Wireless Personal Communications (2019) 106:2023–2035
https://doi.org/10.1007/s11277-018-5920-1

Energy Efficient Data Encryption Techniques in Smartphones

Ghulam Mujtaba[1] · Muhammad Tahir[2] · Muhammad Hanif Soomro[3]

Published online: 11 August 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract
Mobile devices have been increased exceptionally in recent years, consequently data generation has also been raised exceptionally. Most of the data generated by mobile devices is transferred to servers for processing and storage. Managing security of mobile data is a necessary feature of every network and mostly encryption is used to avoid security breaches. The major challenge is that, mobile devices are very small with shortage of resources, on the other hand encryption of data requires extra energy. It is necessary to minimize energy requirements for encryption of data. For this experimental research, an android based application is developed, which optimize energy requirements for both single and double encryption techniques. AES and Blowfish encryption algorithms are used with different files sizes to test the energy requirements for single encryption, it is also examined that energy consumed by Blowfish is 119.311% more than AES. For double encryption methods, AES–Blowfish, Blowfish–AES and XTS–AES combinations of algorithms are used and energy usage is gathered. In double encryption XTS–AES consumed 13.26% less power consumption as compared to AES–Blowfish and 44.97% less then Blowfish–AES combination methods. Results of experiments revealed that AES is more energy efficient for single encryption and for double encryption XTS–AES combination requires less energy.

Keywords Energy efficiency · Encryption algorithms · Smartphone power consumption · Android operating system

✉ Ghulam Mujtaba
gmujtabakorai@gmail.com; mujtaba@gc.gachon.ac.kr

Muhammad Tahir
tahirfattani@gmail.com

Muhammad Hanif Soomro
hanif.soomro@usindh.edu.pk

[1] Department of Computer Engineering, Gachon University, Seongnam, Korea

[2] Department of Computer Engineering, Sir Syed University, Karachi, Pakistan

[3] Department of Information Technology, University of Sindh, Jamshoro, Pakistan

1 Introduction

Handheld devices have been increased rapidly in last few decades, including smart phones. The major reasons of their popularity are size and portability. Various handheld devices, including smart phones and personal digital assistance (PDA), are very small, and can be relocated easily. Moreover, small size of the devices is also featured by low energy requirement. Mostly smart devices are used for personal data processing and storage i.e. user identification information, personal pictures, social security numbers, email and text messages, contact list, and so on [1, 2].

In symmetric cryptographic algorithms, adjusting functional parameters can varied the security level, such as cipher mode, number of rounds and the key size. In recursive cryptographic algorithm, energy consumption increases with each round of recursion. The energy consumption in the cipher is directly related in two criterions: the length of the encryption key and the length of the data that is being used to encrypt [3].

Smart Devices are mainly operated with the help of resource management software, which are known as mobile operating system, these operating systems are customized as per the requirement of ecosystem in which they are used. Mobile operating systems are programmed and implemented on the hardware of smart devices, by keeping resource-shortage in mind. In comparison to other digital devices, smart devices lack various capabilities such as shortage of memory and less processing power. Symbian by Nokia, Windows Mobile by Microsoft, iOS by Apple, and Android by Google are the most common mobile operating systems. Global OS market share sales of android smartphone is 87.7% in 2017 [4]. Major responsibilities of mobile operating system include managing memory, storage and computational resources. One of prime job of mobile operating system is to manage security of stored data, but various operating systems lack proper mechanism of security management and are vulnerable to security threats, including android OS. Due to feature of open source, Android operating system attracts more attackers to explore and manipulate it [5, 6]. Usage of mobile phone users is increased by 4% in last year, as in Jan 2018 global smart phone users are 52% while laptop or desktop users are 43% [7].

Security of smart devices is mostly managed by pattern recognition and alpha-numeric passwords, but these techniques are easily vulnerable for various attack models including dictionary and brute forces attacks [8]. As an alternate method, several types of encryption are used to encrypt and decrypt partial or complete dataset. This customized encryption method is more energy efficient as only required amount of data is being encrypted [9]. Various data storage and security cryptographic algorithms are already implemented but most of them focus on only specific security objective, as Federal Information Processing Standard (FIPS) kernel cryptography is only used to support crypto related operations in the kernel [10] With the increase in security features, more computational resources will be required for smooth processing.

