CS-4331-Special Topics in Security

Fall 2016

LAB #3: Asymmetric Encryption

This lab is meant to give you hands-on experience with some of the concepts of asymmetric ciphers and how asymmetric encryption relates to symmetric encryption. As before, you will use OpenSSL in your VMs (You may use the Kali VM you used for first lab).

1. In this task, you will compare the performance of asymetric key algorithms to that of symmetric key encryption algorithms. You will create files with different sizes and encrypt them and gauge the performance.

   For example, to create a 20byte file of zeros named “myfile.cs4331”, you can use the command:

   ```
   dd if=/dev/zero of=myfile.cs4331 bs=20 count=1
   ```

   This creates 20 blocks of 1 byte each.

   You can also use a value of bs as K, M or G to represent Kilo, Mega and Giga – units.

   Create a file of 16 bytes and another of 20KB

   Create a 16 byte AES key and IV.

   Also create a 2048-bit RSA key-pair.

   You may use the command: openssl genpkey –algorithm RSA –pkeyopt rsa_keygen_bits: 1024 –pkeyopt rsa_keygen_pubexp:3 –out keypair.pem

   This creates a private key; however, the corresponding public key can easily be derived from it.

   You can extract the public key using: openssl pkey -in keypair.pem -out pubkey.pem -pubout

   Try to encrypt the 20KB file using 1024-bit RSA public key.

   ```
   openssl rsautl -encrypt -pubin -inkey pubkey.pem -in myfile.cs4331 -out EncryptedFile.txt
   ```

   Explain the reason behind your observations.

Now encrypt the 16 byte file using your public key.

Decrypt the result using your private key. Time this operation and repeat it multiple times so you can average the time taken. You can use the time command to time the operation (You will get several outputs, but you report the *real* value, which is the time between the start and end of your call).

Example decryption:

```
openssl rsautl -decrypt -inkey keypair.pem -in EncryptedFile.txt -out DecryptedFile.txt
```

Now encrypt the 16 byte file using AES in cbc mode. Decrypt the resulting cipher text and again report an average decryption time after multiple attempts.

Are you able to see any difference between AES and RSA decryption?

To more clearly see the difference, you will use the standard benchmarks.

```
openssl speed rsa1024
openssl speed aes-128-cbc
```

Describe/interpret your observations.s

2. In this task you will create and verify a digital signature.

   Create a text file of about 20 bytes. Use the private key created in the previous step to digitally sign the file (use SHA512 for the hash).

   Verify the signature of your output file (you will use your public key).

   Edit the file and again attempt to verify the signature.

   Explain your observation (s) and provide screen shots to show all your commands.