ISSN 2278-3091
Volume 9, No.5, September - October 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse223952020.pdf
https://doi.org/10.30534/ijatcse/2020/223952020

Energy Consumption Analysis on Different Authentication Protocols

Jared Harem Q. Celis[1], Andres C. Pagatpatan Jr[2]
[1]Eastern Samar State University, Guiuan, Eastern Samar, Philippines, jaredharemcelis@gmail.com
[2]Eastern Samar State University, Guiuan, Eastern Samar, Philippines, andrespagatpatan@yahoo.com

ABSTRACT

Authentication is vital part of security protocol in any data communication and services, especially that now a day's applications and services uses the internet to provide a global and easy access to the users. In any data communication, choosing an authentication protocol must consider the energy consumption of each tool. The researcher focused on the important constraints, aspects of data communication, and examined the power consumption of different types of authentication protocols. This paper presents a comprehensive analysis of energy consumption with different types of cryptographic algorithms that was utilized in the authentication protocol. The researcher explored the effect of different parameters at the convention level (for example, figure suites, validation components, and exchange sizes, and so forth) and the cryptographic algorithms level (figure modes and quality) on general vitality utilization for secure information exchanges. The network administrator uses an authentication protocol that allows local devices to access from the central server. Several number of different authentication protocols was used at the same time for different things. Physical access to the central server requires an authentication key that was configured in the device. Authentication protocol type varies from physical to network controllers.

Key words : Authentication, Cryptography, Energy Consumption, Hash Authentication, Network Security

1. INTRODUCTION

Various authentication protocols differ on how to secure the communication in a computer network. The contents of the communication are the main issue in the internet; authentication protocols are then created to verify and validate the legitimate users to prevent security breach or have fraud control and secured communication. It was developed to have maximum security depending on the compliance needs of an organization. The hardware and Network Operating System (NOS) remote servers must support usage of Authentication Protocol since there is a specific requirement with the different types [1]. With the limited electricity and tags storage, solutions to security issues differs from the flat network to the most complex managed one. Authentication protocols are the key to solve security issues in any network communication.

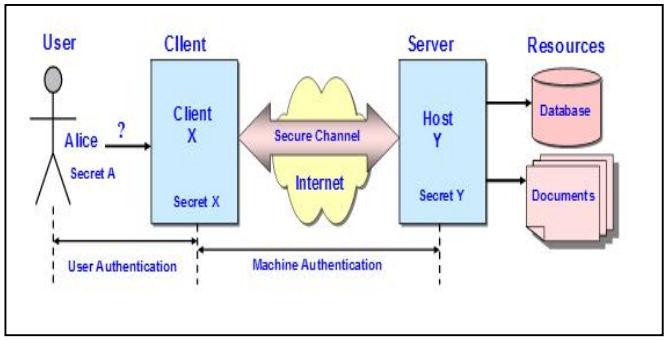

Figure 1: Authentication Process [2]

The illustration in figure 1 shows how authentication works in a client-server architecture. In cryptography, the encryption and decryption procedure, has become an essential element of most information and data security approaches [3]. If there is no authentication protocol, various threats will arise depending on the circumstances that may compromised the classified or sensitive data in an organization.

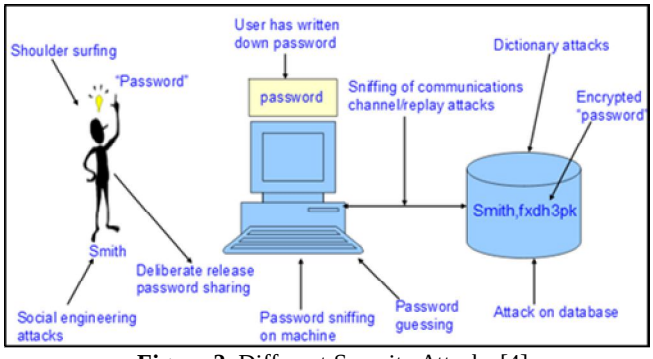

Figure 2: Different Security Attacks [4]

Peopleware, hardware, software attacks might possible arise in an insecure communication as shown in Figure 2. This shows the great significance of using an authentication protocol to avoid security breach [5].

The objectives of the study are as follows:

1. To delve and discuss the differences of each authentication tools, protocols, and algorithms used in the authentication process.
2. To explore and analyze each algorithm used in the authentication process.

