ADDIS ABABA UNIVERSITY

School of Electrical and Computer Engineering

Parallel Computing Project Proposal

submitted to Fitsum Assamnew

Eliyas Girma and Hailemelekot Demtse

Contents

1 Abstract

Public key algorithms are extensively known to be slower than symmetric key alternatives in the area of cryptographic algorithms for the reason of their basis in modular arithmetic. The most public key algorithm widely used is the RSA. Therefore, how to enhance the speed of RSA algorithm has been a significant research topic in the computer security as well as in computing fields. Therefore, the main purpose of this project is to device a parallel computing approach that speed up the execution time of RSA algorithm and alleviate the slowness of the RSA algorithms.

2 Introduction

Now a days huge amount of data is exchanged through Internet network which is pone to hackers and malicious users with different intentions of manipulating other persons data. These malicious users have a different aims for hacking others data which include compromising the integrity of informations, accessing other's data for illegal use and and other related actions which negatively affect the original owner of the data. Compromising data which is available in a networked environment by malicious users has become prevalent in our global system and become among a serious issues which affect the day to day activities of individuals, companies and countries in general. The problems observed in the last American presidential election can be cited as one of best example to show how the hacking related issues are reached too severe level and become a serious threat for our world's peace and and integrity.

In order prevent this issue different professionals come up with different solutions and among these solutions the one is encryption of data. Encryption is one of the methods of enciphering the data which involves hiding the message using a key into some function and deciphering the enciphered message using a key. There is symmetric encryption scheme, which uses single key shared between the participants, on the other hand, public key uses two keys the one for encrypting and the other decrypting.

Over the past two decades, with the rapid evolution in the area of information technology and the internet have created imaginative applications and technologies along the way. Recently, we can send a multimedia message, or get one from almost anyone around the world in a few seconds through the internet. The main aim of encryption is to guard data transmitted from one party not be accessed or compromised by other party who is not allowed to access that particular data —any person or entity other than the receiver. It is desired to hide the message before it is sent to a non-secure communication channel. This is achieved through encryption. Due to its distinctive ability to distribute and manage keys, public key encryption has become the perfect solution for information security . Public key algorithms (e.g., RSA algorithm) rely essentially on hard mathematical problems (modular multiplication and modular exponentiation) of very large integers, ranging from 128 to 2048 bits. With such large numbers, the achievement of the calculation process will not be quick or easy to implement . Seeing the recent rapid developments in hardware and software technologies, it seems that sequential implementation of encryption algorithms are not recommended due to their slow nature. Parallel algorithms on the other hand play a significant role in maintaining rapid growth. Such trend becomes prevalent is not only observed in multi-core processors but also it becomes commonplace in powerful graphics which are included in many general and special purpose computers recently. Graphics Processing Units (GPUs) have been increasingly used as a powerful accelerator in several high computationally demanding applications due to their flexibility and moderate cost. The essential difference between CPUs and GPUs comes from how transistors are composed in the processor. CPUs use large portions of the chip area for caches; while GPUs use most of the area for Arithmetic Logic Units (ALUs). These behavior enables the GPUs to engage in massively parallel activities.

2.1 RSA ALGORITHM

RSA algorithm—invented by Rivest, Shamir and Adelman in 1978—is one of the famous algorithms for public-key cryptography. It is appropriate for encryption and digital signature.

RSA is the utmost far used algorithm in Internet security . In fact, Internet security depends significantly on the security properties of the RSA crypto system. Its security depends upon the insolvability of the integer factorization problem and is believed to be vulnerable given sufficiently long keys, such as 1024 bits or more . The RSA algorithm consists of three steps which include key generation, encryption and decryption ones. It is comprised of public and private keys. Messages encrypted with the public key can only be decrypted using the private ones. The RSA algorithm can be summarized in the following steps :

1. Generate randomly two large prime's p and q of approximately the same size, but not too close together. Which are kept secret.

2. Calculate the modulus n = p*q, and Calculate: $\phi(n) = (p-1)(q-1)$; Where $\phi(n)$ represents the Euler Totient function.