Portability of smart devices has raised significance of wireless networks, which requires more resources for managing privacy and security during for processing, storage and transfer of data [11]. Furthermore, wireless networks require additional devices, which also have raised the requirement of energy, memory availability and battery capacity.

This paper is divided into six sections, the remaining sections of this papers are organized as following: Section two, will describe related work, we explain some prior research which is performed on minimizing energy consumption and maximizing security of smart devices. Section three, will explain methodology of our system and discuss the system

implementation. Section four, will describe flow of the system to get power consumption of smart phone. Section five, elaborates the results, and explains experiments and discussion each of them. Finally, in section six, we conclude the research and discuss some future directions.

2 Related Work

Researchers [12], have introduced an 'AndrABEn' library based on attribute-based encryption. The library is encoded in C language and implemented in Android mobile operating system. It uses key-policy and ciphertext-policy ABE schemes. The library has various significant features including required security level, effective utilization of memory and CPU clock cycles, effective access policy management.

Different algorithms are compared in [13], including AES, Blowfish, RC6, DES, 3DES and RC2. Distinctive features are altered to measure the performance of each algorithm including packet size, key size, data type, block size and so on. With varying packet size, Blowfish produced better results in comparison to other counterparts. AES and DES revealed efficient performance when data type is changed from text to image. The researchers [14] used various programming paradigms for reduction in power consumption in mobile devices, including data storage and sensing modalities, conflicting demands for computation and communication. Their proposed solution is to use active energy profiling for reducing power consumption.

To minimize power consumption, researchers [15] examined various monitoring and sensing devices by using different routing methods, security procedures as well as system architectures and utilized time division multiple access control approach that uses heartbeat rhythm for synchronized sensor node communication, research concludes with proposed schema based on encryption techniques, that continuously operates on user authentication.

In research [16], various implementations of WBAN are implemented and analyzed with different security and privacy techniques to reduce power consumption in mobile devices. Data encryption is major technique, used to manage security and privacy of data against malicious cyber-attacks and eavesdropping. Energy consumption is reduced by can be reduced by using appropriate data network. Size of file is directly proportional to energy consumption, so it is necessary to compress the contents of file while transmitting it in network.

In previous work it is observed that researchers did not implemented double encryption and estimate their power consumption in different file sizes. While in our research we have developed android application in native android programming language and tools, to analyze power consumption and optimize energy requirements for both single and double encryption techniques are used with different files sizes to test the energy requirements for single encryption. Moreover, using double encryption security of data also increased.

3 Methodology

An experimental environment has been created by developing and implementing an android based application. This application implements selected encryption algorithm on selected file and store encrypted contents on local storage, as shown in Fig. 1. Application is named as PowerEncryptor, developed in Java and XML, using Android Software Development Tool (ADT) [17]. User can select type of encryption, category of algorithm

