International Journal of Innovative Research in Computer Science & Technology (IJIRCST)
ISSN: 2347-5552, Volume-10, Issue-4, July 2022
https://doi.org/10.55524/ijircst.2022.10.4.16
Article ID IRPV1048, Pages 142-148
www.ijircst.org

Comparative Analysis of Energy Cost of Sequential and Parallel Cryptographic Algorithms on Different Platforms

Dr. Disha Handa
Assistant Professor, Department of Computer Science and Engineering,University Institute of Computing, Chandigarh University, Mohali, Punjab, India
Correspondence should be addressed to Dr. Disha Handa; dishah@gmail.com

ABSTRACT- Cell phones, smart cards, and health monitoring gadgets are just a few examples of the numerous battery-powered embedded systems utilized to access, alter, and store sensitive and complicated data today. Users are concerned about the protection of their identity credentials, their software packages, and their information. These systems make considerable use of cryptographic algorithms to implement security measures. Many cryptographic algorithms do calculations that are hard to compute and waste a huge amount of energy as a result. In this study, the energy consumption of serial and parallel cryptography algorithms is analyzed. Using an eight-core parallel system and Joule metre (Microsoft's Research Tool), we were able to reduce energy consumption in comparison to sequential algorithms with promising results. The study says that low-frequency symmetric multiprocessors have shown promising results and can make a big difference in green computing, which would be good for society as a whole.

KEYWORDS- Energy Cost, Symmetric Algorithms, Green Computing, Energy

I. INTRODUCTION

The importance of energy costs has grown in the computing industry, as they directly influence the operating expenses of enterprise infrastructures, the cost of power provisioning for computing infrastructures, as well as the power consumption of laptops and other mobile devices. It is well-known that cryptographic algorithms perform enormous and complicated calculations to secure sensitive data from unauthorized access. Due to the rigorous computing inherent to cryptographic techniques, they typically use considerable amount of power. According to [1], encrypting 13.6 kb of data on a mobile device using the Blowfish block cipher algorithm will take approximately 75% of the battery power. Numerous researchers have attempted to contribute towards this topic of crucial importance. Many power models, for example power gating and active body-bias, and many more have been discussed in the literature [2] to solve power consumption difficulties. The authors of the article titled "Computational and Energy Costs of Cryptographic Algorithms on Handheld Devices" conducted a thorough analysis of the costs associated with initiating symmetric/asymmetric key infrastructure based algorithms, and compared them to the costs associated with basic operating system functions. Results indicate that although cryptographic energy costs are significant, such processes should be time-constrained [3]. Experiments were done on a configuration with one K20M GPU and two Xeon E5-2640 v2 CPUs reduced power usage by 74% vs CPU-only parallel AES algorithm and by 21% versus GPU-only parallel AES algorithm on the same platform [4].

This work provides a thorough comparison and analysis of the energy usage of serial and parallel algorithms using a specific experimental setup. This is the first research of its kind to compare sequential versus parallel cryptographic algorithms in terms of their energy consumption on identical machine configurations and platforms. The remainder of the paper is structured as follows: Section 1 discusses the study's motivation, while Section 2 provides an overview of the sequential and parallel algorithms employed in the study. In the next part, the instruments and methods used in the research, as well as the approach employed, are described in depth. Section 4 contains the experimental results and discussion, followed by the conclusion.

II. MOTIVATION OF RESEARCH

There is an essential relation between power and frequency [5] the transition to multi-core CPUs is a response to the rising power consumption of microprocessors. In a multi-core system, all cores can operate at a lower frequency, thereby dividing the power that would normally be allocated to a single core and reducing the system's overall power consumption. It is due to the fact that decreasing frequency also reduces the operating voltage, and power consumption is quadratically dependent on the supply voltage [6]. Symmetric multiprocessor platforms have been proposed as a means of increasing computing cycles while conserving energy. Since the relationship between frequency and power of a system core is nonlinear, the energy consumption of a uni-processor can be reduced by decreasing its operating frequency. However, decreasing the frequency of a single processor will diminish the algorithm's performance. A parallel algorithm consists of a few sequential sub-computations, concurrent computations, and synchronization between the concurrent sub-computations. Consequently, the performance in terms of speed and energy cost of the parallel algorithm is focused on two primary factors: the number of available physical cores as well as the operating frequency of each core, and the design of the parallel algorithm. Research and analysis are conducted in this work on how to reduce the power consumption of

compute-intensive processes or applications by decreasing the frequency. However, this modification will slow down the application.

