Joy of GPU Computing: A Performance Comparison of AES and RSA in GPU and CPU

R. Kingsy Grace, M. S. Geetha Devasena, and S. Manju

Abstract The impact of social media evolution has led to the development of big data where the two Vs, volume of data handled and velocity of data retrieval, have grown exponentially. The volume of data being handled has reached the level of terabyte's and even exabyte's. On other hand, the velocity of data retrieval has also matured through the fourth- and fifth-generation Web. To cope up this fastest-growing generation, the processing units also have been improved. The challenge lies in the secured transmission of data at a high speed irrespective of its size. Encryption and decryption are a time-consuming process in a general-purpose computer. The capability of graphical processing units (GPUs) has been proven for general-purpose computing in many research areas. With the computation capability of GPU, fast encryption and decryption could be achieved. In this paper, GPU-based symmetric and public encryption algorithms are proposed to render high performance. The performance analysis is performed based on the execution time of two security algorithms RSA and AES in normal computing platform, central processing unit (CPU) and GPU platform. The parallel versions of the RSA and AES are written in CUDA C. Results show that the performance of the encryption algorithm has greatly increased by using GPU computing.

Keywords Graphical processing unit · GPU computing · CUDA · RSA · AES

R. Kingsy Grace (✉) · M. S. Geetha Devasena
Department of Computer Science and Engineering, Sri Ramakrishna Engineering College, Coimbatore, India
e-mail: kingsygrace.r@srec.ac.in

M. S. Geetha Devasena
e-mail: msgeetha@srec.ac.in

S. Manju
Department of Computer Science & Engineering, CMR Institute of Technology, Bangalore, India
e-mail: manju.s@cmrit.ac.in

T. Sengodan et al. (eds.), *Advances in Electrical and Computer Technologies*,
Lecture Notes in Electrical Engineering 672,
https://doi.org/10.1007/978-981-15-5558-9_39

1 Introduction

CPU, the heart of the computer, is the major processing unit for applications involving normal computations. The GPU is yet another processing unit which is having many efficient and small cores for executing two or more tasks in parallel. Due to highly parallel structure of modern GPUs, they are more efficient than CPUs. GPUs find place in wide variety of applications such as personal computers, workstations, embedded systems, mobile phones and game consoles.

1.1 GPU Versus CPU

The latest GPUs execute the image processing and computer graphics efficiently. Parallelization of visual data in large amount is achieved using GPUs rather than CPUs because of GPUs are more active in general-purpose parallelization. Usually in desktop systems, the GPU is attached with video card or motherboard or CPUs as external peripheral devices. Originally, GPUs are developed to render real-time effects in computer games. For scientific applications, GPUs provide unprecedented computational power [1].

1.2 GPU Computing

GPU accelerated computing [1] is ubiquitous where GPU is used along with CPU to accelerate applications in various domains including analytics, scientific and enterprise applications. Usually, the jobs run faster in a GPU–CPU environment when compared to a CPU environment. They are also used to power up energy-efficient data centers.

1.3 CUDA Architecture

NVIDIA developed CUDA [2] framework for parallel computing. The CUDA API provides the application developers to use GPU for general-purpose processing. CUDA acts as an intermediate layer between the GPU and the application. The CUDA framework uses the programming languages such as C, C++ and FORTRAN and the programming frameworks such as OpenACC and OpenCL.

1.4 Using GPU in Cryptography

There is an increasing demand for high-speed secure Internet connection due to the advancement of cloud and Internet technology. The speed of the processor cannot catch up with these advancements. The implementation of the entire secure Internet connection demands random number generation that maintains the security of the connection. Therefore, researchers conducted study on fast implementation of cryptography algorithms using GPU [3]. In the current scenario, it is necessary to improve the security of the online data transaction. In the day-to-day use of Internet, millions of users apply block cipher for encryption. The block cipher algorithm performance is not bringing much effect to the user because the input data block that is received from the Internet is small and acceptable. But the input data block received by the server is larger and block cipher algorithm slows down the performance of the server. This algorithm consumes huge amount of computer resources and thus delays response client. There is a requirement for random data generation to keep the entire block cipher algorithm secure. The random number generation is a time-consuming task.