2. RELATED LITERATURES

Computer security major threats were the baseline in having these authentication protocols [6]. The fortification of the security protocols of online business transactions would prohibit unauthorized access to classified information, and would further buildup the trust between the organization and its clients or clienteles. The secret technique in securing the classified information in any organization is through tight authentication. The most common type of authentication is with the use of username and password [7]. Nevertheless, using a single factor authentication as security measures is considerably weak. Similarly, some uses a partial password which is another mode of authentication that would fetch characters randomly from the registered passwords in a database. Furthermore, a new-fangled method of authentication that have been incorporated in various authentication systems is facial recognition that uses image processing instead of a password [8]. PIN and Passwords are susceptible to different deceptive threats such as brute force attack, guessing-password attack and dictionary attack. The objective of the different kinds of attacks is to obtain the login data of the legitimate user to gain unauthorized access.

However, not all kind of cyber-attacks requires advanced technical skills. Several attacks like phishing and social engineering depends on the scheming abilities of the attacker to deceive the victim to disclose sensitive data [9].

If the authentication process is tight enough and less susceptible to attacks, then this can guarantee that only legitimate users can gain access to the sensitive information and only authorized users can alter or make changes to confidential data [10]. Undeniably, security is the primary concern in any authentication processes that generally presents an unquestionable fact. Yet, the norms vary from network domains to software domains where, for an instance, a certain business desires to have security as the topmost priority and the others focuses on the aesthetics of the system [1].

Unfortunately, some security vulnerabilities in a multi-factor authentication and key agreement protocol focuses only on the single gateway node, which in fact is not appropriate in a real setting because the sensor nodes are broadly distributed in a sensor networks [12]. Device authentication guarantees the identification of a user is legitimate, and this will safeguard the confidential data, and evade threats of unauthorized device access along within the network [13]. Even Data mining algorithms can be applied in implementing Intrusion Detection Systems (IDS) [14].

The greatest concern in connecting user individuals to networks in a client-server systems is always security and confidentiality of data. Next, the user's satisfaction on the authentication and access level control on the system. The authentication and access control prevent unauthorized access of illegitimate users to sensitive data. The authentication systems should be implemented carefully and efficiently in order to save memory and computational energy [15].

3. COMMONLY USED AUTHENTICATION AND PROTOCOLS FOR NETWORK CONTROLLERS

To address the problems encountered on the security of open networks, below are the lists of widely used authentication protocols of various organizations:

3.1 Secure Socket Layer (SSL)

Oracle Database is one of those who supports SSL. It performs the server authentication to check the web server that is being accessed, client authentication to check the client's identity, and encrypted connection allows data confidentiality by means of cryptography. To have secure network communication, the client and server performs security handshake to authenticate both entities. It was developed by Netscape communication that provides secure communication for TCP and HTTP connection. It allows confidential information such as login credentials, credit card number, bank account number be transmitted in a secured manner. If TCP is connected, the user sends data to which the server responds. The message establishes a connection attribute that includes the protocol version, the session identifier, the cipher used, and compression method for client and server. After messages are exchanged, the server will send SSL certificate or may request from the client that contains the server's public key.

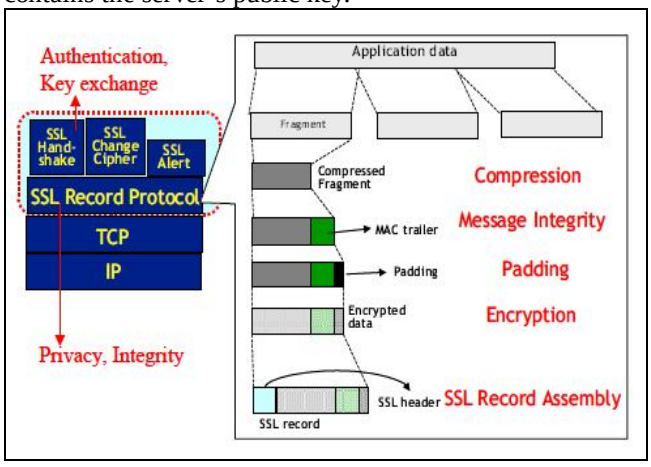

Figure 3: Expanded View of SSL Record Protocol

A Message Authentication Code (MAC) manage the veracity of message. The MAC and message may be encrypted by means of a symmetric cipher. If it uses a block symmetric cipher, padding bytes are included. The encrypted data will have a SSL header which contains different fields that include higher layer protocol.

