| | |
|---|---|
| | Departamento de Informática<br>Faculdade de Ciências e Tecnologia<br>UNIVERSIDADE NOVA DE LISBOA |

Mestrado em Engenharia Informática

**Segurança em Sistemas Informáticos Distribuídos**

**2º Semestre 2005/2006**

Teste de Avaliação de Conhecimentos (Teste Nº 1 – 22/Abtril/2005) – 2ª Chamada

Notas:

- O enunciado tem 4 questões, divididas em duas partes:
  - **Parte sem consulta** (Questão 1): 50 min
  - **Parte com consulta** (Questões 2 e 3 ) : 50 min
- Leia completamente e com atenção cada questão antes de responder. A interpretação do enunciado é um factor de avaliação.

---------- A preencher pelos alunos ----------

Nº de aluno: _____ Nome: _____

Nº Total de páginas entregues: _____ (numere as páginas na forma Pág / TOTAL)

Classificação (a preencher pelo docente):

| **PARTE 1 (sem consulta)** | **PARTE 2 (com consulta)** |
|---|---|

| **1)** | **2)** | **3)** |
|---|---|---|
| a) | a) | a) |
| b) | b) | b) |
| c) | c) | |
| d) | | |
| e) | | |
| f) | | |
| g) | | |
| h) | | |
| i) | | |

**Questão 1)**

a) Segundo a terminologia OSI X.800 defina "mecanismo de segurança" e "serviço de segurança" ? Defina categorias de serviços de segurança e de mecanismos de segurança

b) Ainda de acordo com a terminologia X.800, qual a diferença entre as seguintes noções: confidencialidade orientada à conexão (*connection confidentiality*) e confidencialidade não orientada à conexão (*connectionless confidentiality*)

c) Quando se fala de um esquema de criptografia computacional quando é que se diz que esse esquema é incondicionalmente seguro ? Conhece algum esquema criptográfico que possa ser considerado incondicionalmente seguro ? Qual ?

d) Os algoritmos simétricos mais conhecidos não são "incondicionalmente seguros" mas sim "computacionalmente seguros". Que critérios estão na origem desta última classificação ?

e) Considere que se usa o modo CBC (Cipher Block Chaining) num algoritmo criptográfico simétrico que manipula blocos de tamanho B (ex: DES ou 3DES) e se cifra uma dada mensagem M de dimensão arbitrária L com L>B

Neste modo, cada bloco Ci de cifra (ciphertext) é obtido do bloco Pi em claro (plaintext) com a chave K do seguinte modo:

**Ci = E( K, [ Ci-1 XOR Pi ] )** ... ou **Ci = { Ci-1 XOR Pi }**k

Tente escrever a expressão de recuperação de Pi quando se decifra (utilizando uma ou outra das notaçõe inidcadas), dizendo nomeadamente como é a expressão para recuperar o primeiro bloco em claro P1 a partir do bloco cifrado C1.

**Pi = _____**
**P1= _____**

f) Quais os parâmetros de concepção essenciais estabelecidos pela estrutura de cifra de Feistel como base de partida para a construção de um algoritmo criptográfico simétrico como o DES? Enumere as características de segurança que estão subjacentes a cada um desses critérios

g) Em que consiste a garantia de segurança baseada no critério de avalanche no estudo de um método criptográfico simétrico ?

h) Suponha que lhe é fornecida uma função que implementa um gerador pseudorandom de sequências de 16 bytes. A partir deste gerador diga como poderia construir um programa que permitisse cifrar uma emissão de uma fonte que produz streams de bits em tempo real para um canal de comunicação série bit a bit (do tipo RS-232-C, por ex.,) e como se recuperaria a informação plaintext em tempo real por parte de um receptor.

**Questão 2)**

a) No programa anexo (2-A) utiliza-se CTR. Tendo em conta os dados cifrados (String Input) e a preservação das condições de segurança subjacentes ao programa acharia inconveniente usar ECB em vez de CTR ? Justifique.