The problem of low processing power of CPU to encrypt and decrypt the data with respect to the speed in which data is sent and received via Internet results in slow processing and is rectified using GPU computing [4]. The higher processing capacity of GPU can be used to encrypt and decrypt data instead of using CPU before sending through the network which increases the speed in which the data is encrypted and decrypted. The remaining part of the paper is planned as follows. Section 2 throws light on the GPU trends. Section 3 deals with the RSA and DES algorithms to be analyzed using CPU and GPU. Section 4 discusses with the performance analysis of algorithms in both platform, and the conclusion is presented in Sect. 5.

2 Literature Survey

This section deals with the GPU trends in the computing world where high performance is been rendered. Some of the encryption algorithms using GPU are discussed in this section.

AA Abdelrahman et al. have proposed an efficient AES implementation using CUDA which reduces the execution time when the data size is large. AES algorithm 128 bit block size is implemented in all the GPU architectures. The proposed optimization algorithm in [1] achieves higher encryption speed than CPU. Nowadays, GPUs are used for different general-purpose research areas. Jo Heeseung et al. have proposed an improved data encryption for ODBS. The proposed system in [2] provides not only faster, the overall performance is best for database system. The AES is implemented using CUDA framework and MySQL is integrated. The performance of the GPU is eight times better than CPU for 16 MB data size.

Tuteja and Vaibhav have discussed RSA algorithm for image encryption and decryption [4]. The proposed RSA algorithm with other image processing algorithms

is used for improving encryption and decryption efficiency. The proposed method uses edge detection technique to identify points in the digital image. The public and private keys are generated by RSA algorithm. Each block is having a single image and is processed in different block combinations. For processing, the data is transferred to GPU. Instead of RSA, any other encryption technique is used to obtain the parallelism which increases the speed and efficiency.

Affan Hasan et al. have proposed GPU-based implementation of ECB encryption for 128 bit AES. The AES algorithm [5, 6] is widely used in most of the electronic communication systems. The proposed algorithm in [5] is tested in two GPUs, namely NVIDIA Quadro FX 7000 and Tesla K20c and CPU Intel Xeon X5690. Four approaches have been tested for input data size from 1 MB to 512 MB. The GPU performance is degraded when the input data is randomized and without using cache memory. The execution time is also increased when the randomization is increased in the input data and the repetition is less.

Patchappen et al. have presented the implementation of multi-variant AES cipher using GPU which is based on batch execution [7]. The throughput of GPU-based implementation is higher than CPU only implementation. The National Institute of Standards and Technology (NIST) standardized AES algorithm 2001. The AES has fixed block size with 128 bits. The key size is varied as 128 bits, 192 bits and 256 bits [4, 8]. DES is designed using Feistal cipher. AES is based on substitution and permutation. Experimental results showed that multi-variant AES cipher using GPU outperformed 2.5 times the single core CPU-based implementation for 512 MB data size. When compared with multi core CPU, AES cipher using GPU outperformed 1.6 times for 512 MB data size. Wai-Kong Lee et al. have implemented SSL/TLS-based secure solution for preventing communications from malicious attack. The SSL/TLS needs more computations in the server side. So the GPU architecture is used to implement the cryptographic algorithm. Pascal architecture of the GPU is used to implement SSL/TLS using SHA-3 and proved to be the best one using CUDA [8].

Sonam Mahajan and Maninder Singh have analyzed the RSA algorithm [9]. The authors have implemented both the traditional RSA algorithm and the parallelized RSA algorithm using Compute Unified Device Architecture (CUDA) [8] framework. The CPU-based implementation is compared with the GPU-based implementations for both small and large prime numbers. Lukasz Swierczewski has proposed an optimized 3DES ECB using CUDA framework [10]. The proposed algorithm is tested on three processors: (a) Intel Core 2 Quad Q8200, (b) Intel Core i7 950, (c) Intel Xeon E7—4860 and two graphics cards: (a) nVidia GeForce GTS 250, (b) nVidia Tesla C2050. The GeForce GTS 250 is faster than Xeon E7-4860 (ten cores, twenty threads). The performance of Tesla C2050 is 1.84 times faster than the performance of GeForce GTS 250. Both bit operations and integer operations GPU provide better results. RSA is a public key encryption algorithm [11].

Tejal Mahajan and Shraddha Masih have parallelized Blowfish algorithm using GPU [12] for improving the encryption and decryption speedup. Parallelization allows larger files to transmit through the network efficiently and securely. Experiments are conducted using Intel(R) Core(TM) 2 Duo CPU E7500 @ 2.93 GHz and

NVIDIA GeForce GT 610 (CUDA Cores 48). Bruce Schneier methodology [13] was adopted for CPU-based encryption and decryption. GPU implementation of Blowfish algorithm provides better results when compared to its CPU implementation. Jianwei Ma et al. have proposed different parameters needed for improving the performance of the CBC–AES algorithm implementation in GPU. The proposed implementation in [14] is compared with AES and AES–NI, and the latter is proved to be best in GPU implementation. The proposed AES–NI algorithm is 112 times better than AES algorithm.

