Readme

1. Question:

What are the limitations of using self-signed certificates? What are they useful for?

Answer:

(1). Self-signed certificates generate a rather deterring warning, when the user tries to install your application. Web Start still allows installation but a user might think twice about it, after reading the warning.

(2). When test servers, creating a website needed to test over an https connection, you don't have to pay for a signed certificate for that testing site. You just need to tell the testers that their browser may pop warning messages.

1. Software

Eclipse SDK Version: 4.2.1.

1. Message Digest

This program allows user to enter a string in the terminal, uses MD5 and SHA scheme to hash the input string to generate the output. The Java project includes three files: MessageDigest.java, MD5.java, SHA.java . Here are the steps on how to execute this application.

(1). Execute the ‘MessageDigest.java’ file.

(2). Type the number as shown in the terminal, enter the string (my name) for both MD5 and SHA, the results are shown on the Figure. 1 .

1. Authentication

This program allows user to implement the double-strength password login using message digest. The Java project includes three files:

Protection.java: includes makeBytes, makeDigest(Version 1), and makeDigest(Version 2).

ProtectedClient.java: includes sendAuthentication and main functions.

ProtectedServer.java: includes lookupPassword, authenticate and main functions.

Here are the steps on how to execute this application.

(1). First, execute “ProtectedServer.java”.

(2). Then execute “ProtectedClient.java”.

(3). The final result will show on the screen as the Figure 2.

1. Signature

The java project includes two files: one is “ElGamalAlice.java”, which includes main and four BigInteger functions to execute the ElGamal signature scheme, and the other is “ElGamalBob.java”, which includes main and verifySignature functions.

(1). First, execute “ElGamalBob.java”.

(2). Second, execute “ElGamalAlice.java”.

(3). The final result will show if the signature verified or not.

1. Encryption

There are two files:

CipherClient.java

CipherServer.java

Step 1: Execute “CipherServer.java”.

Step 2: Execute “CipherClient.java”.

Step 3: The result of the decryption will show on the CipherServer’s screen.

1. Public-Key System

There are two files:

pksAlice.java and pksBob.java .

Step 1: Execute “RSAPublicKeyAlice.java”.

Step 2: Execute “RSAPublicKeyBob.java”.

Step 3: Follow the instruction to choose the type of methods of message exchanging and enter a string, and then the result will show on the screen.

1. There are four files:

X509Client.java

X509Server.java

chaoshi.cer

chsoshi.keystore

Step 1: Execute “x509Server.java”.

Step 2: Execute “x509Client.java”.

Step 3: At “x509Client” side, you can the content of server’s certificate and whether it is valid. Then send x509Server a message by using its public key to encipher the content. At “x509Server” side, the server uses its private key to decipher the received ciphertext.