3.2 IP SEC

Internet Protocol (IP) Security is supported by Windows XP, Windows 2000, and Windows 2003 and integrated in Active Directory Service. IP Sec uses cryptography for strong authentication and integrity but depends on the type of algorithm used. The authentication is transport –protocol independent. It is configured by creating global and local policies that show how IP packets were authenticated, validated, and encrypted for hosts or networks.

3.3 Secure Shell (SSH)

SSH is mostly available on most operating systems, a protocol that provides secure remote login and other secured network services. This is a software package that is widely used by enterprises for strong encryption and integrity protection by using Standard Hashing Algorithm (SHA 2). With the help of host key, the client may verify if they are connecting to the exact server. The server keys can be save in the client's local devices or by distribution protocol.

3.4 Kerberos

This is commonly used in a Windows 2000, Windows XP Professional, and Windows Server 2003 Active Directory domain. It can verify the user's identity, network service validity, and/or mutual authentication that verify both. The Kerberos uses tickets instead of passwords [16]. There are two (2) components for Kerberos ticket and authenticator. The ticket is for authenticating user and authenticator is for verification of the user if the same were ticket was granted to. The Kerberos server retrieves the session key by user ticket granting system. It is dependent on Key Distribution Center (KDC) to grant ticket for each user. The user can connect through an encrypted channel to the server.

3.5 Research Framework

The framework of the study is anchored on the concept of cryptography of data in network communication.

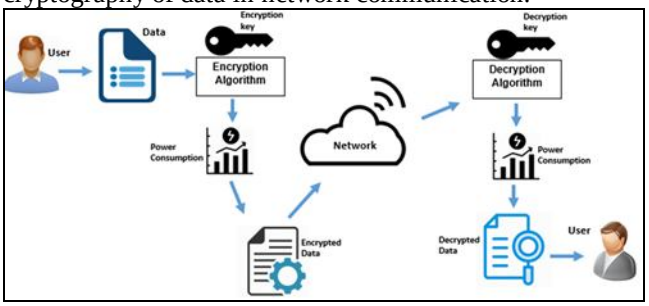

Figure 4: Conceptual Framework

The figure above illustrates how the data is encrypted and decrypted in the network communication. Different authentication protocols will be tested and analyzed to have a better understanding on the energy consumption of each cryptographic algorithm.

4. RESULTS AND DISCUSSION

Aside from ensuring the secrecy of communication, the organization must also consider the expenses on applying such tool to have an idea of the costs in energy or electricity that they may face.

4.1 Cryptographic Algorithms

In the transmission of messages between the client and server, there are digital footprints that can be traced which are saved in the network that usually contains sensitive data like the tracer log details and the user's behavior and many more. There are different kinds of attacks that takes place from the internet [17]. Encryption makes it possible for passwords to be complex in the authentication protocol. But using encryption for authentication may slow down the processing of the computer. The various methods discussed may help the reader decide on the best-suited algorithm for authentication [18].

The modern categories of encryption algorithms are asymmetric and symmetric key. The symmetric key algorithm likewise known as the private key includes Advanced Encryption Standard (AES), Data Encryption Standard (DES), Rivest Cipher 4 (RC4) that relies on a secret-key shared by both ends to encrypt and decrypt communication [19]. In collaboration with symmetric algorithms, Cryptographic Hash Functions like Message-Digest 5 (MD5) algorithm and Standard Hashing Algorithm (SHA) provides integrity by creating a unique fingerprint of messages. The asymmetric key algorithm which is also well-known as the public key uses a pair of keys to cipher and decipher a message, the first key is for encryption and the other key is for decryption. Internet Security Protocol like SSL and IP Sec uses a public key encryption algorithm to authenticate the communication [20].

There are two classes of network security cipher which are the symmetric block and stream cipher. The block cipher includes DES, 3DES, AES, *etc*., that operates on the same block size of Plain Text (PT) and Cipher Text (CT). While the stream cipher like RC4 converts PT to CT one (1) bit at a time. The input key 64 bits is stretched to create a strong encrypted key. The encryption and decryption works through repeated sequence of mathematical computation [21].