3 Implementation of Security Algorithms

This section deals with the implementation of security algorithms AES and DES. Both the algorithms were implemented in two ways. The first one is implementation in CPU, and the second one is CPU–GPU implementation.

3.1 Difference in Implementation of CPU and CPU-GPU

The implementation workflow of security algorithms in CPU and in CPU–GPU is shown in Fig. 1 a and b. First, the key for the whole process is calculated during the initialization process which will be utilized in the whole process. Second, the input data is read inside the input buffer as batches into a small array, which is given as input for the encryption/decryption process. Once processing is done, the output is written to an output file. Here the difference lies in the number of parallel threads that are created in CPU–GPU platform in case of GPU processing. In GPU once input is obtained, memory is allocated in GPU for processing. New threads are created for every batch of inputs and are processed simultaneously. After processing, the data is sent back to the host and written on to the output file. Parallel processing is achieved in CPU–GPU implementation.

3.2 Implementation of RSA and AES

The popular security algorithms RSA and AES are implemented using C. The size of input data is varied from 100 to 500 MB. The input file is encrypted, and the time taken to generate cipher is recorded for CPU platform. Similarly, the same varying size cipher text was given as input to the decryption algorithm, and the time taken for decryption is also recorded. All the functions and sub-functions of both the algorithms were processed serially.

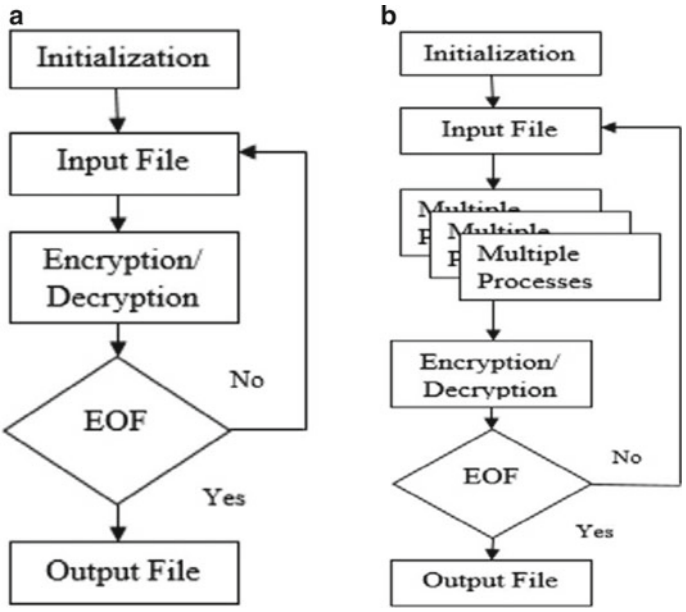

Fig. 1 **a** Implementation of security algorithm in CPU–GPU **b** Implementation of security algorithm in CPU

3.3 Implementation of RSA and AES Algorithms in GPU Using CUDA C

The steps for RSA and AES algorithms are same for CPU execution and CPU–GPU execution except the processing method and CUDA C programming. The execution is done parallel in GPU. The data block size is 64 byte which is transferred to GPU for encryption and decryption. The threads are executed in parallel where the number of threads is based on the number of blocks. The expansion of key is done in CPU and is serial. The data is transferred from GPU to CPU after the execution. Constant memory is used to store the look up table instead of global memory to improve the execution speed of GPU-based algorithm. The CUDA framework is used for the execution of AES on GPU. Every thread is divided into many parts. To reduce the execution time of encryption and decryption process, each part is executed in parallel. The work flow of security algorithms in CUDA is shown in Fig. 2.

