UNIVERSITÉ DE BATNA 2 MOSTEPHA BEN BOULAID

FACULTÉ DES MATHÉMATIQUES ET D'INFORMATIQUE
DÉPARTEMENT D'INFORMATIQUE

CHAPITRE 03

La cryptographie et Java

Dr. Ali BEDDIAF

PLAN

I. Introduction
II. Types de chiffrement
III. Cryptographie en Java
IV. JCE (Java Cryptography Extension)

INTRODUCTION

Définition

- La cryptographie est la transformation d'un texte en clair (plain text) en un texte chiffré incompréhensible (ciphered text)
- Contrairement à la stéganographie et au tatouage qui visent la dissimulation de données cachées, la cryptographie vise à protéger la confidentialité des données en cas d'interception
- Deux opérations sont utilisées: chiffrement et déchiffrement

INTRODUCTION

Exemple

- Chiffrement d'un texte alphabétique par décalage (+nb)
- Déchiffrement par décalage (-nb)

- Message original: bonjour
- Message chiffré (+1): cpokpvs
- Message déchiffré (-1): bonjour

TYPES DE CHIFFREMENT

- Il existe deux familles d'algorithmes de chiffrement :
  - Chiffrement symétrique : une seule clé est utilisée pour le chiffrement et le déchiffrement.
  - Chiffrement asymétrique : une clé publique est utilisée pour le chiffrement et l'autre pour le déchiffrement.

TYPES DE CHIFFREMENT

Algo. symétriques

- Comme exemple: DES, TripleDES, Blowfish, AES, IDEA, RC2, RC4, RC5, RC6

- Ces algorithmes se distinguent par
  - la taille de la clé utilisée (plus la clé est longue plus l'algo est plus sûr)
  - Chiffrement par bloc ou flux
- La vulnérabilité de ces algo. se situe au niveau de la partage de la clé

TYPES DE CHIFFREMENT

Algo. asymétriques

- Ils utilisent deux clés, une publique pour le chiffrement, et une privée connu que par le récepteur et utilisée pour le déchiffrement.
- Ces algo. sont plus sûr par rapport à la première classe.
- Comme exemple: RSA, DSA, Diffie-Hellman, ElGamal, ECC

CRYPTOGRAPHIE EN JAVA

- Java fournit deux bibliothèque: JCA et JCE
  - JCA: contient fonctionnalités cryptographiques de base (package java.security)
  - JCE: contient les fonctionnalités de chiffrement/déchiffrement avec les différents algo, génération de clés (package javax.crypto)

CRYPTOGRAPHIE EN JAVA

JCE(Java Cryptography Extension)

- javax.crypto contient des classes/interfaces, à savoir:
  - SecretKey: fonctions d'une clé secrète
  - Cipher: fonctions de chiffrement/déchiffrement
  - KeyGenerator: génération des clés pour les algo symétriques.
  - Mac: fonctions d'intégrité de type "Message Authentication Code"

CRYPTOGRAPHIE EN JAVA

JCE(Java Cryptography Extension)

- JCE doit être implémentée par des fournisseurs
  - Sun (ex-propriétaire de java): librairie sunjce.jar,sunec.jar …
  - Oracle (propriétaire actuel de java): librairie jce.jar
- Ces librairies se trouve dans JAVA_HOME/lib/ext

JCE (JAVA CRYPTOGRAPHY EXTENSION)

KeyGenerator

- Elle génère des clés pour des algo. symétriques, syntaxe:
  - static KeyGenerator getInstance(String algorithm)

- **algorithm** peut être : AES, Blowfish, DES, DESede, HmacMD5 et HmacSHA1

JCE (JAVA CRYPTOGRAPHY EXTENSION)

KeyGenerator

- L'initialisation se fait par:
  - public void init(int keysize, SecureRandom random)
- Avec deux paramètres:
  - la taille de la clé (keysize)
  - une source pour la génération de nombres aléatoires (SecureRandom)
- la génération de la clé se fait par:
  - public SecretKey generateKey();

```java
import java.security.SecureRandom;

import javax.crypto.KeyGenerator;
import javax.crypto.SecretKey;

public class TestKeyGeneratorDES {

  public static void main(String[] args) {

    KeyGenerator keyGen;
    try {
      keyGen = KeyGenerator.getInstance("DES");
      keyGen.init(56, new SecureRandom());
      SecretKey cle = keyGen.generateKey();
      System.out.println("cle (" + cle.getAlgorithm() + "," +
cle.getFormat()
          + ") : " + new String(cle.getEncoded()));
    } catch (Exception e) {
      e.printStackTrace();
    }
  }
}
```

JCE (JAVA CRYPTOGRAPHY EXTENSION)

Cipher

- Elle permet de chiffrer/déchiffrer une donnée
- Elle possède aussi les deux méthode getInstance et init()
  - Sauf que ici, le premier paramètre de init(…) précise le mode, soit chiffrement ou déchiffrement.
- L'actionnement du chiffrement/déchiffrement se fait par:
  - public byte[] doFinal(byte[] input);

```java
import java.security.*;
import javax.crypto.*;


public class TestCipherDESede {

  public static void main(String[] args) {

    final String message = "Mon message a traiter";

    KeyGenerator keyGen;
    try {
      keyGen = KeyGenerator.getInstance("DESede");
      keyGen.init(168);
      SecretKey cle = keyGen.generateKey();
      System.out.println("cle : " + new String(cle.getEncoded()));

      byte[] enc = encrypter(message, cle);
      System.out.println("texte encrypte : " + new String(enc));

      String dec = decrypter(enc, cle);
      System.out.println("texte decrypte : " + dec);

    } catch (Exception e) {
      e.printStackTrace();
    }
  }
```

Suite…

```java
 public static byte[] encrypter(final String message, SecretKey cle)
      throws NoSuchAlgorithmException, NoSuchPaddingException,
      InvalidKeyException, IllegalBlockSizeException,
BadPaddingException {
    Cipher cipher = Cipher.getInstance("DESede");
    cipher.init(Cipher.ENCRYPT_MODE, cle);
    byte[] donnees = message.getBytes();

    return cipher.doFinal(donnees);
  }

  public static String decrypter(final byte[] donnees, SecretKey cle)
      throws NoSuchAlgorithmException, NoSuchPaddingException,
      InvalidKeyException, IllegalBlockSizeException,
BadPaddingException {
    Cipher cipher = Cipher.getInstance("DESede");
    cipher.init(Cipher.DECRYPT_MODE, cle);

    return new String(cipher.doFinal(donnees));
  }

}
```