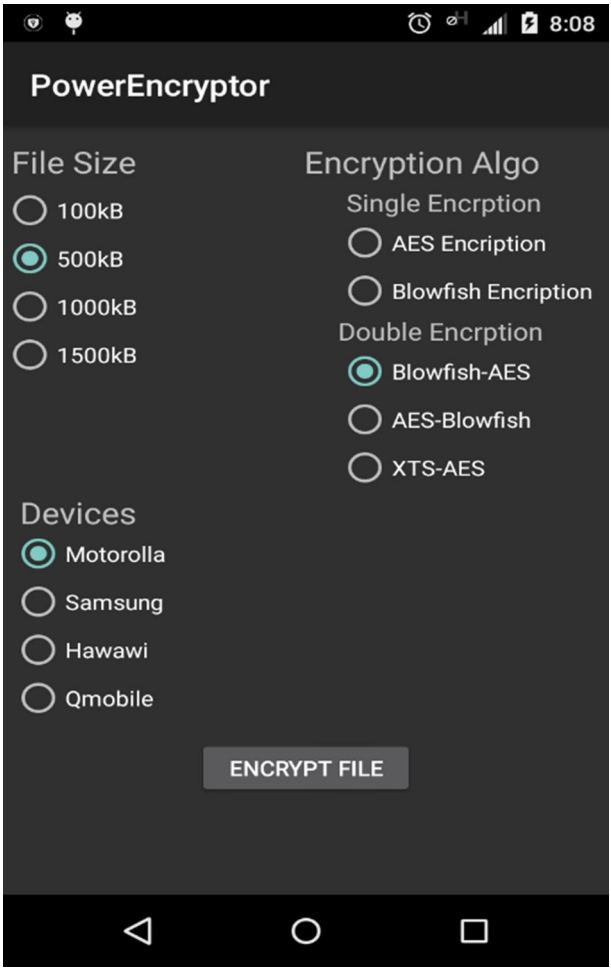

Fig. 1 Screenshot of PowerEncryptor app

and file size. Motorola-X second generation with Android 5.1 Lollipop operating system, is used for data encryption, detailed specification of mobile device is given in Table 1.

PowerTutor v1.5 [18] is slightly modified and used to calculate power consumed by PowerEncryptor during data encryption. PowerTutor has capability to calculate energy

Table 1 Motorola X hardware specifications

Component	Specification
System	Qualcomm Snapdragon S4 Pro
Microprocessor	1.7(GHz) Dual-core, Qualcomm Snapdragon S4
GPU	Quad-core Adreno 320 @ 400 MHz
Main Memory	2048 MB Ram
Storage expansion	16/32/64GB storage, no card slot
Built in storage	16 GB

consumed at system as well as at application level, and show energy consumed by each component of the system including microprocessor, display screen and so on. For this research, application level energy measurement is considered, and CPU power consumption of each application can be calculated individually at a milliwatt level of precision, as shown in Fig. 2.

Using PowerTutor we are able to calculate power consumption by PowerEncryptor during encrypting of data. For this research our main goal is to ascertain energy usage of PowerEncryptor, it can be calculated easily via Eq. 1. We modified then PowerTutor application and to expedite the calculation we utilized the trapezoidal numerical integration equation,'trapz'.

$$\text{Energy} = \int \text{Power}\, dt \tag{1}$$

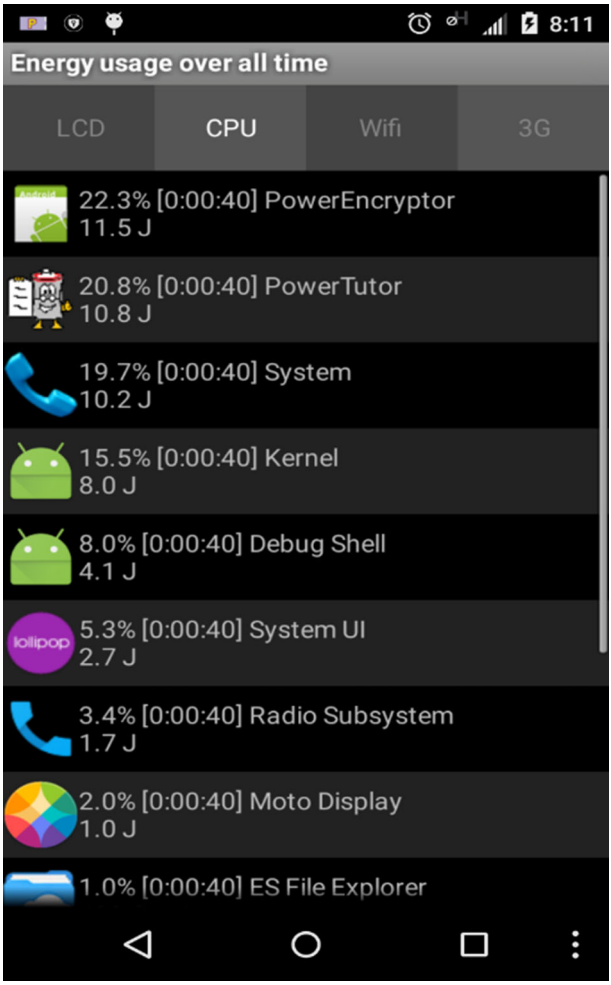

Fig. 2 Screenshot of PowerTutor app

Energy to power relation shows in Eq. 1. The PowerTutor app generates a records of power estimation of PowerEncryptor measurements and stored in .text file, which is easy to read and stored at internal storage of smartphone.