Nota: a execução do programa resulta no seguinte:

```
input : 12345678
plain : 12345678 verificacao: true
```

b) Considere o programa anexo (2-B)

Note que o programa não usa qualquer modo de cifra simétrica (ECB, CBC, CTR, ... etc). Porque é que não é usado nenhum modo de cifra simétrica neste programa ? Justifique.

c) Tente interpretar o programa do anexo (2-C) que demonstra como se pode proteger uma chave privada RSA com uma chave AES usada como wrapping-key.
Qual o interesse que vê na utilização desta técnica, no caso específico de utilização de criptografia assimétrica RSA ? Tente avançar com um possível interesse de utilização desta técnica na sua actual implementação do trabalho prático nº 1.

**Questão 3)**

**Para responder a esta questão tem que considerar o contexto de realização do trabalho prático nº 1 e reflectir nas respostas a partir da sua implementação.**

a) Considera a sua especificação e implementação do protocolo da fase 2 do trabalho prático nº 1. Discuta se a sua solução está bem protegida contra ataques de modificação de mensagens ("*message tampering*"), retransmissão ilícita de mensagens ("*message repalying*") ou de impersonificação de utilizadores principais autenticados nas sessões ("masquerading "). Justifique a sua resposta com base no suporte e na concretização prática da sua implementação, argumentando se a troca de todas as mensagens estão protegidas contra aquele tipo de ataques. Considere todo o contexto de comunicação segura que se encontra concretizado conforme os requisitos do trabalho.

   Sugestão: apresente a sua implementação o mais detalhada possível em diagramas temporais (apresentando a especificação de todas as mensagens trocadas com base na notação usual para protocolos de segurança) e justifique concretamente como e onde se encontram as defesas contra esses ataques ao longo dos diagramas temporais. Use pelo menos um diagrama temporal para especificar o protocolo de autenticação e geração/distribuição de chaves criptográficas para as sessões e outro diagrama temporal para especificar o protocolo de transferência de ficheiros no âmbito das sessões.

b) Tenha em mente o contexto do suporte de comunicação subjacente aos requisitos do trabalho (autenticação e geração/distribuição de chaves, comunicação segura no âmbito de cada grupo de chat e o âmbito da transferência de ficheiros que podem ocorrer nas sessões de chat).

   B1) Que modo de cifra adoptou no âmbito da utilização de cifra simétrica ? Porquê ?

   B2) Diga se consideraria indiferente ou não usar outro ou outros modos em vez do modo adoptado. Justifique adequadamente a sua resposta tendo em conta a comunicação (todo o contexto do suporte de comunicação) que tem lugar no âmbito de uma sessão de chat.

## Programa para a Questão 2-A

```
import java.security.Key;
import java.security.MessageDigest;
import java.security.SecureRandom;

import javax.crypto.Cipher;
import javax.crypto.Mac;
import javax.crypto.spec.IvParameterSpec;
import javax.crypto.spec.SecretKeySpec;

/**
 * Mensagem protegida de tampering, com MAC (DES)
 */
public class CipherMacExample
{
    public static void main(
        String[]    args)
        throws Exception
    {
        SecureRandom random = new SecureRandom();
        IvParameterSpec ivSpec = Utils.createCtrIvForAES(1, random);
        Key key = Utils.createKeyForAES(256, random);
        Cipher cipher = Cipher.getInstance("AES/CTR/NoPadding","BC");
                        // ou ECB ? Ou CBC ? Ou Outro ?


        String input = "12345678"; // input tem 8 bytes
        Mac mac = Mac.getInstance("DES", "BC");
        byte[] macKeyBytes =
             new byte[] { 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08 };
        Key macKey = new SecretKeySpec(macKeyBytes, "DES");

        System.out.println("input : " + input);

        // cifrar

        cipher.init(Cipher.ENCRYPT_MODE, key, ivSpec);

        byte[] cipherText =
         new byte[cipher.getOutputSize(input.length() + mac.getMacLength())];

        int ctLength =
         cipher.update(Utils.toByteArray(input),0,input.length(),cipherText, 0);

        // Calculo de MAC com cifra simetrica

        mac.init(macKey);
        mac.update(Utils.toByteArray(input));
        ctLength +=
         cipher.doFinal(mac.doFinal(),0, mac.getMacLength(),cipherText, ctLength);

        // Decifra

        cipher.init(Cipher.DECRYPT_MODE, key, ivSpec);

        byte[] plainText = cipher.doFinal(cipherText, 0, ctLength);
        int    messageLength = plainText.length - mac.getMacLength();

        // Verificaao Mac

        mac.init(macKey);
        mac.update(plainText, 0, messageLength);
```

```
        byte[] messageHash = new byte[mac.getMacLength()];
        System.arraycopy(plainText,messageLength,messageHash,0,
           messageHash.length);

        System.out.println("plain : " + Utils.toString(plainText, messageLength)
             + " verificacao: " + MessageDigest.isEqual(mac.doFinal(),
             messageHash));
    }
}
```

