Assignment 3

Add encryption and decryption steps that use the following cryptographic algorithms, to the client program that you have written as part of your Assignment 1 work.

1. DES (Key Length: 56 Bits)
2. 3-DES (Key Length: 112 Bits)
3. AES (Key Length: 128 Bits)
4. DH (Key Length: 1024 Bits)
5. RSA with CRT (Key Length: 1024 Bits)

※ Chinese Remainder Theorem (CRT) is an alternative method of representing the private key in RSA. The CRT method of decryption is four times faster overall than the non-CRT representation method.

※ You may use the OpenSSL library for the Socket/RPC version or JCE (Java Cryptography Extension(Java 1.5)) or JCA (Java Cryptography Architecture (Java 6)) for the RMI version. Reference materials/sites are shown on the next pages.

Execute your client-server programs that encrypt-and-write/read-and-decrypt (encrypted/decrypted) records to/from a server, in the same round-robin manner starting with the first cryptographic algorithm. Do experiments in the following order:

1. Encrypt-And-Write with One Client Process Requesting Sequential Access
2. Read-And-Decrypt with One Client Process Requesting Sequential Access

※ Modify the provided **`dbgen`** and **`tracegen`** programs to generate variable-sized database-record and client-access-trace files – note that the size of encrypted data is larger than that of the data prior to encryption.

Measure the mean encryption/decryption time for each cryptographic algorithm, and the mean (remaining) response time in each of the above two cases – this response time should be comparable to what you have measured in the

corresponding case of Assignment 1.

Compare the mean encryption/decryption times for all cryptographic algorithms. For this comparison, draw a bar chart in each (Write/Read) case (with a bar for each cryptographic algorithm) - use "Cryptographic Algorithm" as the x-axis title and "Response Time" as the y-axis title. Each time bar is composed of two parts with the lower one for the mean encryption/decryption time, and the upper one for the mean response time. Also, provide a table that shows the mean times and the standard deviations in each case.

Compare your encryption/decryption times with your team members,' and discuss the difference if any.

About submission

- Tar/zip the files that contain the source code (with each author's name on the corresponding part) and a team report
- Name the resulting file as "**`DIP16_HW3_TeamNo.[tar|zip]`**"
- Submit this file to the TA (Jeesoo Min: kuongee@gmail.com / kuongee@hanmail.net) by 11:59pm on December 20th

(If there's a problem with submitting the HW to TA's "gmail", you can try the mail with "hanmail".)

- ✓ Code
  - ➢ Indent and comment the source code
  - ➢ Make the code compilable and runnable
- ✓ Report – make brief, clear statements
  - ➢ Explain the experiment environments
  - ➢ Mention parameter values used in the experiments
  - ➢ Discuss comparison results by using charts and tables
  - ➢ Explain the results by using the corresponding pseudo code
  - ➢ Conclude the report

※ Writing in Korean is ok, but writing in English is preferred.

<References>

- ✓ OpenSSL

- OpenSSL Project Homepage: http://www.openssl.org
- "Network Security with OpenSSL" by John Viega et al., O'Reilly
  - Chapter 6 Symmetric Cryptography
  - Chapter 8 Public Key Algorithm
- Man Pages in Linux, Section 3 - evp, dh, rsa
- Miscellaneous Links
  - Encryption Using OpenSSL's Crypto Libraries by Vinayak Hegde - Symmetric Crypto: http://tldp.org/LDP/LG/issue87/vinayak.html
  - QADPZ Documentation by the QADPZ Team at the Norwegian University of Science and Technology - RSA: http://qadpz.idi.ntnu.no/doxy/html/RSAcrypter_8cpp-source.html

✓ JCE

- Java™ Cryptography Extension (JCE) Reference Guide (Java 1.5) : http://java.sun.com/j2se/1.5.0/docs/guide/security/jce/JCERefGuide.html
- Java™ Cryptography Architecture (JCA) Reference Guide (Java 6): http://download.oracle.com/javase/6/docs/technotes/guides/security/crypto/CryptoSpec.html

✓ Algorithm Description

- RSA with CRT
  - DI Management – CRT: http://www.di-mgt.com.au/rsa_alg.html#crt
- Google – DES, 3-DES, AES

<Challenge Problem>

Describe a possible security hole/holes in your programs, and how to address/fix it/them.