International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 9 (2015) pp. 22233-22246
© Research India Publications
http://www.ripublication.com

A Novel Light Weight Cryptography for Mobile Database with Performance Analysis

[1]D. Roselin Selvarani, [2]Dr. T. N. Ravi and [3]T. Kannan Loganathan
[1]Department of Computer Science, Holy Cross College,
Tiruchirappalli, Tamil Nadu 620002, India.
drsrani09@gmail.com
[2]Department of Computer Science, Periyar E.V.R. College,
Tiruchirappalli, Tamil Nadu 620023, India.
proftnravi@gmail.com
[3]Department of Computer Science, Oxford Engineering College, Tiruchirappalli,
Tamil Nadu 620009, India.
mail9842469981@gmail.com

Abstract

Cryptography plays a vital role in securing databases. For mobile database, security is extremely important because of the mobility of the user, the portability of the device and wireless connectivity. Mobile devices used for mobile database applications are highly resource constraint in nature. Limited battery power, less memory capacity and low computing capability are some of the restrictions of mobile devices. Once the battery is drained, the data stored inside the mobile device cannot be accessed. Therefore there is a need to design a new light weight algorithm which can effectively encrypt the mobile database by considering the limitations of the mobile device. The main objective of this paper is to provide light weight cryptography, namely RSK algorithm, to secure the mobile database as well as to prove the efficiency of RSK by doing a comparative analysis with the existing symmetric key encryption algorithms. A new crypto hardware tool is designed along with a software and experiments are conducted using both hardware and software co-design. A comparative analysis is done between RSK and existing encryption algorithms to understand the performance of the newly designed RSK. Experiments are conducted at various levels by changing the inputs such as source file and key given to the encryption algorithms. It is found that RSK algorithm outperforms all the other algorithms, in all the experiments conducted, in terms of Power consumption and Time utilized by both encryption and decryption processes. Also it is proved that the throughput of RSK is higher than all the other algorithms.

Keywords: RSK algorithm, Mobile database, Security, Light weight cryptography, Power consumption, Encryption and Decryption.

1. Introduction

Security is an important issue to be considered in Mobile Computing due to the resource constraint nature of mobile devices. Mobile devices have limitations such as less computing capacity and limited battery power, low memory capacity, small screen size and narrow bandwidth. These limitations may facilitate Denial of Service attacks. It also encounters various problems due to the mobility of the user, portability of the device and wireless communication. When the database is stored inside the mobile device, that is Mobile database, it must be carefully secured or protected because of the problems faced by mobile devices. The fundamental requirements of a database are Confidentiality, Integrity, Authentication, Authorization, Access control and Non-repudiation. For mobile database, in addition to the fundamental requirements, it has to solve the problems due to the mobility of the users, the portability of the hand held devices and wireless links. In order to protect the database in general, and particularly for the mobile database Encryption plays a major role [1].

To reduce the risk of intentional or accidental disclosure of sensitive data in portable devices, the need for encryption is strongly recommended in [2, 3]. Various Encryption algorithms are widely available and used for data security. They are classified into two types namely Symmetric key (Secret key) encryption algorithms and Asymmetric key (public key) encryption algorithms. In symmetric key cryptography, a single key is used for encryption as well as for decryption. Whereas in asymmetric, two keys are used: Private key and Public key. Public key is known to all whereas Private key is known only to the authenticated user. The strength of the symmetric key encryption depends on the size of the key. As the length of the key increases, the strength of the Symmetric key encryption also increases. Symmetric key algorithms are further divided into two types such as Block cipher and Stream cipher. In block cipher, a block or group of bits are operated simultaneously whereas in stream cipher only one bit is operated at a time. Data Encryption Standard (DES), Advanced Encryption Standard (AES) and Blowfish are some of the examples for Block cipher symmetric encryption algorithms. Compared to symmetric key encryption, asymmetric key encryption algorithms are computationally intensive with more mathematical calculations and requires more computing time and battery power. Therefore, for resource constraint Mobile devices, Symmetric key algorithms are preferable [4].

The remaining part of this paper is organized as follows. Section 2 reviews the existing symmetric encryption algorithms as well as comparative analysis of the existing algorithms. Section 3 explains the methodology of the proposed work under the headings Software description and Hardware description. Section 4 analyses and discusses the results obtained through various experiments conducted and Section 5 concludes the paper.