A. Cryptographic Algorithms Used for the Study

Broadly two distinct categories of cryptographic algorithms exist. Techniques based on an asymmetric infrastructure and techniques based on a symmetric infrastructure. Asymmetric cryptographic techniques employ two distinct keys for encryption and decryption, whereas symmetric algorithms, keep a single key for both encryption and decryption. In addition, Symmetric infrastructure-based approaches fall into two categories: block ciphers and stream ciphers. Stream Ciphers are a prevalent form of an encryption algorithm. Using an encryption transformation, they encrypt each character of a plaintext message individually. A block cipher encrypts data in predetermined-size blocks. Triple DES, DES, AES, and Blowfish are the most frequently used block cipher algorithms. This investigation included two block ciphers: Blowfish and PBlock, where Blowfish is the sequential cryptographic algorithm and PBlock is the parallel version of Blowfish. The study included a few stream cipher algorithms: RC4, RC4A, PARC4, and PARC4-I, where RC4 and RC4A are the sequential algorithms and PARC-I and PARC4 are the parallel algorithms.

Bruce Schneier created Blowfish [8] as a symmetric encryption technique in 1993. It has a block size of 64 bits and a key length that varies from 32 bits to 448 bits. By performing several encryptions during key scheduling, enormous pseudo-random lookup tables are generated. All required tables are dependent on the user-supplied complicated key. Multiple attacks, including differential and linear cryptanalysis, have been demonstrated to be ineffective against this method. However, this also implies that the approach cannot be utilized on computers with limited memory space. Since then, Blowfish has received substantial attention as a robust encryption technique. It is not patented and does not require a license.

Ron Rivest created the RC4 stream cipher in 1987 [9]. The maximum key length of the encryption is 2048 bits (256 bytes). The algorithm is quite quick. it is used in many applications because of its speed of encrypting/decrypted data in the form of bits/bytes. It further consists of two sub-algorithms: one for key creation and the other for encryption. For encryption, the generator's output is XORed with the data stream.

One of the most formidable alternatives to the RC4 algorithm is RC4A. Bert and Preneel made the proposal [10]. It has a modified key stream generator that provides a higher level of security than RC4. Most attacks against RC4 are less successful against RC4A. In addition, RC4A takes fewer instructions per output byte and it is possible to exploit the inherent parallelism to improve performance.

The PARC4 method is a parallel stream cipher built on the PASCS framework [11]. RC4 is comprised of two sub-algorithms: KSA for key stream generation and PRGA for encryption and decryption. In addition, KSA [12] executes a fixed number of iterations, but the number of PRGA algorithm calls varies on the length of the input data. PASCS is utilized to parallelize the PRGA algorithm. However, PRGA is built on the fundamentally sequential exchange shuffle model. The input to the PASCS framework must be delivered in the form of fixed-length individual blocks. First, the input data has been separated into blocks of predetermined size. The output of each block is then concatenated to form the cipher-text. Multiple cores do all of these activities in parallel to achieve speed advantages.

Another parallel algorithm, PARC4-I, is built on the PASCS framework [13]. This algorithm divides the text input into 256-byte blocks of a specified size. Multiple data blocks are then simultaneously encrypted using PRGA. As described in Section 5.2, each index pointer increments PRGA to generate four separate bytes; hence, the first four bytes of plaintext can be retrieved collectively for encryption or decryption. Finally, the output of each block is concatenated to form the whole encrypted text using the loop unrolling approach. This method cuts down on the costs of function calls because PRGA is only processed 16 times for every 64 bytes of data instead of 32 times in RC4A or 64 times in RC4.

The PBlock execution paradigm is built on the data parallel model, and this model maps readily to the PIFNS framework [14]. "Tasks are statically assigned to processes, and each task does the same operations on different data," says the description of the method.

III. METHODOLOGY

This research aims to determine the energy and time costs associated with sequential and parallel cryptography methods. We give a simple but useful set of case studies that use sequential and parallel algorithms on two different platforms, namely battery-powered devices and desktop computers, to show how much it costs to use encryption techniques in different situations.
Metrics and Methods of Measurement

- Throughput is a parameter for encryption algorithms that measures the speed of conversion.
- Energy cost is an additional essential metric that reveals how much energy an encryption algorithm consumes when executing encryption and decryption operations.