4.2 Power Consumption Evaluation Method

Power consumption can be calculated in many ways, one tool that can be used to profile the energy consumption for

encryption and decryption algorithm is a joule meter application named PowerScope. This can determine clearly the cost of energy spent during a certain period of individual procedures or processes [21]. The application can calculate the total energy consumption by assimilating the product of instantaneous current and voltage over time. The calculations is shown in formula (1) below.

$$E \simeq V_{\text{meas}} \sum_{t=0}^{n} I_t \Delta t \quad (1)$$

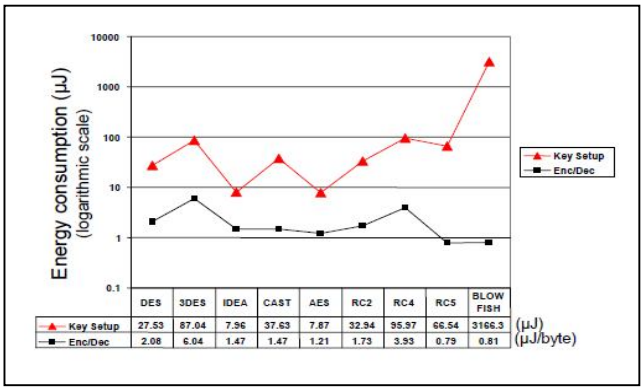

Figure 5: Energy Consumptions for Various Symmetric Cyphers

The illustration in Figure 5 shows the energy consumed due to the symmetric cipher. For each cipher, energy values and energy per byte values during the encryption and decryption were marked. Results have shown for one (1) specific mode of every cipher block the Electronic Code Book (ECB) wherein the PT block encrypts to the same CT for the same key. The Blowfish costs greatest and AES has the least energy consumption.

Hash algorithms are the least complex of all other algorithms, which means least energy consumption as well. The Hash-based Message Authentication Code (HMAC) a keyed encryption, the bid width of the key is 0 to 128 bits that consumes very small amount of energy and MD2 has the highest value.

Table 1: Energy Consumption Characteristics of Hash Functions

Algorithm	MD2	MD4	MD5	SHA	SHA1	HMAC
Energy (µJ/B)	4.12	0.52	0.59	0.75	0.76	1.16

The SHA, SHA1, MD5, and MD4 were lower consumption than the MD2 and HMAC. This test had been run using the algorithms to encrypt and decrypt 1000 bytes of file in a joule meter application tool PowerScope.

There are three (3) approved asymmetric algorithms namely Digital Signature Algorithm (DSA), Elliptic Curve Digital Signature Algorithm (ECDSA), and Rivest-Shamir-Adleman (RSA) [12]. The data in the Table 2 denotes the energy consumption of each algorithm.

Table 2: Power Cost of Digital Signature Algorithms

Algorithm	Sign [mJ]	Verify [mJ]
RSA-1024	304	11.9
ECDSA-160	22.82	45.09[3]
RSA-2048	2304.7	53.7
ECDSA-224	61.54	121.98[3]

If RSA is selected, verify operation must be done to check the server's authenticity through certificate. A random 48 bytes secret key will be done through RSA encrypt operation by means of a server's public key, in RSA sign operation a derivative random numbers of 32 bytes will prove the possession of its private key. In RSA the server can check the client's identity by means of verifying their digital signature and verify the identity to the client by decrypting the secret key using the private key. To conserve the electricity usage, they can reduce the amount of data communication in a typical internet based SSL handshake.

The data in Table 2 shows the comparison of the energy used by Rivest-Shamir-Adleman (RSA) and Elliptic-Curve Cryptography (ECC) for the generation and verification of signatures. The energy rate of RSA authentication is lesser, it is concealed by more luxurious signal operation, and both are essential for authentication. While ECDSA signs are lower than RSA signs and ECDSA authentications are within fair range of RSA authentication. The RSA 1024 to RSA 2048 the energy consumption of signaling raises by the n^{th} power of greater than seven, and ECDSA 224 ciphers is three times less expensive as ECDSA 160 ciphers. To summarize, an RSA 1024 cipher task is the same as transmitting 5,132 bytes, compared to 385 bytes for an ECDSA 160 cipher process.

The table 3 is a comparison of the energy consumed by key exchanges. The RSA key interchange protocol depends on party A in encrypting a random produced secret key with party B's public key, the party B will use the private key to decrypt. In the ECC, the client and the server performs a single ECDH operation to generate the secret key.

