Homework 2

Track A

# **LAB 0 –** Determine

|  |  |  |
| --- | --- | --- |
| Rewrite as Extended Euclidean | | |
|  |  | *extended euclidean* |
|  |  | *substitution* |
|  |  |  |
| Use Division Algorithm to Solve Euclidean Equations | | |
|  |  |  |
|  |  | *solve and to find* |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  | *Repeat until , using as and as* |
|  |  |
|  |  |
|  |  |  |
|  |  |  |
|  |  | *Repeat until , using as and as* |
|  |  |
|  |  |
|  |  |  |
|  |  |  |
|  |  | *Repeat until , using as and as* |
|  |  |
|  |  |
|  |  |  |
|  |  |  |
|  |  | *Repeat until , using as and as* |
|  |  |
|  |  |
|  |  |  |
|  |  |  |
|  |  | *Repeat until , using as and as* |
|  |  |
|  |  |
|  |  |  |
|  |  |  |
|  |  | *When ,* |

# **LAB 5 –** (OpenSSL) Generate Random Number and Test Primality

## (10pts) Based on the script primeTest, write another shell script using OpenSSL command line. The script is called isprime. It requires an integer as an input parameter. If the input integer is a prime, the script will show “\*\*\* is a prime”. If not, the script will show “\*\*\* is not a prime”.

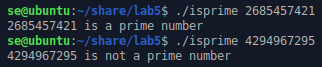
|  |
| --- |
| #!/bin/bash  input**=$1**  # Use OpenSSL to check if prime  result**=$(openssl prime $input)**  # Verify OpenSSL Ouput and display message  **if** **[[** "$result" **==** **\***is\ prime **]];** **then**  **echo** "$input is a prime number"    **elif** **[[** "$result" **==** **\***not\ prime **]];** **then**  **echo** "$input is not a prime number"  **fi** |

A screenshot of a computer program

Description automatically generated

## (5pts) Use the program to decide if the following numbers are prime numbers:

* 1. 2685457421 – Yes, this is a prime number
  2. 4294967295 – No, this is not a prime number



## (5pts) If an integer passes the Miller-Rabin primality test, does that guarantee the number is a prime number? Why?

No, an integer passing the Miller-Rabin primality test does not guarantee that it is prime. Since the Miller-Rabin primality test is a probabilistic primality test, it can only indicate that the integer is a probable prime. In other words, since the test relies on randomness in its computations, there are scenarios where a composite number might pass the test as if it were prime.

# **LAB 6 –** (OpenSSL) Create RSA Public/Private Key (512bits) without Password Protection

## (8pts) In Step 1, what is the used? Why?

The exponent, , is shown during RSA private key generation in Step 1. This number is popular for RSA key generation because it is a prime number, thus making it relatively prime to the numbers and , where and are the prime factors of the modulus . Also, it is small enough to not slow down the encryption or decryption processes, while still being large enough that when combined with a large modulus (i.e. 2048-bit or 4096-bit), it creates a high level of complexity. This complexity makes it practically impossible for someone to break the encryption by trying to factorize the modulus and obtain the private key.

Alternatively, is not shown during the RSA private key generation in Step 1 as it is randomized during each generation. Unlike the exponent, it is created by multiplying two randomly selected large prime numbers, and . This process ensures that each key pair is unique and not predictable. However, you can use the following OpenSSL command on the private key to display the value of :

$ **openssl** rsa -in private512.key -modulus -noout

or the following OpenSSL command on the public key:

$ **openssl** rsa -pubin -in public512.key -text -noout

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## (12pts) A certificate, Amazon.cer, can be found in the lab 6 folder, answer the following questions:

* 1. What is the signature hash algorithm used to create the certificate?

1.2.840.113549.1.1.11 (sha256WithRSAEncryption)

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

* 1. What is the public key cryptography used to create the signature?

RSA

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* 1. What is the size of the public key?

2048

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* 1. What is the used in the public key in the certificate?

A screenshot of a computer

Description automatically generated

* 1. Who issued the certificate?

Starfield Services Root Certificate Authority - G2 of Starfield Technologies, Inc.

A screenshot of a computer

Description automatically generated

A computer screen shot of white text

Description automatically generated

* 1. Why can you trust the certificate?

This certificate is trustworthy because it has been issued by a trusted root certificate authority (Starfield Services Root Certificate Authority - G2), it has been signed using RSA encryption, and it is still within its validity timeframe (May 25, 2015 - December 31, 2037)

# **LAB 7 –** (OpenSSL) Create RSA Public/Private Key (4096bits) with Password Protection

## (8pts) Use symmetric cipher and message authentication code to provide confidentiality, integrity, and authentication. (use the giving cryptographic operations only).

A diagram of a rectangular object with black text

Description automatically generated

## (8pts) Use symmetric cipher and digital signature to provide confidentiality, integrity, and authentication. (use the giving cryptographic operations only).

A diagram of a data flow

Description automatically generated

## (4pts) Compare the two scenarios as designed in a) and b). What are their advantages and limitations?

Scenario A, the symmetric cipher and MAC, offers efficient data encryption, ensuring confidentiality while maintaining data integrity and authentication. The addition of a MAC provides an extra layer of security on top of the cipher as it verifies data integrity and source authentication. However, securely managing and distributing the shared secret keys for the symmetric cipher can be challenging. Scenario B, the symmetric cipher with a digital signature, improves security by providing a means to authenticate the sender's identity and prevent data tampering. However, these are susceptible to various attacks, including replay attacks, man-in-the-middle attacks, and key compromise attacks.

# **LAB 8 –** (OpenSSL) Combination of RSA and AES to Encrypt a File

## (6pts) What is the throughput when you sign and verify a message using a 1024-bit key in RSA? Which one is faster? Why?

RSA 1024-bit Sign throughput:

RSA 1024-bit Verify throughput:

The verify throughput is faster than the sign throughput because verifying a signature involves simpler computations with a small public exponent (e.g., 3, 5, 17, 257, or 65537), thus making it quick and inexpensive, whereas generating signatures requires more computationally intensive operations with a private exponent ().

A computer screen shot of a computer screen

Description automatically generated

## (6pts) What is the throughput when you encrypt a 1024-bit message block using DES CBC mode? Compared with a 1024-bit RSA verify throughput, which one is faster? Compared with a 1024-bit RSA sign throughput, which one is faster?

DES CBC 1024-bit throughput: **\***

RSA 1024-bit Sign throughput:

RSA 1024-bit Verify throughput:

Comparatively, DES CBC encryption is slower than RSA verification but faster than RSA signing for 1024-bit operations.

***\**** *While the throughput for 1024 bit is not explicitly stated in the speed test, it can be estimated by finding the average between 64 bytes and 256 bytes:*

A screenshot of a computer

Description automatically generated

## (8pts) What is the plaintext you restored from the cipher text?

After decrypting the cipher text with the password obtained from RSA decryption (hello world), the restored plaintext is “Belief creates the actual fact. William James”

A computer screen shot of a computer code

Description automatically generated