2. Review of Literature

In [5], the authors evaluated the performance of various block cipher symmetric algorithms such as AES, DES, 3DES, RC2, Blowfish and RC6 based on the few performance metrics such as encryption time, CPU process time and CPU clock cycles and battery power. Although encryption algorithm plays a vital role in securing data or information, they consume a lot of computing resources namely CPU time, memory and battery power. They compared the results of the above said algorithms in terms of the encryption time at two different encoding bases: hexadecimal base encoding and base 64 encoding. Also a study is performed on the effect of changing pocket size, changing data types (text, documents and image) and changing key size on power consumption. The simulation results proved that there is no significant difference when the results are displayed either in hexadecimal base encoding or in base 64 encoding. It is concluded that in the case of changing pocket size, Blowfish has more significant performance than other algorithms in throughput and power consumption. It is also found that in the case of changing data type JPEG, RC2, RC6 and Blowfish have disadvantage over other algorithms in terms of time consumption. Also it is found that 3DES still has a low performance when compared to DES. The effect of key size of the algorithm proves that when the key size increases, there is a proportionate increase in battery and time consumption.

Advanced Encryption Standard (AES) was originally called as Rijndael and was developed by two Belgian cryptologists, Vincent Rijmen and Joan Daemen [6]. It uses variable key lengths such as 128,192 or 256 bits, but the default length is 256. AES provides strong encryption and was selected by National Institute of Standards and Technology (NIST) as a Federal Information Processing Standard in November 2001, which can be used to protect electronic data. AES encryption is fast and flexible and it can be implemented on various platforms especially in small devices. Brute force attack is the only effective attack known against it, in which the attacker tries to test all the characters combinations to unlock the encryption.

Scalable Encryption Algorithm (SEA) is a symmetric encryption algorithm specially designed for resource constrained devices. It is proposed for low computation time, minimum memory and power utilization, less code size with basic operations. Plain-text, key and microprocessor size are the parameters for SEA. It was originally intended for software implementations in microcontrollers, smart cards and small embedded systems. Modified SEA was designed with modular adder in a Field Programmable Gate Array device for improved performance such as less memory and power consumption with higher throughput. It is highly flexible in nature [7].

Blowfish was designed in 1993 by Bruce Schneier. It has a Feistel Network, iterating a simple encryption function 16 times. It takes a variable length key, ranging from 32 bits to 448 bits; default 128 bits. It is suitable for applications where the key does not change often, like a communications link. It is significantly faster than most encryption algorithms when implemented on 32-bit microprocessors with large data caches. Blowfish is unpatented, license-free, and is available free for all users [8].

Random Number Addressing Cryptography (RAC) is a common key cryptography which encrypts the data through Random Addressing store by using the Random Number. In this technique a register file is used as buffer, and its storage capacity is

less than the quantity of the data. Therefore the data file is physically divided and stored in blocks. RAC does only memory access without doing any arithmetic and logic operations. To implement RAC, Hardware/Software co-design was developed with a single chip multimedia mobile processor called HCgorilla. The main purpose of this processor is to achieve low power utilization and high performance. The software was developed in VC++ and run on a general PC with a Pentium CPU and 2k byte register file [9].

In [10], the authors focused on measuring and modeling power consumption of crypto algorithms and minimize vulnerability subject to a power constraint. They implemented a hardware/software experimental test-bed set-up to measure the battery power consumption of encryption algorithms. It is found that the power consumption changes linearly with the number of rounds of DES, IDEA, and GOST encryption algorithms. But the difference in the power consumption values for a fixed-key length and number of rounds is not significant for these three encryption algorithms. GOST has the smallest rate of increase in power consumption. The power consumption of RC4 varies nonlinearly with respect to the key length. By using the proposed solution, it is understood that significant gains in terms of security subject to resource constraints can be achieved. The gain in security is more pronounced, if the lower and upper bounds on the resource constraints are different for packets with different priorities.

In [11], the authors presented a detailed theoretical study on DES, 3DES, AES and Blowfish symmetric encryption algorithms. They also compared these algorithms based on CPU time, memory and power utilized. They found that among the above said algorithms Blowfish performed better than the other algorithms.

In [12], the authors provided analysis and comparison of some of the symmetric key cryptographic ciphers RC4, AES, Blowfish, RC2, DES and Triple DES on the basis of encryption time with the variation of various file features like different data types, data size, data density and key sizes. From the simulated results the authors found that encryption time does not depend upon data type and date density of the file. And also they found encryption only depends upon the number of bytes present in the file. As the size of data increases, the encryption time is also increased. They concluded that AES appears to be fastest block cipher with encryption rate of 108MB/sec at bare minimal parameter, but RC4 stream cipher with encryption rate of 270MB/sec comes out to be fastest among all analyzed cipher algorithms.

3. Methodology

The proposed light weight cryptography is called RSK algorithm named after the authors Dr. T.N. Ravi, D. Roselin Selvarani and T. Kannan Loganathan. The main purpose of developing this algorithm is to provide better security for mobile database by minimizing the computing time (both encryption and decryption time) and power utilization since the mobile devices are highly resource constraint in nature in which the mobile database is stored. It is implemented using both hardware and software co-design. As the proposed algorithm does not contain complicated arithmetic and logical computations compared to all other existing algorithms it works better.

