Computer Security Assignment 2

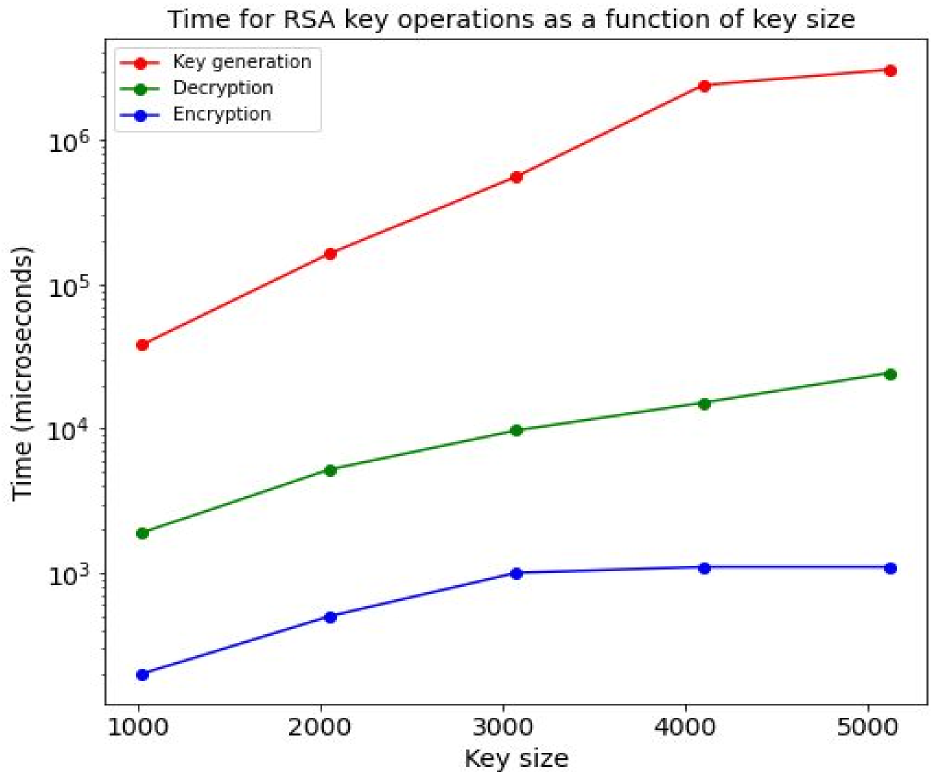
Problem: Our task was to analyze the time requirements for various RSA key operations such as key generation, encryption and decryption as a function of key size.

Solution Plan: We decided to use JAVA for the task, the plan is to use the inbuilt libraries in JAVA for key generation, encryption and decryption. For each of the operations we compute the time taken by it and finally plot graphs to compare the time requirements of key generation, encryption and decryption.

Work Done:
Link to our code:
https://github.com/Bharat-Reddy/RSA-Time-Analysis/blob/master/RSA-TimeAnalysis/src/timeAnalysis.java

- We decided to use the Java KeyPairGenerator class present inside the java.security package to generate both public and private keys for RSA.
- For encrypting and decrypting data we used the Cipher class inside the javax.crypto package.

For our analysis we performed all the 3 operations(key generation, encryption and decryption) with 5 different key sizes(1024, 2048, 3072, 4096, 5120 bits). For each key size we perform all the 3 operations for 10 times and finally calculate time taken for each operation to be the average over all runs. After repeating the process for each key size, we then plot the graph showing how the time taken varies for each of these operations as a function of key size



Learnings:

We have learned the relation between time requirements for RSA key operations - key generation, encryption and decryption. We found out that time taken for key generation is way higher than decryption and encryption operations, because it takes a lot of time to generate two large primes. Though decryption and encryption operations uses modular exponentiation which is polynomial in time, the time taken for decryption is higher than encryption, since the exponent part in decryption operation (private key ‘d’) is very large compared to exponent part in encryption operation (public key ‘e’).
i.e.
Time complexity of encryption is proportional to $(\log(n))^2(\log(e))$ and
Time complexity of decryption is proportional to $(\log(n))^2(\log(d))$
Since d >> e, time taken for decryption is greater than time taken for encryption.

References:

1. Key Generation Tutorial: How to Generate RSA Keys in Java
2. Encryption and Decryption: Java Asymmetric Encryption Decryption Example with RSA
3. Cipher Class reference: Cipher (Java Platform SE 7)
4. RSA time complexity reference: Why RSA Decryption process takes longer time than the Encryption process?

Done By:

Team 4

J. Vinod Kumar Reddy -- B160769CS
Shah Kenil Ramesbhai -- B160632CS
T. Bharat Bhushan Reddy -- B160198CS