4 Discussion

PowerEncryptor application is featured by ease of use and very efficient in encryption. User interface is provided with various clearly mentioned options. Initially, user choose file size, encryption algorithm and company of the selected mobile device. User is provided with facilities of either selecting single encryption or double encryption. In case of single encryption, single algorithm choose by user will be implemented for data encryption with 128-bits private key, while for double encryption single algorithm is selected with its private key to encrypt data, and then second algorithm is selected with second private key to re-encrypt the data. In last, cypher data will be stored on external storage, as shown in Fig. 3.

During the process of encryption, PowerTutor application is used for calculation of energy consumed by PowerEncryptor. It calculates energy of both single and double

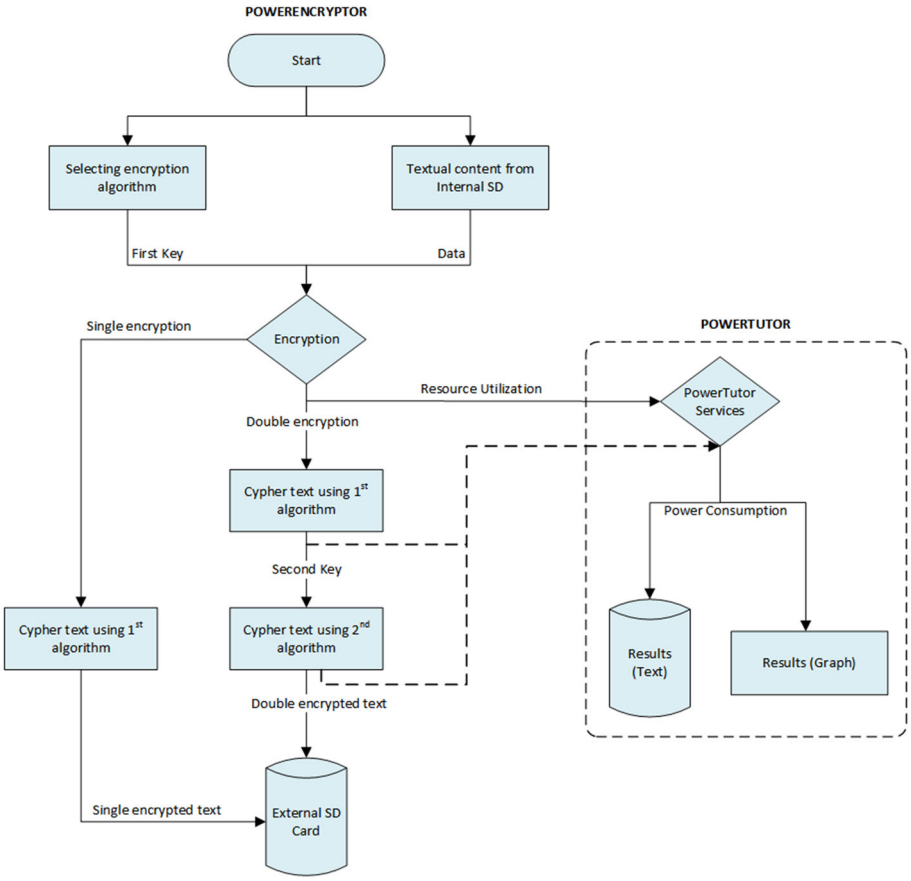

Fig. 3 Flow chart of PowerEncryptor and PowerTutor

encryption types. PowerTutor generate results in textual as well as in graphical formats, also provides the facility to store textual results.

Advance encryption standard (AES) is introduced by National Institute for Science and Technology, United States of America. It requires comparatively low energy for encryption of contents, and it also supports cross operating platforms [5, 19].

Blowfish is powerful cryptographic algorithm across variety of platforms. It performs encryption usually in round network and key-dependent substitution and permutations operations are performed in each of its round. It uses key with size ranging from 32 bits to 448 bits key. However, its performance decreases with the increase in size of key [20, 21].

Initially, single encryption method was used to make data secure with single private key, it lacks proper mechanism of security. Later, double encryption technique is introduced, in which once encrypted data is encrypted again and it uses two random private keys and two distinct algorithms to make data more protected [22].

IEEE has initiated XTS–AES algorithm, which is also approved by National Institute of Science and Technology (NIST), United States. It uses two keys for data encryption, hence demonstrated as a better option in comparison to implementation of single encryption algorithm [5, 23, 24].