3.1. Software Description

3.1.1. Algorithm for the Proposed Work - RSK

1. Create a class Crypto with cryptocharacters as data member and
Enc_Fun( ) and Dec_Fun( ) as member functions.
2. Enc_Fun( )
2.1. Read Source File, Destination File and Key.
2.2. Assign Source File contents into char array.
2.3. Compute the absolute of Sine of the key and assign it to d.
2.4. Convert d into a string and then encode all the characters in the string into an array of ASCII value of *Bytes.*
2.5. Modify the elements of byte array using the length of char array.
2.6. Interchange the position of the characters in the char array using the length of the byte array.
2.7. Modify the elements of char array using Cryptocharacters.
2.8. Write the contents of char array into the Destination File.
3. Dec_Fun( ) // Dec_Fun( ) is the reverse of the Enc_Fun( ).

3.2. Hardware Description

3.2.1. Hardware Tool

To execute the software effectively, a hardware tool is designed (Figure 1). The circuit diagram of the tool is also given in Figure 2. AT89C52 microcontroller in the hardware tool is used to perform the encryption and decryption function of the various algorithms.

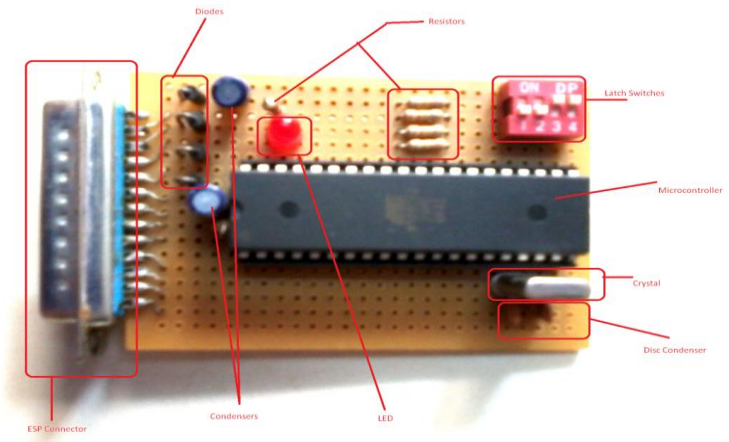

Figure 1: Hardware Tool

3.2.2. Circuit Description

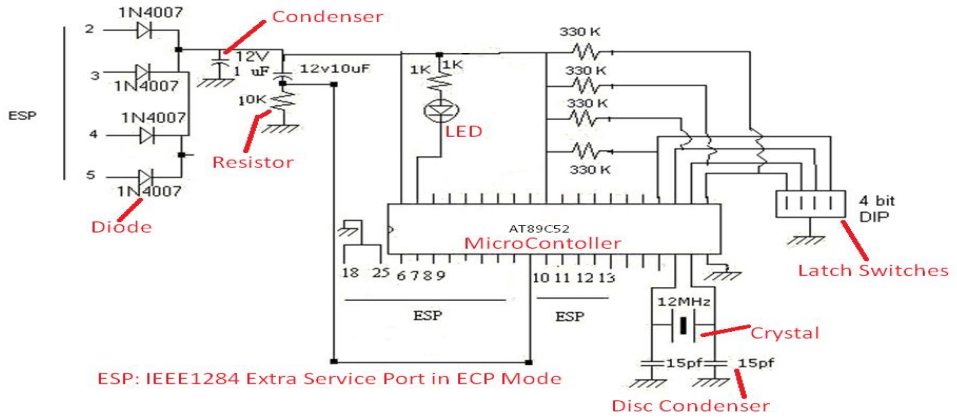

Figure 2: Circuit Diagram of The Hardware Tool

In this circuit diagram, diodes are used to draw power from port (works as rectifier diodes). Resistors are used to resist / limit current levels based on the requirement. Condensers / Disc Condensers are used to store and release power (charge & discharge) to avoid small spikes and surges in the circuit and for smooth operations of 12MHz crystal. The 12v10µF condenser is used to provide Power-on-Reset pulse to the microcontroller. Light Emitting Diode (LED) is used as status indicator. Latch Switches are used to configure the encryption / decryption algorithms selection. Crystal is used to generate clock pulses for the microcontroller. Microcontroller (AT89C52) is the brain of the circuit performing all the communication, encryption and decryption tasks. P_2 & P_3 are configured to connect with computer's Extra Service Port (ESP) in which it performs data transfers with the computer. Microcontroller is programmed to select communication speed based on the host computer's processor speed. P_0 is connected with status indicator LED. P_2 is connected with Dual in-line package (DIP) switches – used to set crypto modes of the device. There are 4 switches in the DIP, in which the first 2 switches are set to on mode always.

