Java Encryption

COMP620 Information Privacy and Security

Programming Assignment

- This project should be done by teams of two students
- One student creates a program that writes encrypted data to a file
- The other student’s program must read the encrypted data and display it in plaintext

Encryption Program should

- Read a text password from the keyboard
- Read a plaintext file
- Create a message authentication code (MAC) from the plaintext
- Encrypt the plaintext file
- Write both the encrypted data and MAC to an output file

Decryption Program should

- Read a text password from the keyboard
- Read the encrypted file
- Decrypt the file
- Create a message authentication code (MAC) from the plaintext
- Verify the MAC created against the MAC received
- Display the decrypted data

Design Questions

- Encryption algorithm to use
- How to convert the text password to an encryption password
- Format of the output file which contains the text and MAC

Deadline

- Upload the programs to Blackboard under the userid on one student on the team
- Comments in the programs should credit both team members
- Due before midnight on Thursday, August 30, 2018

Java Encryption Classes

- The standard Java library provides several classes and methods for cryptography
- The **`javax.crypto`** package provides the tools to do encryption and “Message Authentication Code” (MAC)

Creating Cipher Keys

- The class javax.crypto.spec.SecretKeySpec can be used to create a key

- SecretKeySpec secretKey =
  new SecretKeySpec(key, "AES");
- where key is an array of 16 bytes

- Other encryption algorithms are supported

Encrypt and Decrypt Class

- The class javax.crypto.Cipher can be used to encrypt or decrypt bytes
- Cipher cipher = Cipher.getInstance(*algorithm name*);
- where algorithm name is a string indicating which encryption algorithm is to be used
- “AES/ECB/PKCS5Padding” will work

- cipher.init(Cipher.ENCRYPT_MODE, secretKey);

Encrypting

cipherObj.update(byte[] data)

- will encrypt the data

byte[] ciphertext = cipherObj.doFinal(byte[] data)

- will complete the encryption and returns the ciphertext

Message Authentication Code (MAC)

- Your program needs to calculate a MAC on the plaintext and include it in the file
- The class java.security.MessageDigest can create a MAC
- java.security.MessageDigest.getInstance("SHA-1") will return a MessageDigest object

- mdObj.update(bytes) will add data to the hash

Completing the Hash

- mdObj.digest(bytes) will return a byte array with the hash

Bytes, not Strings

- The encryption methods use arrays of bytes, not Strings
- There are methods to convert between Strings and byte arrays
- You have to decide how to write the data to the file
- There are multiple encoding methods
  - Base64.getDecoder().decode(stringInput)
  - Base64.getEncoder().encodeToString(bytes)