## Programa para a Questão 2-B

```
import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec;

public class TesteExemplo
{
    public static void main(
        String[]    args)
        throws Exception
    {
        byte[]        input = new byte[] {
                0x00, 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77,
                (byte)0x88, (byte)0x99, (byte)0xaa, (byte)0xbb,
                (byte)0xcc, (byte)0xdd, (byte)0xee, (byte)0xff };
        byte[]        keyBytes = new byte[] {
                0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07,
                0x08, 0x09, 0x0a, 0x0b, 0x0c, 0x0d, 0x0e, 0x0f };

        SecretKeySpec key = new SecretKeySpec(keyBytes, "RC4");
        Cipher        cipher = Cipher.getInstance("RC4", "BC");
        System.out.println("input text : " + Utils.toHex(input));

       // Cifrar

        byte[] cipherText = new byte[input.length];

        cipher.init(Cipher.ENCRYPT_MODE, key);

        int ctLength = cipher.update(input, 0, input.length, cipherText, 0);

        ctLength += cipher.doFinal(cipherText, ctLength);

        System.out.println("cipher text: " + Utils.toHex(cipherText)
            + " bytes: " + ctLength);

        // Decifrar

        byte[] plainText = new byte[ctLength];

        cipher.init(Cipher.DECRYPT_MODE, key);

        int ptLength = cipher.update(cipherText, 0, ctLength, plainText, 0);

        ptLength += cipher.doFinal(plainText, ptLength);

        System.out.println("plain text : " + Utils.toHex(plainText) +
            " bytes: " + ptLength);
    }
}
```

## Programa para a Questão 2-C

```
import java.security.Key;
import java.security.KeyPair;
import java.security.KeyPairGenerator;
import java.security.SecureRandom;

import javax.crypto.Cipher;

public class AESWrapRSAExample
{
    public static void main(
        String[]    args)
             throws Exception
    {
        Cipher cipher = Cipher.getInstance("AES/ECB/PKCS7Padding", "BC");
        SecureRandom random = new SecureRandom();

        KeyPairGenerator fact = KeyPairGenerator.getInstance("RSA", "BC");
        fact.initialize(2048, new SecureRandom());

        KeyPair     keyPair = fact.generateKeyPair();
        Key         wrapKey = Utils3.createKeyForAES(256, random);

        // wraping da chave RSA
        cipher.init(Cipher.WRAP_MODE, wrapKey);

        byte[] wrappedKey = cipher.wrap(keyPair.getPrivate());

        // unwraping da chave RSA
        cipher.init(Cipher.UNWRAP_MODE, wrapKey);

        Key key = cipher.unwrap(wrappedKey, "RSA", Cipher.PRIVATE_KEY);

        if (keyPair.getPrivate().equals(key))
        {
            System.out.println("Chave recuperada com sucesso.");
        }
             else
             {
                 System.out.println("Erro na recuperação da chave.");
             }
    }
}
```