3.2.3. System Organization

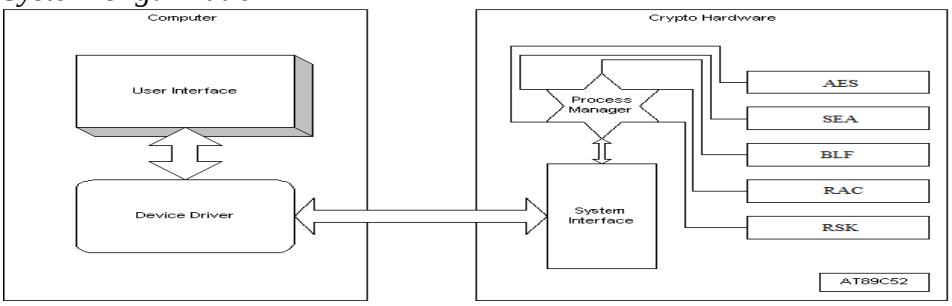

Figure 3: System Organization

The MSecure User Interface communicates with the Crypto hardware through Device driver. The System interface of the Crypto hardware helps the process manager to select the crypto algorithms based on the requirements. AT89C52 is the microcontroller used to perform the encryption and decryption process of all the algorithms or the specific algorithm depends on the selection of mode.

The software is developed using VC++. It is designed in such a way that when it is executed, a user interface (Figure 4, Figure 5) will appear. Inputs are given using this dialogue box. The implementation works in 2 different modes: Auto mode and User mode. In Auto mode, all the selected algorithms such as AES, SEA, BLF, RAC and RSK are executed one by one automatically and produce a report and graphs. Suppose the user wants to encrypt/decrypt the file using a specific algorithm among the above mentioned then the user mode is selected. The inputs are a source file, a destination filename, a report filename and a key, if it is a user mode.

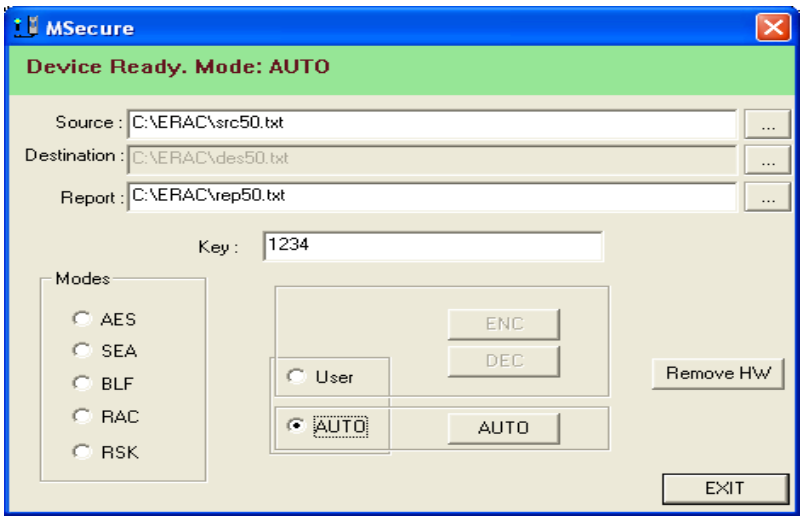

Figure 4: User Interface In Auto Mode Selection

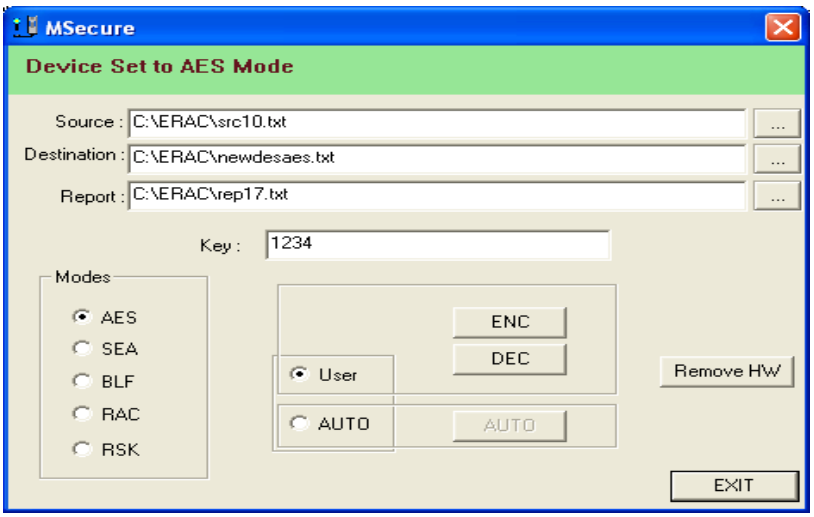

Figure 5: User Interface in User Mode Selection

If auto mode is selected, destination file name is not needed because all the algorithms are executed in a sequential way. Source file is one, which contains the actual data to be encrypted. The destination filename and the report filename do not contain anything when they are given as input. During the execution, the content of the source file is encrypted and the resultant encrypted content will be stored in the destination file and the result will be stored in the report file. The report file contains the size of source file in bytes, the starting and the ending time of both encryption and decryption process, that is the time taken for both encryption and decryption along with the amount of power consumed by encryption/decryption for every algorithm, if the mode is selected as auto. If the user mode is selected, the same type of report is obtained for the specified algorithm. Two types of graphs are generated in the auto mode by the system. The first graph represents the time taken for both encryption and decryption procedure by all the algorithms and the second graph represents the power consumption of various algorithms during encryption and decryption process.