5 Results

5.1 Result of File Sizes

Applying encryption algorithms on varying sizes were transmitted are shown in Fig. 4, that illustrate a function of time which estimates power consumption. In Table 2 the energy results are shown in tabulated form.

Experimental setup is arranged to acquire energy consumption during different encryption methods and numerous sizes of files. For encryption, four text files with varied sizes are taken i.e. 100 KB, 500 KB, 1000 KB and 1500 KB. On each file size, single and double encryption methods are used, and energy usage is gathered. Five different experiments are performed on each file size, and for each cryptographic algorithm. To summarize the experiments, average of all five experiments is measured and stored as shown in Table 3.

5.2 Single Encryption

For single encryption, AES and Blowfish algorithms are used and for double encryption AES–Blowfish, Blowfish–AES and XTS–AES combinations of encryption algorithms are used.

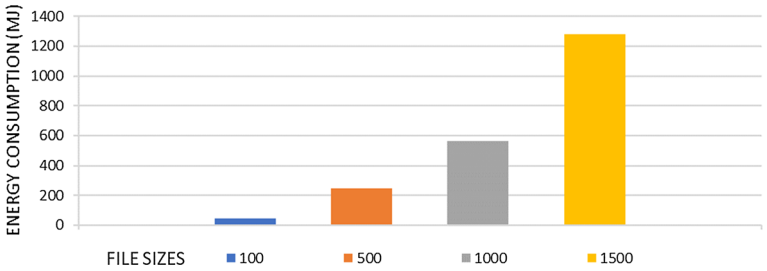

Fig. 4 Energy consumption of file transmission

Table 2 Average energy consumption of file transmission

File size (kB)	Energy consumption (mJ)
100	45.2
500	249.6
1000	563.2
1500	1281.4

Table 3 Average energy consumption of file encryption algorithms

File size (kB)	Encryption algorithms	Energy consuption (mJ)
100	AES	18.8
	Blowfish	12.6
	Blowfish–AES	204
	AES–Blowfish	22.2
	XTS–AES	13.8
500	AES	51
	Blowfish	151.6
	Blowfish–AES	219.2
	AES–Blowfish	112
	XTS–AES	47
1000	AES	79.2
	Blowfish	381
	Blowfish–AES	228.2
	AES–Blowfish	189.6
	XTS–AES	103
1500	AES	90
	Blowfish	400.6
	Blowfish–AES	232.2
	AES–Blowfish	314.8
	XTS–AES	395.4

Single encryption is separately performed by using AES and Blowfish, which reveals that AES has consumed 59.75 mJ energy, to encrypt all four text files, for five different experiments, while Blowfish consumed 236.45 mJ of energy. Hence it is proved for single encryption method, AES algorithm is more energy efficient than Blowfish algorithm, as shown in Fig. 5. Parallel to Fig. 5 from the first experiment of single encryption technique, Fig. 6 is a graph of power consumption as a function of time.

5.3 Double Encryption

Double encryption is performed by sequentially using two algorithms on same data. Three different combinations of algorithms are selected i.e. AES–Blowfish, Blowfish–AES and XTS–AES. Again, four varied sizes of text files are selected for double encryption and experiments are repeated for five times. Results for all combination of cryptographic algorithms are calculated. Results of double encryption revealed that Blowfish–AES is worst combination with the total average energy consumption, i.e. 220.9 mJ, while next worst combination is AES–Blowfish, which produces 159.65 mJ of total average energy