Microsoft's "Joule metre" simulator is utilized to measure and compare the energy cost and throughput. The entire computer system and the primary hardware components are represented by power data in the Joule meter [15]. Using this software package, data values of power consumption for a particular application may also be monitored. If desired, the values can also be saved to a file. Fig.1 depicts the interface of the simulator.

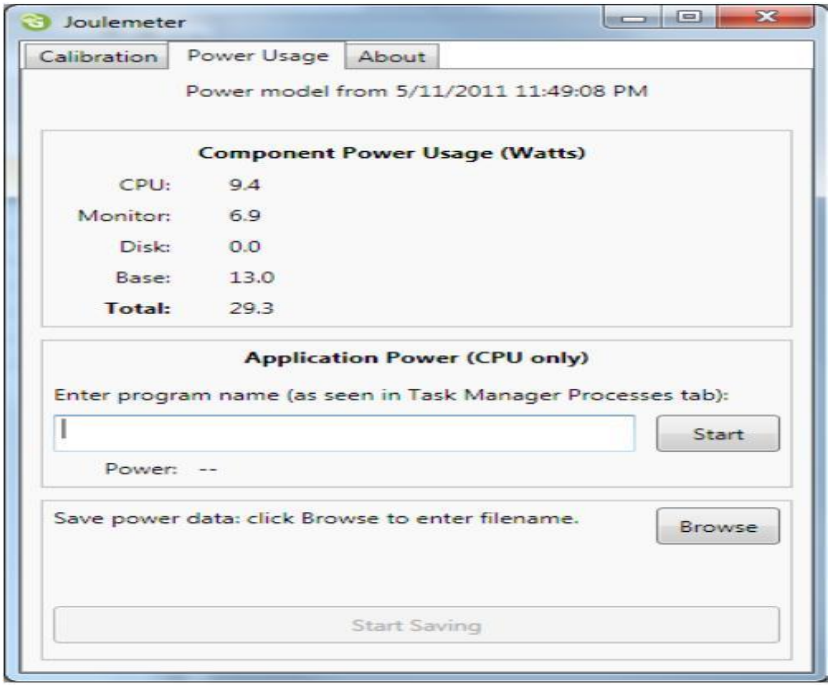

Figure 1: Power metering interface exposed by Joule meter

A Joule metre evaluates power consumption using a power model that monitors computer system's application resource usage and hardware power status [16]. This status is generated by doing a calibration process. On laptops, calibration is possible without the use of an external power metre. A Watts UP PRO power metre is necessary for desktop computers. If such a metre is unavailable, it is possible to monitor estimated power data. In this study, no external power metre is employed to monitor the energy consumption of the suggested parallel algorithms; therefore, the machine must be calibrated using the numbers in Table 4.

After specifying these parameters, the application's name must be specified on the Power Usage tab before it can be launched. We then execute the programme using code blocks and examine the Joule metre to determine the application's energy consumption at each time stamp. By calculating the total energy consumed by each instance in joules, we can calculate the energy consumed by the application over time. Figure 2 illustrates the Excel file produced by the Joule metre for the PARC4 parallel algorithm.

	A	B	C	D	E	F	G	H	I	J	K	L
1	TimeStamp (ms)	Total Power (W)	CPU (W)	Monitor (	Disk (W)	Base (W)	Application (W)					
2	6.3544E+13	100.8	0.8	35	0	65	--					
3	6.3544E+13	100.3	0.3	35	0	65	Waiting for [latest_par_rc4_c1.1_second (1)] application data					
4	6.3544E+13	100.9	0.9	35	0	65	Waiting for [latest_par_rc4_c1.1_second (1)] application data					
5	6.3544E+13	103.2	3.2	35	0	65	Waiting for [latest_par_rc4_c1.1_second (1)] application data					
6	6.3544E+13	103.8	3.8	35	0	65	0					
7	6.3544E+13	103.5	3.5	35	0	65	1.1					
8	6.3544E+13	103.8	3.8	35	0	65	1.7					
9	6.3544E+13	103.9	3.9	35	0	65	1.6					
10	6.3544E+13	103.6	3.6	35	0	65	1.3					
11	6.3544E+13	103.2	3.2	35	0	65	1.3					
12	6.3544E+13	103.4	3.4	35	0	65	1.4					
13	6.3544E+13	103.5	3.5	35	0	65	1.5					
14	6.3544E+13	103.5	3.5	35	0	65	1.3					
15	6.3544E+13	103.2	3.2	35	0	65	1					
16	6.3544E+13	103.9	3.9	35	0	65	1.5					
17	6.3544E+13	103.8	3.8	35	0	65	1.8					
18	6.3544E+13	103.5	3.5	35	0	65	1.5					
19	6.3544E+13	103.6	3.6	35	0	65	1.5					
20	6.3544E+13	101.7	1.7	35	0	65	0					
21	6.3544E+13	100.5	0.5	35	0	65	0					
22	6.3544E+13	100	0	35	0	65	0					
23	6.3544E+13	100.1	0.1	35	0	65	0					
24	6.3544E+13	100.6	0.6	35	0	65	0					
25	6.3544E+13	100.5	0.5	35	0	65	--					
26	6.3544E+13	100.2	0.2	35	0	65	--					
27							18.5					