When the software is executed, it will check for the hardware tool connection. If it is connected, then depending on the mode selection, the hardware tool will execute either all the algorithms or the user specified algorithm and produce the result. Different types of experiments are conducted as follows:

1. An experiment is conducted to understand the performance of RSK against various symmetric algorithms using a single source file and a single key.
2. Experiments are conducted to understand the performance of RSK against various symmetric algorithms by changing the Source file size.
3. Experiments are conducted to understand the performance of various symmetric algorithms by changing the Key size.

4. Results and Discussions

4.1. Comparison with a single file and a key

A source file which contains 10200 bytes of data and a key value 1234 are given as inputs for this experiment. Table 1 contains the result of the report file generated in this experiment. The bar diagrams produced by this experiment are represented in Figure 6 and Figure 7. Figure 6 depicts the time taken by the various encryption algorithms for encryption and decryption whereas Figure 7 represents the power consumption of various encryption algorithms during encryption and decryption process.

From Table 1, it is understood that

1. RSK algorithm performs better than AES, SEA, BLF and RAC in Time consumption as well as Power Consumption.
2. The time taken for both encryption and decryption processes is almost same for all the algorithms.
3. The Power consumption of RSK is very less than that of the other algorithms. Next to RSK is RAC, then SEA followed by BLF and finally AES.
4. The power consumption of encryption is the same as the power consumption of decryption procedure in RSK, RAC and SEA. In AES and BLF, the battery

power needed for encryption is greater than the power consumption required for decryption.

Table 1: Time utilized and Power consumed by various Encryption algorithms during Encryption and Decryption with same File Size and same key

Algorithm	Encryption			Decryption		
	File Size (bytes)	Time Taken (ms)	Power Consumed (mw)	File Size (bytes)	Time Taken (ms)	Power Consumed (mw)
AES	10200	3250	4.38	10200	3250	4.25
SEA	10200	2515	1.33	10200	2516	1.33
BLF	10200	2546	2.04	10200	2547	1.98
RAC	10200	2406	0.96	10200	2406	0.96
RSK	10200	2375	0.70	10200	2375	0.70

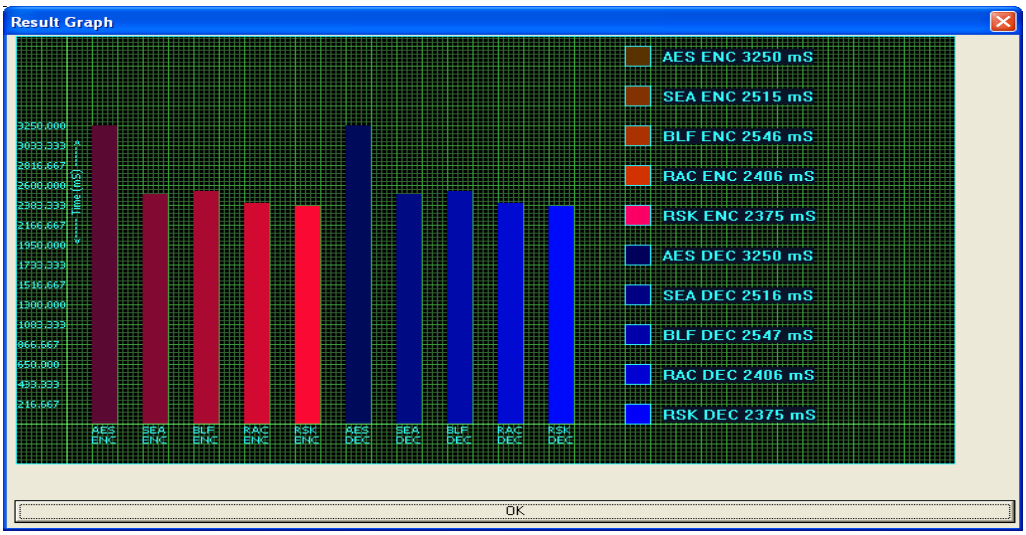

Figure 6: Time Analysis Graph

4.2. Comparison by changing the file sizes

A common key 1234 with different source files contain 10200, 20402, 30604, 40806 and 51006 bytes respectively are given as inputs for conducting 5 different experiments. The results obtained by these experiments are summarized and shown in the table (Table. 2) and the corresponding bar diagrams are represented in Figure 8, Figure 9 and Figure10. Figure 8 shows the throughput of various encryption algorithms. Figure 9 depicts the time taken by the various encryption algorithms with different file sizes for encryption and decryption. Figuren10 represents the power

consumption of various encryption algorithms with different file sizes during encryption and decryption.