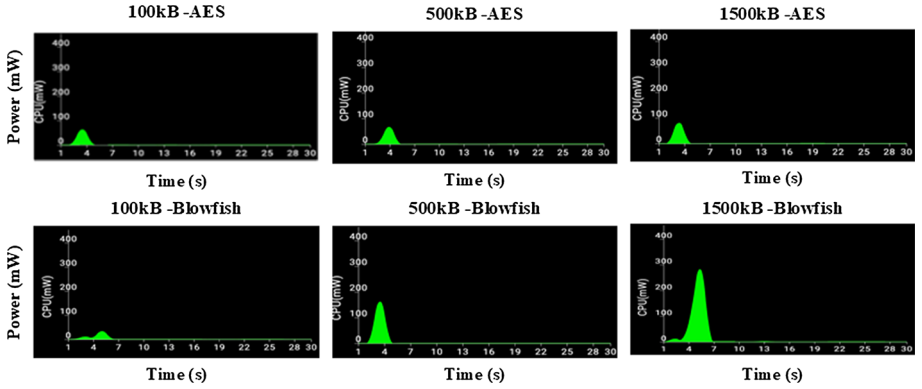

Fig. 5 Power consumption results of single encryption technique

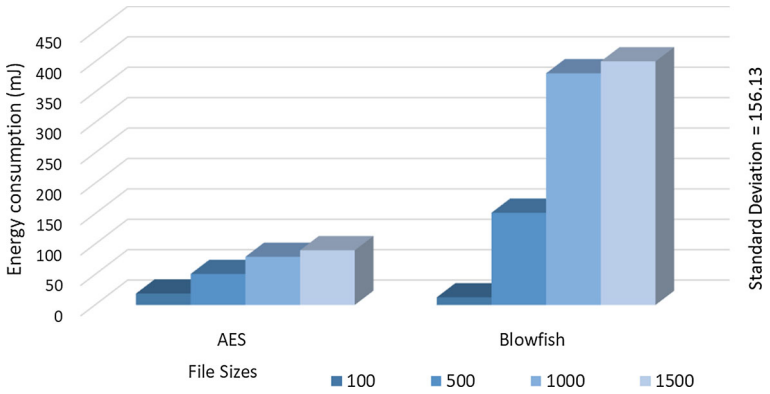

Fig. 6 Power consumption results of single encryption technique

consumption. The combination of cryptographic algorithms XTS–AES, consumed only 139.8 mJ of energy and proved as a best combination for double encryption method, as shown in Fig. 7. Similar to Fig. 7 from the second experiment of double encryptions algorithms technique, Fig. 8 is a graph of energy consumption as a function of time.

In double encryption using AES–Blowfish, energy utilization graph increased with the increase in file size, because of high energy utilization by Blowfish. However, in Blowfish–AES algorithm energy utilization slightly changed with the change in file size. In XTS–AES, energy usage is very high, for bigger file size.

Mobile devices have been increased exceptionally in recent past years, consequently data generation has also been raised exceptionally. Most of the generated data by mobile devices is transferred to servers for processing and storage. Managing security of mobile data is a necessary feature of every network and mostly encryption is used to avoid security breaches. The major challenge is that, mobile devices are very small with shortage of resources, on the other hand encryption of data requires extra energy. It is necessary to minimize energy requirements for encryption of data.

For this experimental research, an android based application is developed, which optimize energy requirements for both single and double encryption techniques. AES and Blowfish encryption algorithms are used with different files sizes to test the energy

