a JCA que recebe um texto na linha de comando e assina o mesmo. O processo de geração da assinatura e verificação da mesma deve ser feito **sem a utilização da classe Signature**, detalhando-se na saída padrão cada um dos passos executados, inclusive apresentando o digest e a assinatura no formato hexadecimal. A classe *MySignature* deve ser implementada com os métodos *initSign*, *update*, *sign*, *initVerify* e *verify* com funcionalidades equivalentes aos respectivos métodos da classe Signature. O programa fonte deve ser enviado como anexo via e-mail com o título INF1416 – TrabLab 1 – Grupo N, onde N identifica o número do grupo. Prazo de entrega: 26/9/2016 - 11:59h.

2. Construir um programa Java utilizando a JCA que não use interface gráfica e que seja executado em uma linha de comando com argumentos, da seguinte forma:

DigestCalculator <SP> Tipo_Digest <SP>Caminho_ArqListaDigest <SP>Caminho_Arq1... <SP>Caminho_ArqN

onde,
Tipo_Digest – Tipo do digest a ser calculado (MD5 ou SHA1)
Caminho_ArqListaDigest - Informa a localização do arquivo que contém uma lista de digests conhecidos para arquivos.
Caminho_Arq1 Caminho_Arq2 ... Caminho_ArqN - Informa a localização dos N arquivos que devem ser processados.
<SP> - Caractere espaço em branco.

O arquivo com a lista de digests utiliza o formato ASCII e é formado por zero ou mais linhas formatadas da seguinte maneira:

Nome_Arq<SP>Tipo_Digest<SP>Digest_Hex[<SP>TipoDigest<SP>Digest_Hex]<EOL>

onde,
Nome_Arq - Nome de um arquivo qualquer, sem informar o caminho.
TipoDigest - Indica o digest em seguida (MD5 ou SHA1).
Digest_Hex - Digest em hexadecimal com base no tipo de digest especificado anteriormente.
<SP> - Caractere espaço em branco.
<*EOL*> – Caractere que marca o fim de linha (\n).
[] – O segundo digest pode ou não existir para um nome de arquivo.

O programa deve executar o seguinte procedimento:
1 - Calcular o digest solicitado do conteúdo de cada um dos N arquivos fornecidos;
2 - Comparar os digests calculados com os respectivos digests presentes no arquivo ArqListaDigest, se existirem, e com os digests dos arquivos da linha de comando;

3 - Imprimir na saída padrão uma lista com o seguinte formato:

Nome_Arq1<SP>Tipo_Digest<SP>Digest_Hex_Arq1<SP>(STATUS)
Nome_Arq2<SP>Tipo_Digest<SP>Digest_Hex_Arq2<SP>(STATUS)
. . . . . .
Nome_ArqN<SP>Tipo_Digest<SP>Digest_Hex_ArqN<SP>(STATUS)

onde:
<SP> - Caracter espaço em branco.
Nome_Arq1 .. Nome_ArqN - Correspondem aos nomes dos arquivos fornecidos na linha de comando, descartando o caminho.
Tipo_Digest – Tipo do digest calculado (MD5 ou SHA1)
Digest_Hex_ArqN – Digest formatado em hexadecimal calculado para o arquivo N.
STATUS - Corresponde a um dos status definidos abaixo:
OK = Status do arquivo cujo digest calculado é igual ao digest fornecido no arquivo ArqListaDigest e não colide com o digist de outro arquivo na linha de comando.
NOT OK = Status do arquivo cujo digest não é igual ao digest fornecido no arquivo ArqListaDigest e não colide com o digist de outro arquivo na linha de comando..
NOT FOUND = Status do arquivo cujo digest não foi encontrado no arquivo ArqListaDigest e não colide com o digest de outro arquivo na linha de comando.
COLISION = Status do arquivo cujo digest calculado colide com o digest de outro arquivo de nome diferente encontrado no arquivo ArqListaDigest ou com o digest de um dos arquivos fornecidos na linha de comando.

4 - Os digests calculados para os arquivos com status NOT FOUND devem ser acrescentados no final de uma linha existente para um nome de arquivo ou no final do arquivo de lista de digests para um nome de arquivo não existente, mantendo seu formato padrão. Os digests calculados para os arquivos com status COLISION não devem ser acrescentados no arquivo de lista de digests.

Observação 1: O nome do programa executável deve ser DigestCalculator.

Observação 2: O código fonte deve ser compilado com o Sun JDK 1.7 ou 1.8.

Observação 3: Utilize o método *update(byte[] input, int offset, int len)* da classe *MessageDigest* que atualiza o digest utilizando o array de bytes *input*, iniciando em *offset*.

Observação 4: Se os argumentos da linha de comando forem omitidos ou insuficientes para a execução do programa, deve-se imprimir uma mensagem com a orientação de execução e, em seguida, o programa deve ser encerrado.

O programa fonte deve ser enviado como anexo via e-mail com o título INF1416 – TrabLab 2 – Grupo N, onde N identifica o número do grupo. Prazo de entrega: 28/9/2016 - 11:59h.