4 CPU and GPU Performance Comparison

The RSA and AES algorithms executed in both CPU and CPU–GPU platform are compared for their performance. The time taken to encrypt and decrypt the data is

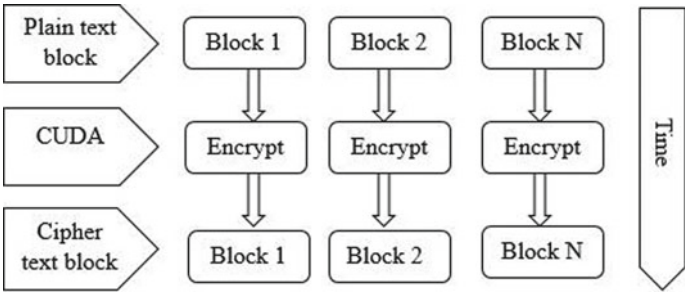

Fig. 2 Work flow of algorithms in CUDA

taken as the performance parameter for comparison. The execution time is compared for both the platforms for different size of the input files.

4.1 Performance Analysis of RSA Algorithm

The RSA algorithm is executed in CPU using C language with the varying size of input file such as 100 MB, 200 MB, 300 MB, 400 MB and 500 MB. The time taken to perform the encryption and decryption of these files using RSA algorithm in CPU is noted and is shown in Table 1. The CPU–GPU execution time is also shown in Table 1. Based on these values, the graph representation is shown in Fig. 3 a and b. The configurations used for comparison of results are Intel Core i7 4720HQ CPU with NVIDIA GTX 960 M GPU (640 cores) and Intel Core i5 4210U CPU with GeForce 830 M GPU (256 cores).

Table 1 Execution time of RSA

Time taken/input size (MB)	Core i7 and GTX 960 M				Core i5 and GeForce 830 M			
	CPU		GPU		CPU		GPU	
	Encrypt	Decrypt	Encrypt	Decrypt	Encrypt	Decrypt	Encrypt	Decrypt
100	3	8	5	6	10	17	9	10
200	9	21	7	8	14	30	12	13
300	11	26	8	10	19	42	13	15
400	13	32	9	12	25	49	14	17
500	16	40	11	12	25	49	16	20

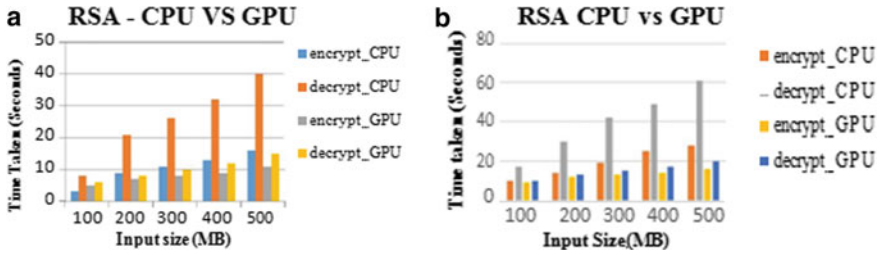

Fig. 3 **a** Execution time of RSA in core i7 and GTX 960 M **b** Execution time of RSA in core i5 and GeForce 830 M

Table 2 Execution time of AES

Time Taken/input size (MB)	Core i7 and GTX 960 M				Core i5 and GeForce 830 M			
	CPU		GPU		CPU		GPU	
	Encrypt	Decrypt	Encrypt	Decrypt	Encrypt	Decrypt	Encrypt	Decrypt
100	34	35	4	3	44	45	7	6
200	72	71	7	5	85	87	9	7
300	81	86	9	7	107	108	11	9
400	101	106	12	9	135	140	15	12
500	120	125	15	11	182	186	20	15

4.2 Performance Analysis of AES Algorithm

The performance of AES algorithm is analyzed with sample input files with sizes of 100 MB, 200 MB, 300 MB, 400 MB and 500 MB. The execution time of AES algorithm in CPU is observed and shown in Table 2. The execution time of AES algorithm using GPU is observed and shown in Table 2. Figure 3 a and b shows the performance analysis on AES in Core i7 and Core i5 with and GTX 960 M. The performance comparison is done with two varying configurations such as Intel Core i7 4720HQ CPU with NVIDIA GTX 960 M GPU (640 cores) and Intel Core i5 4210U CPU with GeForce 830 M GPU (256 cores).

The execution time of RSA and AES in CPU-GPU shows that there is a drastic increase in the performance of the algorithms especially when a high end graphic card is being used. This shows the increase in performance with GPU (Figs. 4 and 5).

