# Thư viện hỗ trợ mã hoá trong java

Mã hoá đối xứng:

java.security.

giới thiệu :

* Là một function hỗ trợ mã hoá messge với số bit khoá cố định.
* Là một one-way function, chỉ mã hoá không giải mã.
* Mã hoá sử dụng messagedigest có thể mạnh hoặc yếu tuỳ vào cáo giá trị đầu vào.

Các loại mã hoá hỗ trợ:

- MD2 và MD5 với 128bit.

* SHA1 với 160bit.
* SHA256, SHA383, SHA512 với số bit tương tự.

Function hỗ trợ :

.MessageDigest :

* MessageDigest.getInstance("MD5") : tạo message băm.
* .update(plaintext) : tính toán chiều dai message băm với plain text.
* .digest() : đọc mã băm.

.KeyGenerator :

* KeyGenerator.getInstance("HmacMD5") and .generateKey() : tạo kye
* Mac.getInstance("HmacMD5") : Creates a MAC object.
* .init(MD5key) : Intializes the MAC object.
* .update(plaintext) and .doFinal() : Calculates the MAC object with a plaintext string.

Mã hoá bất đối xứng:

.KeyGenerator :

* KeyGenerator.getInstance("DES") , .init(56) , and .generateKey() : Generates the key.
* Cipher.getInstance("DES/ECB/PKCS5Padding") : Creates the Cipher object (specifying the algorithm, mode, and padding).
* .init(Cipher.ENCRYPT\_MODE, key) : Initializes the Cipher object.
* .doFinal(plainText) : Calculates the ciphertext with a plaintext string.
* .init(Cipher.DECRYPT\_MODE, key) : Decrypts the ciphertext.

.doFinal(cipherText) : Computes the ciphertext.

2. Chương trình sử dụng public key

Bước 1:

KeyGenerator keyGen = KeyGenerator.*getInstance*("AES") : tạo keygenerator khai báo kiểu AES.

keyGen.init(128): khái báo khoá đài 128bit hoặc 256bit.

SecretKey key = keyGen.generateKey(): khởi tạo key.

cipher.init(Cipher.***ENCRYPT\_MODE***, key): mã hoá cipher text bằng key vừa tạo.