3. Choose a random encryption exponent e less than n such that the GCD $(\phi(n), e) = 1, 1 < e < \phi(n)$.

4. Calculate the decryption exponent d using The Extended Euclidean algorithm:$d.e = 1 mod \phi(n)$. Which d is the multiplicative inverse of e modulo $\phi(n)$.

5. The encryption function is:$E(M) = M^e mod n$.

6. The decryption function is: $D(C) = C^d mod n$.

7. The RSA keys are: The public key is (n, e), and the private key is (p, q, d).

3 Probelem Statement

As we have discussed in the previous sections, most of the public key encryption algorithms manifest computationally slow characteristics due to computationally intensive nature. The main objective of the problem is implementing RSA encryption and decryption algorithms using parallel computing approach in order to alleviate the slowness and improve the performance and efficiency RSA algorithms.

In addition, the project is aimed to compare and contrast the performance of different parallel computing environments through execution of the parallel algorithms in different parallel computing platforms.

4 Methodlogy

The RSA encryption and decryption algorithm is going to be implemented using different platforms, mainly using sequential and parallel computing platforms. The development technologies which are selected to be used for this project are C/C++ and Java programming languages and OpenMP and OpenCL APIs. As the aim of this project is not only implementing RSA encryption and decryption algorithms on parallel and sequential development environments, after implementing the algorithms using aforementioned development environments different analysis works will be done in order to create the platform to compare and contrast not only different implementation mechanismsserial and parallelbut also different platforms which are categorized within the same development mechanism like OpenCl and OpenMP.

5 How the project will be implemented?

Before delving into thinking the implementation of the project using, it is better to think about how RSA algorithm is working and figure out the part of the algorithm that shares the larger amount of execution time and the one that do not share much percentage of the execution time. So, in the case of RSA the largest share of execution time is taken by the encryption and decryption tasks while public and private key generation tasks do not share much amount of execution time. So, after identifying the aforementioned section the next thing that to figure out the part of the code that is going to implemented in parallel and the part of the code that is going to be implemented in sequential manner. So, as it is described above in this project the generation of private and public keys are going to be implemented

in sequential manner while the encryption and decryption algorithms are expected to be implemented in parallel manner.

5.1 Proposed Algorithm to be Implemented

As it is known there are different ways of implementing RSA algorithm in parallel computing environments however for this project Repeated Square-and-Multiply Method is going to be used for our implementation. The pseudo code for repeated-square-and multiply algorithm for RSA encryption/decryption is given below:

```
Procedure: square_repeat
Model: Thread Model based on repeated square-and-multiply
Input: base, Power, modulus
Output: Result
Declare:
Global: N, Number_of_Thread, Cipher
Local: None
Parbegin
Declare Result:=1
Assign N:=Power/Number_of_Thread
for Pi, i =1 to Num_proc do In Parallel
if Power mod Number_of_Thread := 0
for i:= 1 to Power
for j := 1 to N
Assign Result := Result * Base
end for
end for
else
Assign N := N - 1
```

```
for i:= 1 to Power
for j := 1 to N
Assign Result := Result * Base
end for
end for
Assign Result := Result * Base;
end if
end In Parallel
Assign Cipher := Result mod Modulus
Parend
```

For thread based platforms like C-Pthread and Java-Threads, the following algorithms will be used to utilize the performance gain of multi-core systems.

```
Procedure: square_repeat
Model: Thread Model based on repeated square-and-multiply
Input: base, Power, modulus
Output: Result
Declare:
Global: N, Number_of_Cores, Cipher
Local: None
Assign N:=Power/Number_of_Cores
Divide the whole modular exponentiation and reduction into N parts
Parbegin
Declare Result:=1
If Power mod 2 := 0
for j := 1 to N
Assign Result := Result * Base
end for
else
for j := 1 to N
Assign Result := Result * Base
end for
Assign Result := Result * Base;
end if
Assign Cipher := Result mod Modulus
Parend
```

References

[1] HM Fadhil, and MI Younis, Parallelizing RSA Algorithm on Multicore CPU and GPU,International Journal of Computer Applications (2014).

[2] Sapna Saxena, Parallel Algorithms for Public Key Infrastructure Based Security Techniques, December 2014, Chitkara University, India