Analysis of Time taken for RSA algorithm

V.G.Vinod Vydiswaran
02329011

IT653 Network Security Assignment 1

August 16, 2003

1 Key Size vs. Time

1.1 Observations

1. The time required for encryption and decryption is much **less** than that for key generation, when key sizes are smaller (< 512). As key sizes increase, the encryption and decryption times increase.

2. This experiment was done with encryption being done block-by-block. It was observed that without block-by-block encryption, the time taken for encryption and decryption was much higher. It can be concluded that for very large message sizes, the encryption and decryption time is much higher.

3. The time taken for key generation increases exponentially to the key size.

4. The time taken for encryption and decryption are almost equal and they take time exponential to key size.

5. Time $= O(e^{|key|})$

6. The exponential time taken for encryption and decryption is primarily because of the large size of the key, and hence more computation. As key size increases, the exponent to be taken for encryption and decryption also increases, so computation increases exponentially.

7. It is estimated that for constant encryption key, the time taken for encryption will grow linearly with message size. This is corroborated in the second experiment as explained in section 2.

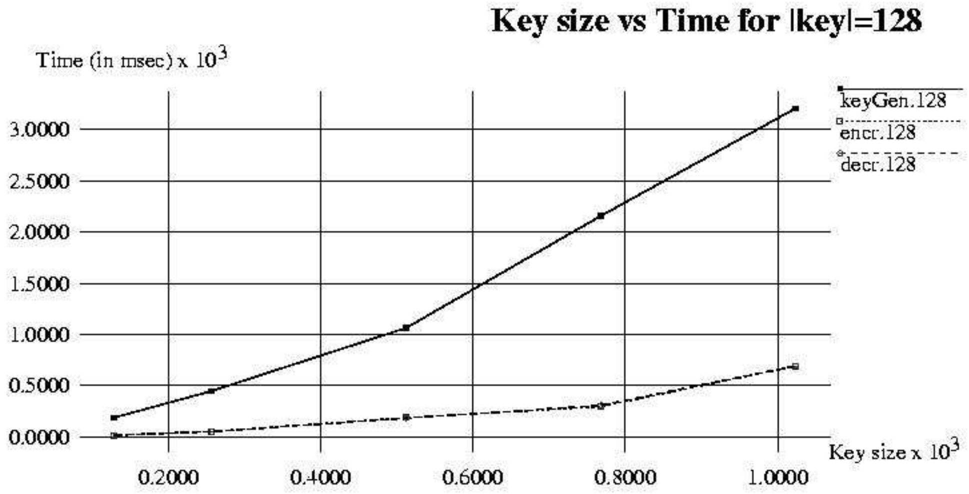

Figure 1: Key Size vs. Time for message size of 128 bytes

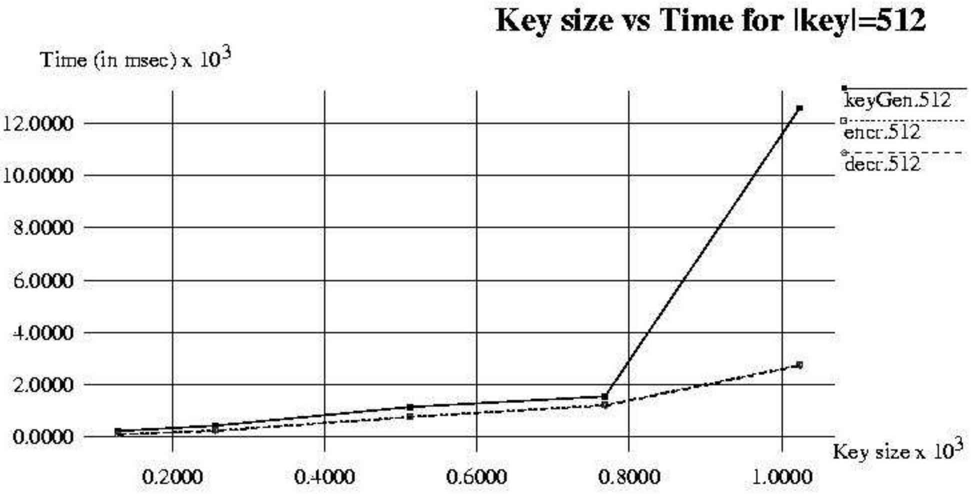

Figure 2: Key Size vs. Time for message size of 512 bytes

2 Message Size vs. Time

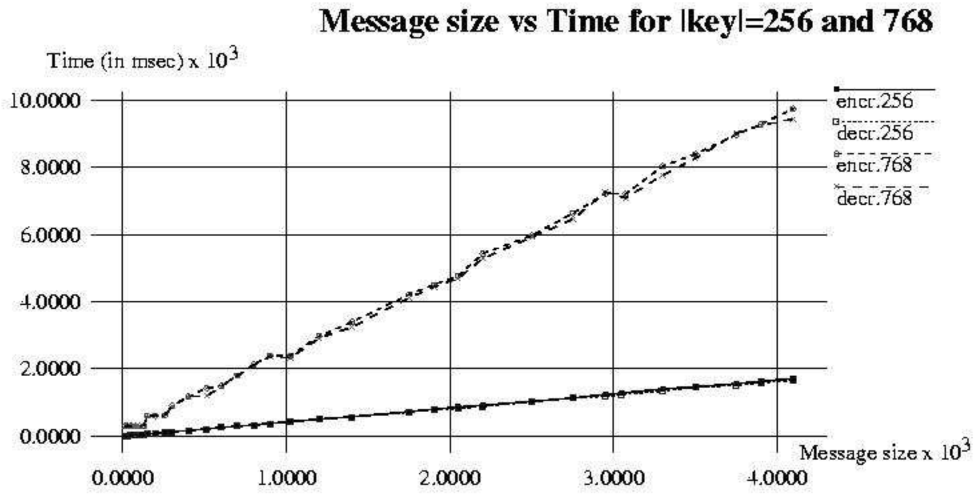

Figure 3: Message Size vs. Time for key sizes of 256 and 768

2.1 Observations

1. The time required for encryption and decryption are nearly equal.
2. The order of the time taken for encryption and decryption varies linear to the message size.