Figure 2: Data file of PARC4 consisting joules consumed at each time stamp

To compare the energy consumption of proposed and existing algorithms, the following test conditions have been utilized:
Platform 1: A laptop powered by an Intel Core 2 Duo CPU T5270 with a 1.40 GHz clock speed and 2GB of RAM, which supported a 32-bit version of Windows 7.

Platform 2: A desktop with an AMD FX (tm) - 8320 Eight-Core processor clocked at 3.5 GHz, 8 GB of RAM, and a 64-bit version of Windows 7 operating system.
In both test scenarios, Windows 7 with a Joule metre has been installed to monitor energy usage at the application level using the system's default frequency and voltage settings. The calibrated and non-calibrated states of a system are detailed in Table 1.

Table 1: Calibrated and Non-calibrated specification

State	Laptop	Desktop
Default / non-calibrated	1.2 GHz frequency and 1.2 voltage	3.5 GHz frequency and 1.332 voltage
Calibrated [operate processor by using low frequency]	1.5 GHz frequency and 0.9 voltage	2.3 GHz frequency and 0.9 Voltage

IV. RESULT AND DISCUSSION

This is the first study to compare sequential and parallel cryptographic techniques based on their energy consumption, despite the fact that many researchers are currently working in this field, with some modifying previous algorithms and others developing new energy-efficient cryptographic primitives [17]. This section discusses the exhaustive results and thorough analysis of energy usage by sequential and parallel algorithms utilizing the experimental setup presented. The following tables describe the energy characteristics of each of these algorithms.

Table 2: Energy consumed by Blowfish and PBlock with system's default frequency and voltage

Platform 1			Platform 2		
Algorithm	**µJ/B**	**MB/s**	**Algorithm**	**µJ/B**	**MB/s**
Blowfish	12.21875	1.103448	Blowfish	0.1546875	1.855072464
PBlock	12.53125	1.939394	PBlock	0.28515625	5.333333333

Table 3: Energy consumed by existing and proposed parallel algorithms for stream cipher technique using system's default frequency and voltage

Platform 1			Platform 2		
Algorithm	**µJ/B**	**MB/s**	**Algorithm**	**µJ/B**	**MB/s**
RC4	0.283008	24.7343	RC4	0.036914063	32
PARC4	0.353125	40.99608	PARC4	0.058789063	128
RC4A	0.256953	30.08226	RC4A	0.031640625	39.38461538
PARC4-I	0.345371	51.71717	PARC4-I	0.067773438	131.2820513

The results in Tables 2 and 3 are based on the system's default frequency and voltage, i.e., non-calibrated states. It can be extrapolated from these data that PBlock gives a 1.5X speedup over the serial version on Platform 1 and approximately 2.5X on Platform 2, while parallel methods require more energy than sequential algorithms. Similar to existing sequential algorithms, PARC4 and PARC4-I are much faster, but require more J/B than their sequential counterparts. The description of the result defined by Platform 2 is comparable to that of Platform 1. Platform 1's processor operates at a low frequency and voltage, whereas Platform 2's processor operates at a significantly higher frequency. Even with non-calibrated states, Platform 2 yields superior outcomes compared to Platform 1. Serial algorithms are slower and consume less energy, whereas parallel algorithms are faster but consume more energy. With parallel computing, it is possible to operate each core at a low frequency to reduce energy consumption while maintaining the same performance as sequential approaches. Thus, all studies have been conducted on calibrated processor states. Table 4 lists the low power states of Platform 2.