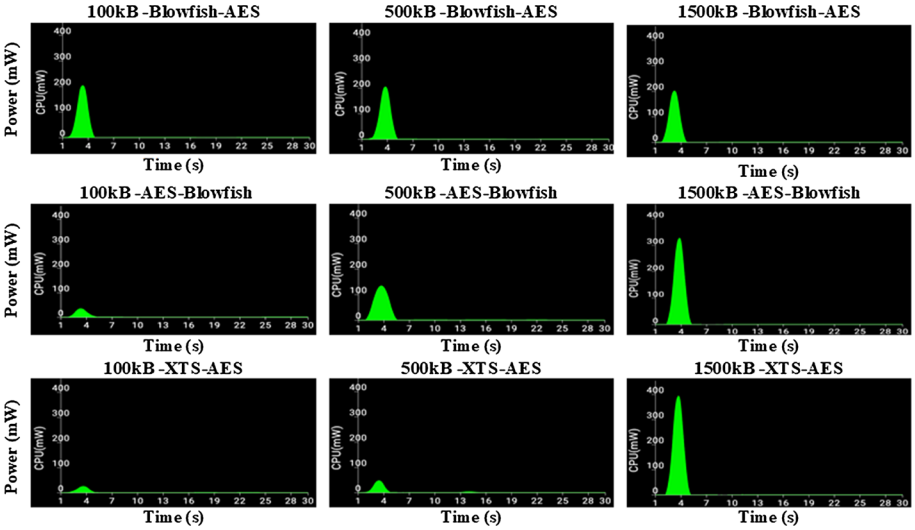

Fig. 7 Power consumption results of double encryption technique

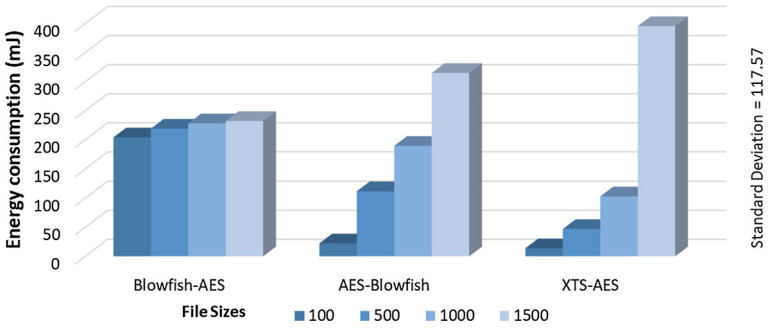

Fig. 8 Energy consumption results of double encryption technique

requirements for single encryption, it is also examined that energy consumed by Blowfish is 119.311% more than AES. For double encryption methods, AES–Blowfish, Blowfish–AES and XTS–AES combinations of algorithms are used and energy usage is gathered. In double encryption XTS–AES consumed 13.26% less power consumption as compared to AES–Blowfish and 44.97% less then Blowfish–AES combination methods. Results of experiments revealed that AES is more energy efficient for single encryption and for double encryption XTS–AES combination requires less energy.

6 Conclusion and Future Directions

The challenge of energy utilization during data encryption in mobile devices can be reduced by using appropriate choice of algorithms. Single and double encryption methods are implemented by android-based application and results are evaluated on android based mobile device. Five distinct categories of encryption algorithms are selected for experimental results including AES, Blowfish, AES–Blowfish and so on. Comparative analysis of

cryptographic algorithms for energy consumption is performed and it is evaluated that AES is the best cryptographic algorithm for single encryption method. While for double encryption method, three choices were selected, and it is revealed that, and XTS–AES is best possible combination.

Results may be enhanced by performing experiments on different devices and more cryptographic algorithms may be tested, on different platforms. Also, energy utilization during data encryption may be measured by using Graphical Processing Unit (GPU) in mobile devices, which may bring fruitful results for industry. In the future, it would be interesting to estimate power consumption of different algorithms using one core of mobile devices as well as on multicore core devices and analyze that which method will be meaningful to reduce power consumption of devices during encryption of data. Also, it seems that it may worthful to analyze power estimation of different open source OS like LineageOS [25], architectures and platform like iOS, windows phone, black berry.

References

1. Halpert, B. (2004). Mobile device security. In *Proceedings of the 1st annual conference on information security curriculum development*. Kennesaw, Georgia: ACM.
2. Mylonas, A., Kastania, A., & Gritzalis, D. (2013). Delegate the smartphone user? Security awareness in smartphone platforms. *Computers and Security*, *34*, 47–66.
3. Pfitzmann, A., Pfitzmann, B., Schunter, M., & Waidner, M. (1997). Trusting mobile user devices and security modules. *IEEE Computer Journal*, *30*, 61–68.
4. Global mobile OS market share in sales to end users from 1st quarter 2009 to 2nd quarter 2017. https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/. Accessed Aug 2018.
5. Alomari, M. A., & Samsudin, K. (2011). A framework for GPU-accelerated AES-XTS encryption in mobile devices. In *TENCON 2011-2011 IEEE region 10 conference* (pp. 144–148). IEEE.
6. Gotzfried, J., & Muller, T. (2013). ARMORED: CPU-bound encryption for android-driven ARM devices. In *Eighth international conference on availability, reliability and security (ARES), 2013* (pp. 161–168). IEEE.
7. Digital in 2018: Worlds internet users pass the 4 billion marks. https://wearesocial.com/blog/2018/01/global-digital-report-2018. Accessed Aug 2018.
8. Karri, R., & Mishra, P. (2003). Optimizing the energy consumed by secure wireless sessions: Wireless transport layer security case study. *Mobile Networks and Applications*, *8*(2), 177–185.
9. Rogers, R., Lombardo, J., Mednieks, Z., & Meike, B. (2009). *Android application development: Programming with the Google SDK*. Sebastopol: O'Reilly Media Inc.
10. Al-Subaihin, A. A., Sarro, F., Black, S., Capra, L., Harman, M., Jia, Y., et al. (2016). Clustering mobile apps based on mined textual features. In *Proceedings of the 10th ACM/IEEE international symposium on empirical software engineering and measurement* (p. 38). ACM.
11. Elminaam, D. S., Abdul, H. M., Kader, A., & Hadhoud, M. M. (2008). Performance evaluation of symmetric encryption algorithms. *IJCSNS International Journal of Computer Science and Network Security*, *8*(12), 280–286.
12. Lee, S., & Annavaram, M. (2012). Wireless body area networks: Where does energy go? In *IEEE international symposium on workload characterization (IISWC), 2012* (pp. 25–35). IEEE.
13. Chin, C. A., Crosby, G. V., Ghosh, T., & Murimi, R. (2012). Advances and challenges of wireless body area networks for healthcare applications. In *International conference on computing, networking and communications (ICNC), 2012* (pp. 99–103). IEEE.
14. Latr, B., Braem, B., Moerman, I., Blondia, C., & Demeester, P. (2011). A survey on wireless body area networks. *Wireless Networks*, *17*(1), 1–18.
15. Ameen, M. A., Liu, J., & Kwak, K. (2012). Security and privacy issues in wireless sensor networks for healthcare applications. *Journal of Medical Systems*, *36*(1), 93101.
16. DelBello, C., Raihan, K., & Zhang, T. (2015). Reducing energy consumption of mobile phones during data transmission and encryption for wireless body area network applications. *Security and Communication Networks*, *8*(17), 2973–2980.

