Universidade da Beira Interior
Faculdade de Engenharia
Departamento de Informática

Segurança Informática

Guia para Aula Laboratorial 6

1º Ciclo em Engenharia Informática
1º Ciclo em Informática Web
1º Ciclo em Matemática e Aplicações

Computer Security

Guide for Laboratory Class 6

B.Sc. in Computer Science and Engineering
B.Sc. in Web Informatics
B.Sc. in Mathematics and Applications

Sumário

Implementação de um programa em C recorrendo à biblioteca de funções criptográficas fornecida pelo OpenSSL. Implementação de um pequeno programa em Java para cálculo do valor do *Secure Hash Algorithm 1* (SHA1) para um ficheiro, recorrendo à *Java Cryptography Architecture* (JCA).

Summary

Implementation of program written in C programming language resorting to the library of cryptographic functions provided by OpenSSL. Implementation of a small program in Java to calculate the value of the Secure Hash Algorithm 1 (SHA1) of a given file, resorting to the Java Cryptography Architecture (JCA).

Pré-requisitos:

Algumas das tarefas propostas a seguir requerem o acesso a um sistema com compiladores de programas escritos nas linguagens de programação C e Java, e que disponibilizem as bibliotecas OpenSSL e Java Cryptography Architecture (JCA). Sugere-se, assim, o uso de uma distribuição comum de Linux, onde todas estas condições estarão provavelmente preenchidas.

1 Implementação em C do Modo ECB de uma Cifra usando OpenSSL

Implementation of the ECB Mode of a Block Cipher using OpenSSL

O OpenSSL fornece um conjunto de funções em linguagem C para implementação de aplicações que necessitem de mecanismos criptográficos com elevado padrão de qualidade. Estas funções podem normalmente ser acedidas fazendo *include* dos ficheiros `.h` da biblioteca `openssl`, e.g., a inclusão da linha seguinte no cabeçalho de um ficheiro com código fonte C permite usar funções de cifra e decifra do *Data Encryption Standard* (DES):

```
#include <openssl/des.h>
```

Tarefa 1 *Task 1*

Crie um programa em linguagem C que permita cifrar e decifrar ficheiros em modo ECB com a cifra DES. Tanto o nome do ficheiro de entrada, como a chave de cifra e o nome do ficheiro de saída devem ser passados como argumentos ao programa, pela ordem pela qual foram indicados. O programa deve chamar-se `des-ecb` e deve cifrar o ficheiro colocado no terceiro argumento caso se invoque o programa com `$ des-ecb -e nome-in.txt key nome-out.txt`. Caso se use a opçao `-d` (em vez de `-e`), o programa deve proceder à decifra do ficheiro de entrada para o ficheiro de saída.

Em princípio, os seguintes recursos ser-lhe-ão úteis:

- Especificação da linha #include <openssl/des.h> no cabeçalho;
- Declaração das estruturas de dados DES_cblock e DES_key_schedule;
- Utilização das funções DES_set_odd_parity(&Key2);, DES_set_key_checked(&Key2, &schedule); e DES_ecb_ecb_encrypt(const_DES_cblock *input, DES_cblock *output, DES_key_schedule *ks, int enc);
- Utilização dos valores pré-definidos `DES_ENCRYPT` e `DES_DECRYPT`.

Q1.: Qual é o tamanho do bloco na DES?
☐ 8 bytes. ☐ 16 bytes. ☐ 32 bytes. ☐ 64 bytes.
Nota: esta informação é importante para a parte da definição dos *buffers* de leitura e escrita nos ficheiros de entrada e saída.

Q2.: Como se compila um programa escrito em linguagem C que use funções da *toolkit* do OpenSSL?
☐ `$ cc programa.c`
☐ `$ cc programa.c -libopenssl`
☐ `$ cc programa.c -ssl`
☐ `$ cc programa.c -lssl`
☐ `$ cc programa.c -lcrypto`

Tarefa 2 *Task 2*

Teste o programa que implementou antes, recriando a experiência sugerida na tarefa 9 do guia laboratorial anterior.

Q3.: Consegue cifrar um ficheiro com o OpenSSL e decifrá-lo com o seu programa?
☐ Mas é claro que sim. É canja de galinha.
☐ Parece-me que não...
☐ Desde que use a codificação da chave certa em ambas implementações...

2 Implementação em Java de um Programa para Cálculo do SHA1 de um Ficheiro

Implementation of a Program in Java for Calculating the SHA1 Hash Value of a File

Como o OpenSSL está para a C, assim o JCA está para a Java. O software fornecido nesta arquitetura, se disponível, pode ser normalmente acedida fazendo *import* das sub-classes do pacote `java.security.*`.

Tarefa 3 *Task 3*

Crie um programa em linguagem Java que permita calcular o SHA1 de qualquer ficheiro. Pode usar um *Integrated Development Environment* (IDE) para desenvolver o código, mas deve considerar compilá-lo no terminal, só para descargo de consciência. O programa deve correr no terminal também, e o ficheiro para o qual se calcula o valor resumo deve ser passado como primeiro argumento. Note que, no caso do Java, o nome do programa não conta como primeiro argumento (i.e., `argv[0]` é o primeiro argumento a seguir ao nome.). Deve ser possível chamar o programa com
`$ java sha1sum filename.txt`.
O nome do programa deve ser, portanto `sha1sum`, e o resultado do SHA1 deve ser expresso em hexadecimal.

Para esta implementação, os seguintes `imports` ser-lhe-ão potencialmente úteis:

```
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.io.FileReader;
import java.io.IOException;
import java.io.FileNotFoundException;
```

Por ser pedido o resultado do *hash* em hexadecimal, deixa-se uma sugestão para a implementação de um procedimento que converte uma sequência de bytes numa cadeia de caracteres que os representam em hexadecimal.

```
static final String HEXES ="0123456789abcdef";

public static String getHex(byte[] raw) {
  if (raw == null) {
    return null;
  }
  final StringBuilder hex = new StringBuilder
      (2 * raw.length);
  for (final byte b : raw) {
    hex.append(HEXES.charAt((b & 0xF0) >> 4)).
        append(HEXES.charAt((b & 0x0F)));
  }
  return hex.toString();
}
```

A função `main` não irá precisar de mais do que 6 linhas de código para fazer o que é preciso, entre elas estarão algumas semelhantes a:

```
MessageDigest oMD = MessageDigest.getInstance(
    "SHA1");
```

```
FileReader oIn = new FileReader(args[0]);
```

```
oMD.update((byte) buff[0]);
```

e

```
System.out.println("SHA1(file): " +
                   getHex(oMD.digest()));
```

Q4.: Como se compila um programa em Java no terminal?
☐ `$ cc sha1sum.java`
☐ `$ jcc sha1sum.java`
☐ `$ java sha1sum.class`
☐ `$ javac sha1sum.class`
☐ `$ javac sha1sum.java`

Tarefa 4 ***Task 4***

Teste o programa que implementou antes, calculando por exemplo o *hash* de um dado ficheiro usando o comando `$ openssl dgst -sha1 -hex file` e o programa que implementou, comparando os dois resultados no final.

Q5.: Os valores resumo têm de ser necessariamente iguais?
☐ Claro que sim.
☐ Não, neste caso não.

Q6.: O valor resumo obtido foi igual pelas duas vias?
☐ Sim, foi.
☐ Não, não foi, pelo que está na hora de verificar o que está mal.