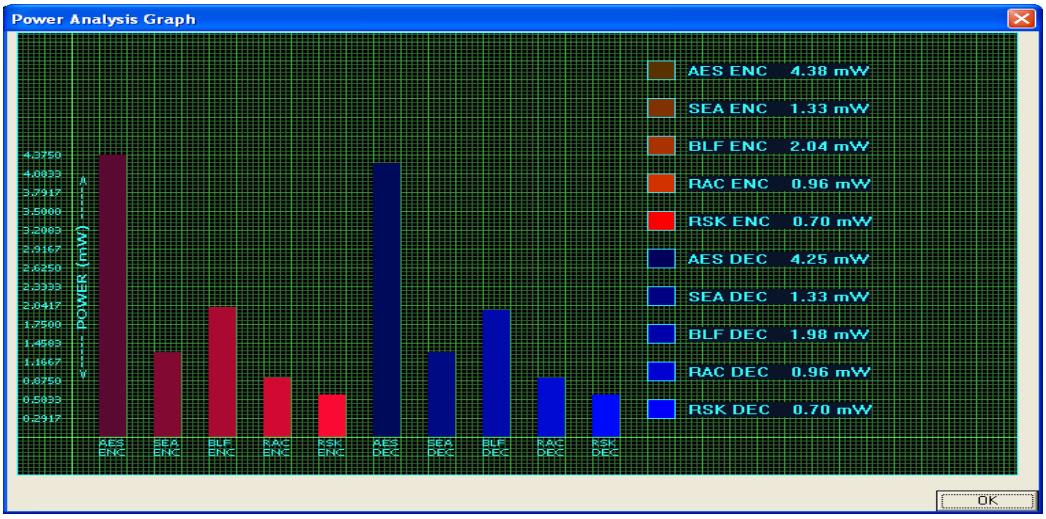

Figure 7: Power Analysis Graph

Table 2: Encryption Time (in ms) of various Encryption algorithms with different File Sizes (in bytes)

File Size (bytes)	AES	SEA	BLF	RAC	RSK
10200	3250	2515	2546	2406	2375
20402	4313	2828	2906	2609	2547
30604	5359	3140	3265	2813	2719
40806	6422	3469	3625	3031	2890
51006	7469	3782	3985	3234	3062

Table 2 reveals that

1. As the size of the source file is increased, the time taken for encryption and decryption processes by various encryption algorithms is also increased.
2. The time taken for encryption using RSK is very less compared to that of all the other algorithms irrespective of the source file size.
3. Throughput of an encryption algorithm is calculated by dividing the total file size encrypted by total encryption time for each algorithm. It is calculated from the results obtained from Table 2. The throughput is 5.71, 9.73, 9.37, 10.86 and 11.26 bytes/ms for AES, SEA, BLF, RAC and RSK algorithms respectively. The result derived proves that the throughput of RSK algorithm is higher than that of all the other algorithms. When the throughput of RSK is

compared with that of AES, RSK proves approximately double the time greater performance than that of AES (Figure 8)