17. Rogers, R., Lombardo, J., Mednieks, Z., & Meike, B. (2009). *Android application development: Programming with the Google SDK*. Sebastopol: O'Reilly Media Inc.
18. Gordon, M., & Zhang, L. (2011). Powertutor. http://ziyang.eecs.umich.edu/projects/powertutor/. Accessed Aug 2018.
19. National Institute of Standards and Technology (NIST), Computer Security Division. Announcing the Advanced Encryption Standard (AES). Gaithersburg, MD, USA, 2001. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf. Accessed Aug 2018.
20. Umaparvathi, M., & Varughese, D. K. (2010). Evaluation of symmetric encryption algorithms for MANETs. In *IEEE International conference on computational intelligence and computing research (ICCIC), 2010* (pp. 1–3). IEEE.
21. Stallings, W. (2006). Cryptography and network security (4th ed.). Upper Saddle River: Prentice Hall Publication (pp. 232–314).
22. Ambrosin, M., Conti, M., & Dargahi, T. (2015). On the feasibility of attribute-based encryption on smartphone devices. In *Proceedings of the 2015 workshop on IoT challenges in mobile and industrial systems*. ACM.
23. Dworkin, M. (2010). Recommendation for block cipher modes of operation: The XTS–AES mode for confidentiality on storage devices. NIST special publication 800.
24. Martin, Luther. (2010). XTS: A mode of AES for encrypting hard disks. *IEEE Security and Privacy*, *8*(3), 68–69.
25. LineageOS Android free and open-source operating system. https://www.lineageos.org/. Accessed Aug 2018.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Ghulam Mujtaba is currently perusing Ph.D. in Computer Science from Department of Computer Engineering, Gachon University Seongnam, Korea. He received B.S. Computer Sciences (2013) from COMSATS Institute of Information Technology (CIIT), Lahore, and M.S. Computer Science (2016) from Indus University, Karachi, Pakistan. His research interest includes Video Coding Standard, Mobile Computing and HCI.

Muhammad Tahir received the B.S. degree in Computer Engineering from Sir Syed University, M.E. degree in Computer System from NED University and Ph.D. in Information Science from University of Roma Tor Vergata. He is currently Associate Professor in Sir Syed University of Engineering and Technology, Karachi. His research interests include IP Switches/Routing, IPv4 Protocol, Firewall, Security and Cryptography and Wireless Networks.

Muhammad Hanif Soomro received B.S. Information Technology from University of Sindh, Jamshoro and M.S. Information Technology from Quaid-e-Awam University of Engineering, Science and Technology, Nawabshah. Currently he is perusing Ph.D. in Computer Science form National university of Computer and Emerging Science Karachi, Pakistan. His area of research are Machine Learning, Deep Learning and Security and Cryptography.