Table 4: Low power states of AMD-8320 processor

Voltage	Frequency
1.3	2900
1.1875	2300
1.0625	1700
0.95	1400

After calibrating the system with the parameters listed in Table 4, the energy consumption of parallel algorithms

dramatically decreased. Figure 4 demonstrates that the transmission rate is 1.18 MB/s and the energy usage is 7.1125 J/s when the PBlock method is applied to many cores with low-frequency operation.

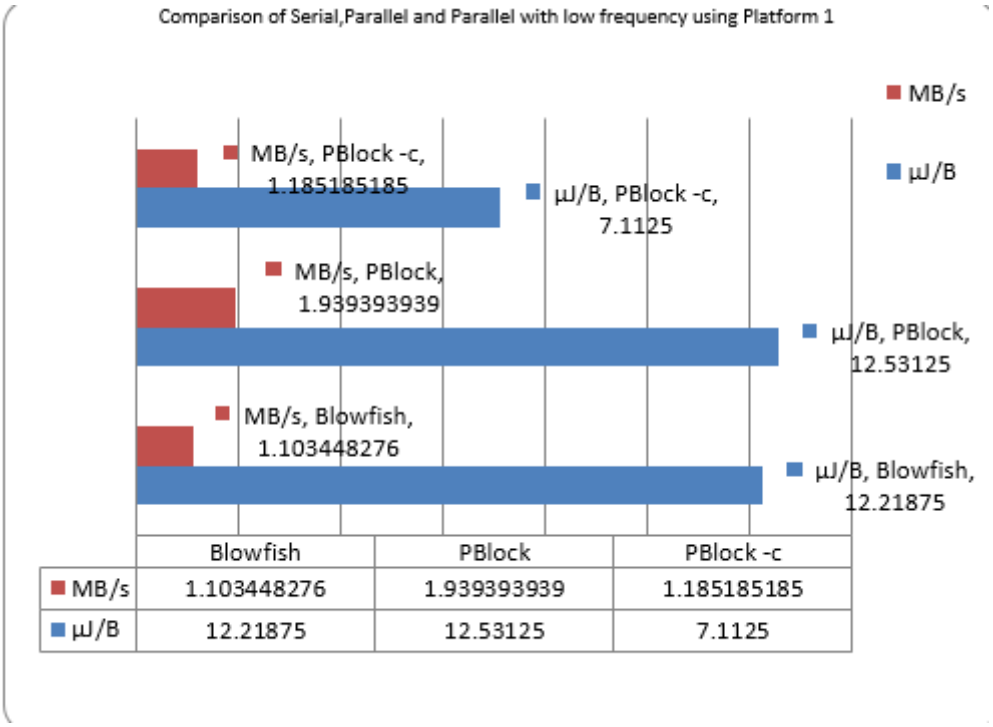

Figure 3: Comparison of serial, parallel and parallel with calibration for energy consumption using platform 1

Figure 3 demonstrates that the blowfish algorithm consumes less energy than the PBlock parallel approach at the system's default frequency and voltage, although the PBlock algorithm is significantly quicker. Conversely, if frequency is decreased, PBlock's energy consumption drops dramatically while maintaining the same level of performance. Similarly, in Figure 4, both algorithms have been conducted on Platform 2 after the frequency has been scaled down. Again, the results demonstrate that with low frequency, the PBlock algorithm consumes less energy while maintaining the same level of performance as the sequential implementation.