3. Time $= O(|msg|)$
4. As key size increases, the time taken also increases.
5. It is observed that due to block-by-block encryption, the time taken is not exactly a linear function, but a step ("staircase") function. This is because, for a range of message sizes, the key size remains constant, and hence encryption and decryption times are also constants.

3 Primality confidence parameter (n) vs. Time

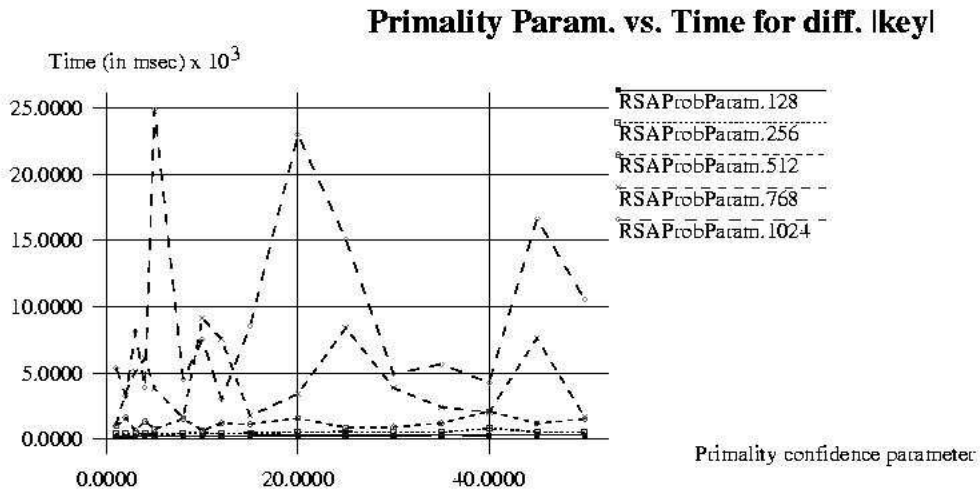

Figure 4: Primality Confidence parameter vs. Time for various key sizes

3.1 Observations

1. The time required for generating key increases as primality parameter increases.
2. The increase in time is more prominant for larger key sizes.
3. The parameter n is converted to confidence value as $1 - \frac{1}{2^n}$. This means the time function is almost constant function for a fixed key size.
4. Time $= O(|key|)$ and a constant wrt n
5. There are a lot of "hills" and "valleys" in each plot for a fixed key size. The reason is not known.

4 Appendix A: Code

Please refer the following java programs in the code/ folder:

1. **RSAmod.java**: This java program is modified RSA program incorporating block-by-block encryption.

2. **RSAprob.java**: this java program generates only the keys and measures the time.

5 Appendix B: Sample output

Please refer **sample.out** for a sample output of prime numbers generated, the cipher text generated and the clear text retrieved, in addition to the time taken for key generation, encryption and decryption.

6 Appendix C: Script program

Please refer the following two scripts in the code/ folder:

1. **runRSAkey.pl**: To obtain key generation, encryption, and decryption time for various message sizes and key sizes.

2. **runRSAprob.pl**: To obtain key generation time for keys with different primality confidence parameter.