5 Conclusion

In recent days, GPU computing plays a major role in all the domains of computer science and engineering. The encryption and decryption algorithms take more time

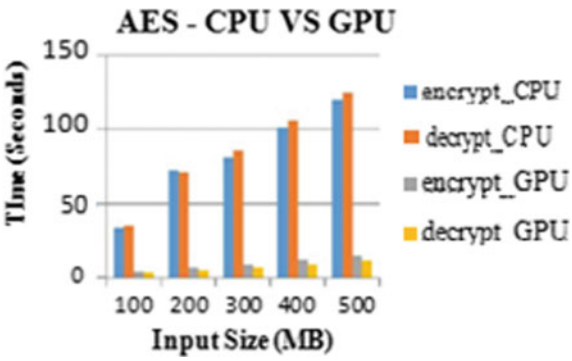

Fig. 4 Execution time of AES in core i7 and GTX 960 M

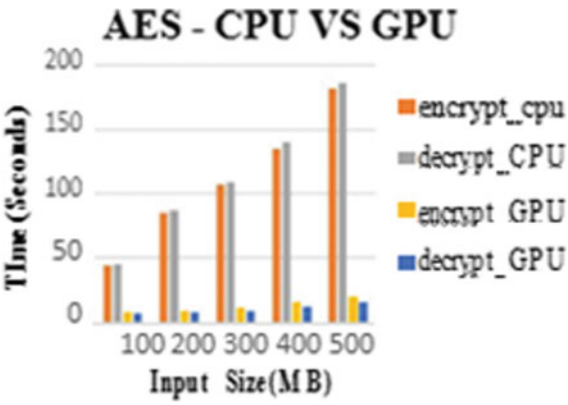

Fig. 5 Execution time of AES in core i5 and GeForce 830 M

when the size of the file to be encrypted is large. The proposed GPU-based RSA and AES are implemented both in traditional programming language and CUDA framework. The proposed work demonstrates the encryption and decryption algorithms based on GPU shows eight times better performance compared to that on CPU. Execution time of both CPU and GPU is compared in varying file size, and the GPU provides less execution time than its GPU counterpart.

References

1. Abdelrahman AA, Fouad MM, Dahshan H, Mousa AM (2017) High performance CUDA AES implementation: a quantitative performance analysis approach. IEEE Comput Conf https://doi.org/10.1109/sai.2017.8252225
2. Jo H, Hong S-T, Chang J-W, Choi DH (2013) Data encryption on GPU for high-performance database systems. In: Procedia computer science. vol 19, pp. 147–154 https://doi.org/10.1016/j.procs.2013.06.024

3. Renan CD, Lobato RS, Spolon R, Cavenaghi MA (2011) Using GPU to exploit parallelism on cryptography, 6th iberia n conference on information systems and technologies (CISTI 2011), pp. 1–6
4. Tuteja V (2014) Image encryption using parallel RSA algorithm on CUDA. Int J Comput Networks Commun Secur 2:232–235
5. Khan AH et al.: AES-128 ECB encryption on GPUs and effects of input plaintext patterns on performance, Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), (2014) 15th IEEE/ACIS International Conference on. IEEE https://doi.org/10.1109/snpd.2014.6888707
6. Yuan Y et al (2014) Acceleration of AES encryption with openCL, IEEE ninth Asia joint conference on information security (ASIA JCIS). https://doi.org/10.1109/asiajcis.2014.19
7. Patchappen M, Yassin YM, Karuppiah EK (2015) Batch processing of multi-variant AES cipher with GPU, IEEE second international conference on computing technology and information management (ICCTIM)
8. Lee W-K, Wong X-F, Goi B-M, Phan RC-W (2017) CUDA-SSL: SSL/TLS accelerated by GPU, 2017 IEEE international carnahan conference on security technology (ICCST). https://doi.org/10.1109/ccst.2017.8167848
9. Mahajan S, Singh M (2014) Analysis of RSA algorithm using GPU programming. Int J Network Secur Appl 6(2014) https://doi.org/10.5121/ijnsa.2014.6402
10. Swierczewski L (2013) 3DES ECB optimized for massively parallel CUDA GPU architecture, arXiv e-prints, arXiv:1305.4376
11. Rivest RL, Shamir A, Adleman L A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21(1978):120–126
12. Mahajan T, Masih S Enhancing blow fish file encryption algorithm through parallel computing on GPU. IEEE Int Conf Comput Commun Control (IC4–2015)
13. Schneier B (2007) Applied cryptography: protocols, algorithms, and source code in C, Wiley
14. Ma J, Chen X, Xu R, Shi J (2017) Implementation and evaluation of different parallel designs of AES using CUDA, IEEE second international conference on data science in cyberspace. pp. 606–614 https://doi.org/10.1109/dsc.2017.19