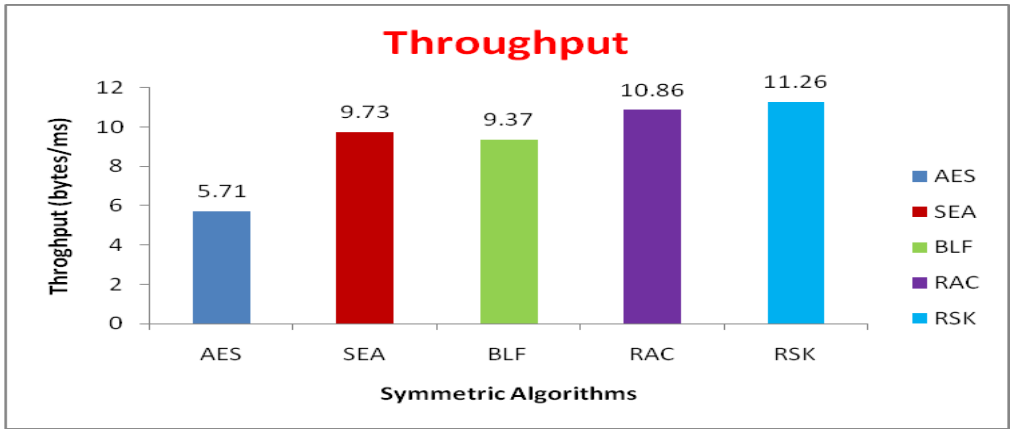

Figure 8: Throughput of Various Symmetric Encryption Algorithms

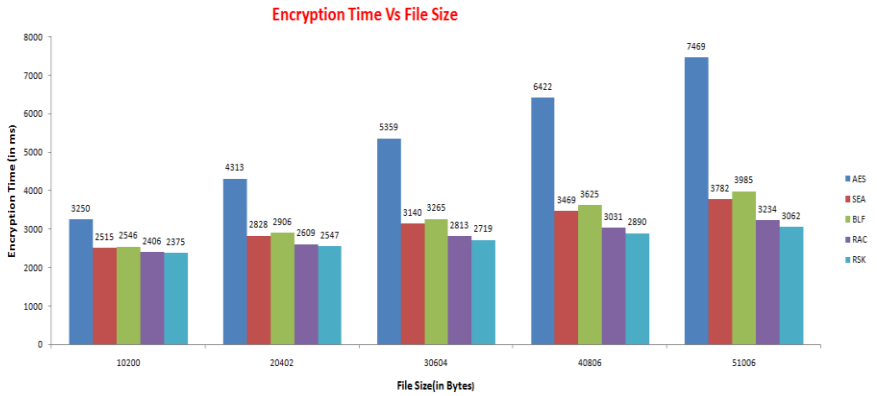

Figure 9: Time taken for Encryption of the various algorithms with different file sizes

Figure.9 illustrates the time taken by the various encryption algorithms with different file sizes during the encryption process.

A comparative analysis is made based on the power consumption of various encryption algorithms with different file sizes. The values obtained are shown in Table 3. The results obtained are diagrammatically represented using bar graph (Figure 10).

From the results obtained, it is clear that

1. The power consumption of RSK algorithm is much lesser than that of all other algorithms irrespective of the file sizes.
2. It is found that the battery power needed for RSK algorithm is approximately 6 times lesser than the power required for AES algorithm.

From the result produced by the system, it is also understood that the time taken for decryption process is almost the same as the time taken for encryption process for all

the selected algorithms with the same file size. Similarly, the power consumption of encryption process is same as the decryption process for all the selected algorithms with the same file size.

Table 3: Power consumption (in mw) of various Encryption algorithms with different File Sizes (in bytes)

File Size	AES	SEA	BLF	RAC	RSK
10200	4.38	1.33	2.04	0.96	0.70
20402	8.56	2.67	3.89	1.72	1.46
30604	12.81	3.82	5.74	2.49	2.22
40806	17.11	5.22	7.72	3.32	2.86
51006	21.32	6.43	9.64	4.15	3.56

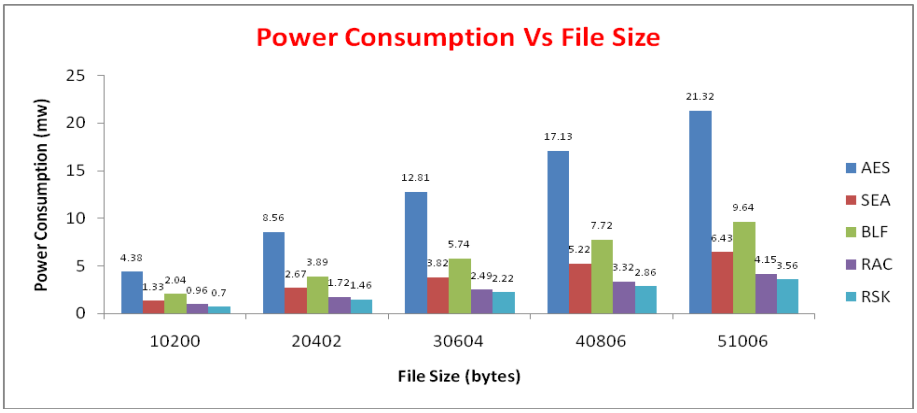

Figure 10: Power Consumed For Encryption By The Various Encryption Algorithms With Different File Sizes

4.3. Comparison by changing the Key sizes

A source file of 10,200 bytes is given as input along with various key sizes such as 128 bits, 256 bits and 512 bits respectively for conducting 3 different experiments. The performance is analyzed based on the time consumption and power consumption during encryption using AES and RSK algorithms.

Table 4: Time Consumption and Power Consumption of AES and RSK algorithms with different key sizes during encryption process

File Size (bytes)	Key Size (bits)	Time taken (ms)		Power consumed (mw)	
		AES	RSK	AES	RSK
10200	128	3250	2375	4.25	0.70
10200	256	3250	2375	4.31	0.76
10200	512	3250	2375	4.38	0.82

From the above table it is obvious that

1. Time taken for encryption process remains the same by both AES and RSK algorithms even though the key size differs.
2. The power consumed for encryption process differs by both AES and RSK algorithms as the key size differs. It is found that as the length of the key is increased, the power consumption of the encryption algorithms is also increased.
3. The power consumption of RSK algorithm during encryption process is much lesser than the power consumption of AES algorithm irrespective of the key sizes. It is found that the battery power needed for RSK is approximately 6 times lesser than the power required for AES algorithm.