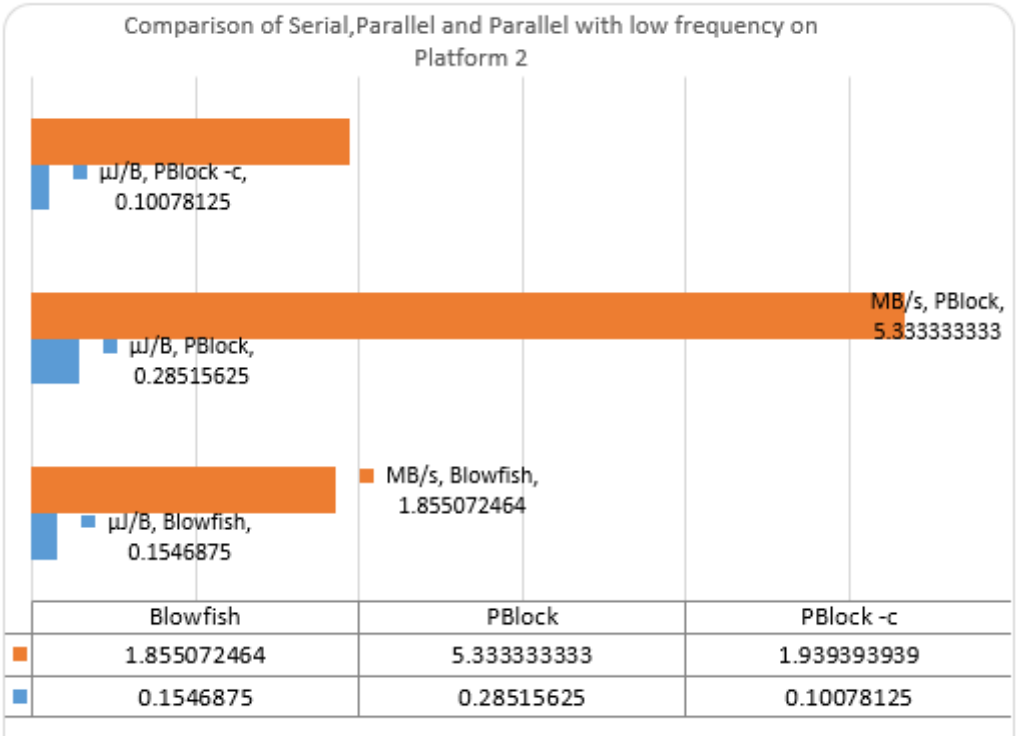

Figure 4: Comparison of serial, parallel and parallel with calibration Blowfish and PBlock for energy consumption using platform 2

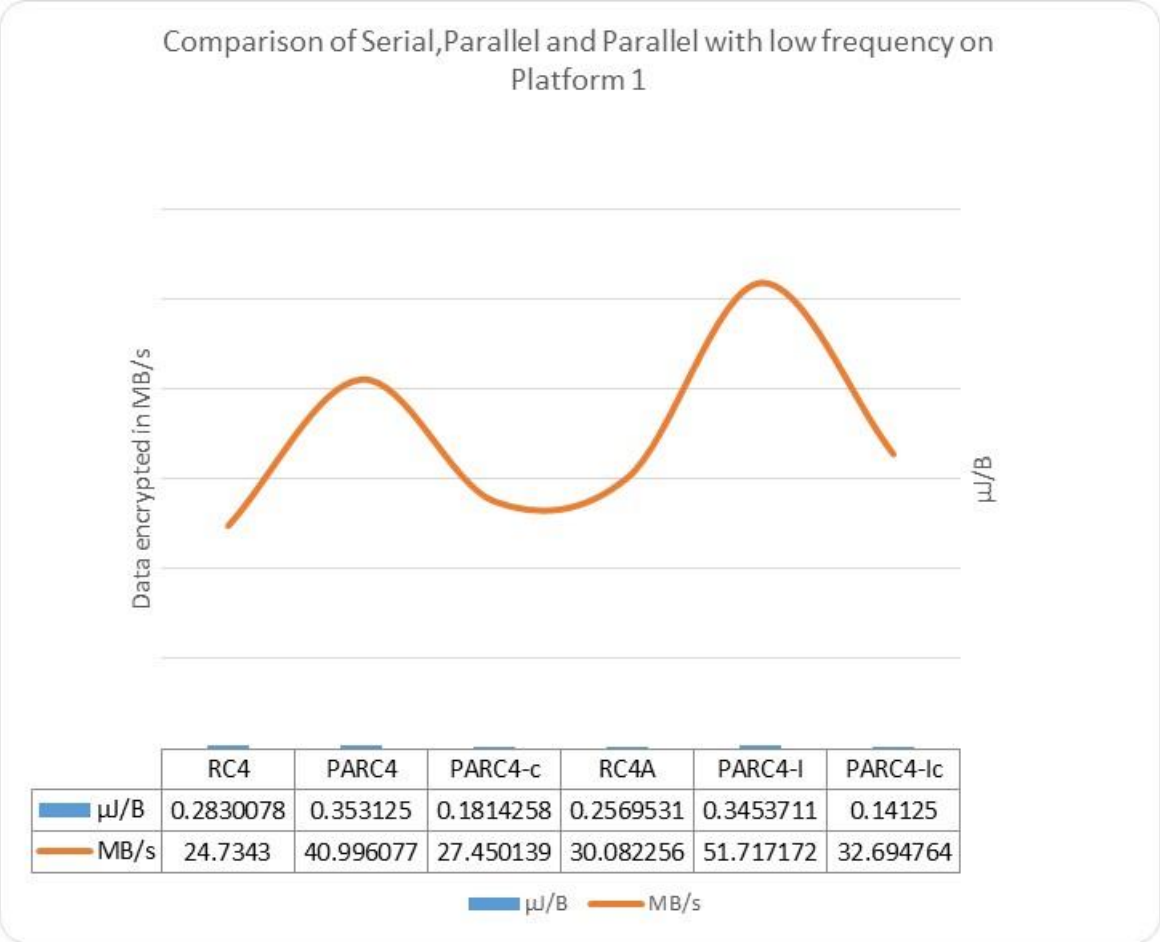

Figure 5: Serial and Parallel algorithms for stream ciphers technique with default and calibrated frequency using platform 1

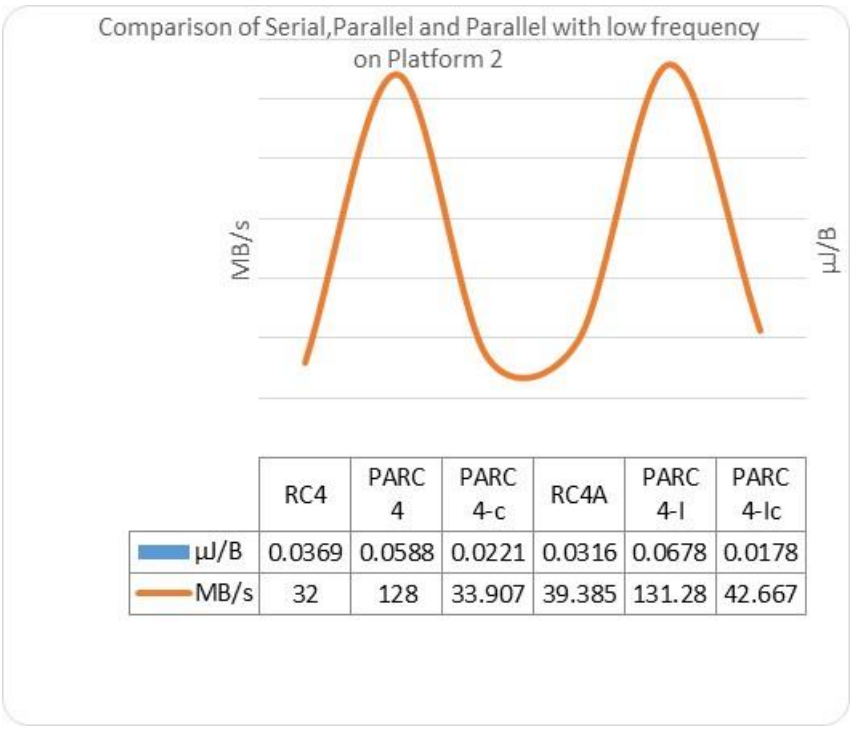

Figure 6: Serial and Parallel algorithms for stream ciphers technique with default and calibrated frequency using platform 2

Figures 5 and 6 demonstrate that while operating at a lower frequency, PARC4 and PARC4-I consume less power while maintaining a high throughput on both platforms. Thus, it has been discovered that by increasing the number of cores, the calculation performed by each core can be decreased, thereby enhancing performance in terms of time. However, a decrease in frequency will result in an increase in energy. At the same level of performance, the

parallel approach consumes less energy than the sequential algorithm.

V. CONCLUSION

The research analyzed and contrasted the energy consumption of the proposed parallel algorithms PARC4, PARC4-I, and PBlock with the serial algorithms RC4, RC4A, and Blowfish. Parallel algorithms are significantly quicker than serial algorithms, although they consume more energy. SMPs provide the opportunity to reduce the frequency and voltage of the machine via the dynamic scaling of voltage and frequency. This method can make parallel algorithms more energy-efficient. The investigation reveals that the PBlock parallel approach spends 58% less energy than the Blowfish technique, while PARC4 and PARC4-I consume 63% and 54% less energy than the RC4 and RCA algorithms, respectively. On the other hand, a speed-related compromise must be made. Therefore, the gain in time will become a gain in energy. Overall, the study found that SMPs with low frequency yielded promising results and potentially make substantial contributions to green computing and, ultimately, to society. Moreover, our results indicate that block cyphers spend more energy than stream cyphers during execution, since faster algorithms cost less energy since they operate at a higher level of power for a shorter period of time, and stream cyphers are faster than block cyphers.