The power consumption of AES and RSK algorithms is represented using the following line chart (Figure 11).

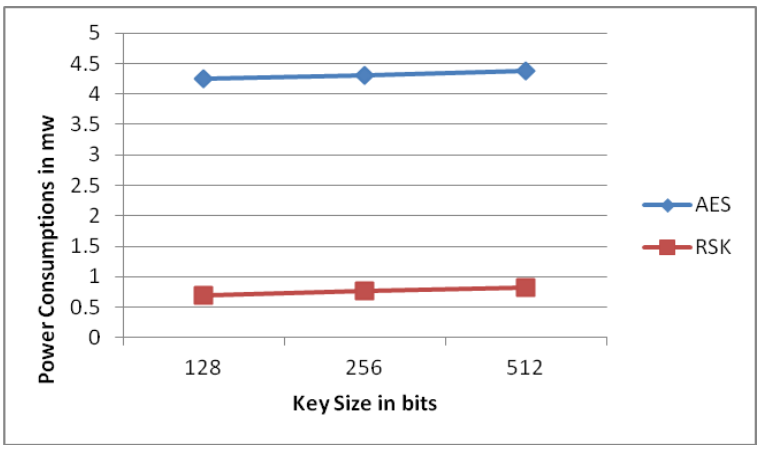

Figure 11: Power Consumption Vs Key size

5. Conclusion

Designing a new enhanced light weight algorithm is a need for today's mobile technology scenario as the application of mobile database is becoming popular day by day. So a novel RSK algorithm has been designed and implemented using a hardware tool and the performance of RSK is analyzed by doing a comparative analysis between RSK with some of the existing symmetric algorithms at various levels. The experimental results justify that RSK outperforms other algorithms. RSK algorithm performs better than AES, SEA, BLF and RAC in terms of time consumption as well as power consumption with same key size and file size. As the size of the file is increased, the time taken for various encryption algorithms is also increased. The time taken for encryption using RSK is very less compared to that of all the other algorithms irrespective of the source file size. It is found that the battery power needed for RSK is approximately 6 times lesser than the power required for AES algorithm. Time used by both AES and RSK algorithms for encryption remains the same irrespective of the key sizes. But in term of power consumption, the power

utilized by both AES and RSK algorithms differ as the key size differs. It is concluded that the proposed RSK algorithm outperforms and achieves optimum result than all other algorithms in terms of power consumption and time taken for both encryption and decryption process.

References

[1] Roselin, D., Selvarani and Ravi, T.N., December 2014, "A Review on the role of Encryption in Mobile Database Security", International Journal of Application or Innovation in Engineering & Management (IJAIEM), 3(12), pp. 76-83.

[2] Faith, M., Heikkila, 2007, "Encryption : Security Considerations for Portable Media Devices", IEEE Security and Privacy.

[3] Wesley, Chou, 2008, "Considerations for an Efficient Mobile Workforce", Wireless Broadband Technologies, IEEE, Computer Society.

[4] Chandramouli, R., Bapatla, S., Subbalakshmi, K. P., 2006, "Battery Power-Aware Encryption", ACM Transactions on Information and System Security, 9(2), pp. 162–180.

[5] Diaa Salama Abdul, Elminaam, Hatem Mohamed Abdul Kader and Mohie Mohamed Hadhoud, Dec. 2008, " Performance Evalution of Symmetric Encryption Algorithms", International Journal of Computer Science and Network Security (IJCSNS), 8(12).

[6] https://engineering.purdue.edu/kak/compsec/NewLectures/Lecture8.pdf

[7] Praveen, B., Kumar, Ezhumalai, P., Ramesh, P., Sankara, S., Gomathi, Sakthivel, P., February 2010, "Improving the Performance of a Scalable Encryption Algorithm (SEA) using FPGA", International Journal of Computer Science and Network Security, 10(2).

[8] https://www.pocketbrief.net/related/BlowfishEncryption.pdf

[9] Masa-aki, Fukase, Tomoaki, Satot, 2006, "Innovative Ubiquitous Cryptography and Sophisticated Implementation", IEEE, pp. 364-369.

[10] Chandramouli, R., Bapatla, S., and Subbalakshmi, K. P., May 2006, "Battery Power-Aware Encryption", ACM Transactions on Information and System Security, 9(2), pp. 162–180.

[11] Narender Tyagi,Anita Ganpati, 2014, "Comparative Analysis of Symmetric Key Encryption Algorithms", International Journal of Advanced Research in Computer Science and Software Engineering , 4(8), pp. 348-354.

[12] Ranjeet, Masram, Vivek, Shahare, Jibi, Abraham, Rajni, Moona, July 2014, "Analysis and Comparison of Symmetric Key Cryptographic Algorithms Based on various File features", International Journal of Network Security & Its Applications (IJNSA), 6(4), pp. 43-52.