REFERENCES

[1] G.N, P.K, Performance Enhancement of Blowfish Algorithm by Modifying its function. In:Innovative Algorithms and Techniques, Industrial Electronics and Telecommunications, pp. 241-244, 2007.

[2] Kapoor B.,Verma S., Power Management Design and Verification, Journal of Low Power Electronics, Vol. 7, pp. 41-48, 2011.

[3] RifaPous H., Herrera-Joancomartí J, Computational and energy costs of cryptographic algorithms on handheld devices. Future internetVol. 3, 31-48,2011.

[4] Fei, X., Li, K., Yang, W. and Li, K., 2020. Analysis of energy efficiency of a parallel AES algorithm for CPU-GPU heterogeneous platforms. Parallel Computing, 94, p.102621.

[5] Korthikanti VA., Agha G, Analysis of parallel algorithms for energy conservation in scalable multicore architectures, Proceedings of the IEEE International Conference on Parallel Processing, pp. 212-219, 2009 .

[6] Chandrakasan AP,Brodersen RW, Minimizing power consumption in digital CMOS circuits, Proceedings of the IEEE international Conference, pp. 498-523,1995.

[7] Mani K,Jee B, On the Edge: A Comprehensive Guide to Blade Server Technology, 1st edition, John Wiley & Sons2007.

[8] Schneier B., Description of a new variable-length key, 64-bit block cipher (Blowfish), In Fast Software Encryption, pp. 191-204, 1994.

[9] Schneier B., Applied cryptography: protocols, algorithms, and source code in C, Wiley, 2008.

[10] Paul S,Preneel B., A New Weakness in the RC4 Keystream Generator and an Approach toImprove the Security of the Cipher, Fast Software Encryption, pp. 245-259,2004.

[11] Handa D.,Kapoor B., PARC4: High performance implementation of RC4 cryptographic algorithm using parallelism,Proceedings of IEEE International Conference on Optimization, Reliabilty, and Information Technology (ICROIT), pp. 286-289,2014.

[12] Fluhrer S, Mantin I, Shamir A., Weaknesses in the key scheduling algorithm of RC4. In International Workshop on Selected Areas in Cryptography, pp. 1-24,2001.

[13] Handa D, Kapoor B.,PARC4-I: Parallel Implementation of Enhanced Rc4a Using Pascs And Loop Unrolling Mechanism. Computer Applications: An International Journal (CAIJ), Vol. 2 2015,DOI:10.5121/caij.2015.2203. 25.

[14] Handa D, Kapoor B., Performance Analysis of PBlock Algorithm Implemented Using SIMD Model to Attain Parallelism. InEmerging ICT for Bridging the Future-Proceedings of the 49th Annual Convention of the Computer Society of India,Vol. 2, pp. 71-80, 2015.

[15] Bekaroo G, Bokhoree C,Pattinson C, Power Measurement of Computers: Analysis of the Effectiveness of the Software Based Approach,Int. J. Emerg. Technol. Adv. Eng. Vol. 4, 755-762, 2015.

[16] Sodhro, A.H., Sangaiah, A.K., Sodhro, G.H., Sekhari, A., Ouzrout, Y. and Pirbhulal, S., 2018. Energy-Efficiency of Tools and Applications on Internet. In Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications (pp. 297-318). Academic Press.

[17] Dubrova, E., 2018. Energy-efficient cryptographic primitives. Facta Universitatis, Series: Electronics and Energetics, 31(2), pp.157-167.

ABOUT THE AUTHOR

Dr. Disha Handa is Academic Coordinator (Specialization) in University Institute of Computing, Chandigarh University (NIRF ranked). She is the former women scientist (WOS-B) from 2017-2021(June). She has completed her PhD in Parallel cryptographic algorithms in 2015. Her research areas are Acoustic analysis, Parallel programming models and machine learning models. Recently she has completed the project "Design and development of a smart back panel for women security" which is based on women's scream patterns.

ORCID ID- https://orcid.org/0000-0002-6289-2791
ResearcherID: C-9867-2017Scopus Author ID: 56154315800