Table 3: Power Cost of Key Exchange Computation

Algorithm	Client [mJ]	Server [mJ]
RSA-1024	15.4	304
ECDSA-160 ECDH	22.3	22.3
RSA-2048	57.2	2302.7
ECDSA-224 ECDH	60.4	60.4

ECC key generation only includes the generation of arbitrary numbers that turn to be the client's private key, with execution of ECDH operation to calculate the public key. RSA key generation is consumes more time as it requires generating huge prime values.

Comparing the AES 128-bit keys for cryptanalysis and SHA1 for hashing, the rate of block encryption and ciphering is cheaper than the public key processes.

Table 4: Energy Cost of Symmetric Key and Hash Algorithms

Algorithm	Energy [μJ]
SHA-1	5.9 μJ /byte
AES-128 Enc	1.62 μJ /byte
AES-128 Dec	2.49 μJ /byte

The energy numbers or power consumption for AES with 128-bit keys and SHA1 are shown in Table4. AES 128 numbers contain a setup key. The values are the mean from inputs ranging from 64 to 1024 bytes.

Upon checking the power consumption of SSL handshake protocol with use of RSA and ECC algorithms (ECDSA/ECDH) implementing public key operations, SSL handshake can be executed amongst server and client neither having nor lacking client authentication. With the event of SSL handshake having no authentication, the client and server operations are executed through the following:

RSA handshake. The client executes verification and encryption using the two (2) RSA public key operations, and server execute decryption as RSA private key operation.

ECC handshake. The client executes authentication by ECDSA, and an ECDH process is executed to calculate the public secret key. Now, the server executes an ECDH process to compute the public secret key.

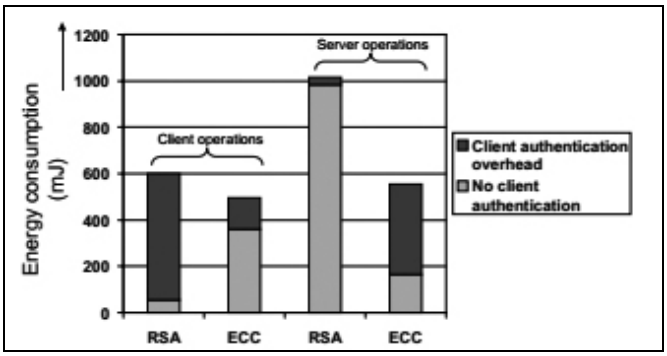

Figure 6: Energy consumption for client-server architecture in SSL Handshake verifying client authentication

In the Figure 6 above shows the energy consumption of SSL handshake by means of RSA and ECC algorithms for the client and server architecture. The client's energy consumption of RSA handshake is greatly efficient compared to ECC handshake, if the SSL handshake phase lacks client authentication. If client authentication is present in SSL handshake, ECC handshake consumes lesser energy than RSA.

5. CONCLUSION

In summary, we have performed both an experimental and theoretical study on energy consumption of different authentication protocols. Several cryptographic algorithms and energy consumption evaluation methods have been identified, discussed and tested respectively. Energy consumption on authentication protocols varies depending on the complexity of algorithms used for encryption. The more complex the cryptology, the more time it would take to process the decryption. But, the complexity of algorithm will ensure on a more secured network communication of any organization.

ACKNOWLEDGEMENT

This work has been made possible through the shared ideas and publications of various researches from different authors cited. We would like also to extend our thanks to the support of the Eastern Samar State University and its administration.

REFERENCES

1. Microsoft. **Authentication Services Protocols Overview**, available at https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-authsod/953d700a-57cb-4cf7-b0c3-a64f34581cc9, retrieved August 18, 2017.
2. H. Abie. **Different ways to authenticate users with the pros**, Norsk. Regnesentral, available at https://www.nr.no/directdownload/4380/Abie_-_Different_Ways_to_Authenticate_Users_with_the_Pros.pdf, retrieved September 7, 2017.
3. J. C. T. Arroyo, and A. J. P. Delima. **An Improved Affine Cypher using Blum Blum Shub Algorithm**, *International Journal of Advanced Trends in Computer Science and Engineering*, Vol. 9, No. 3, June 2020.
4. B. Pomeroy and K. Shorter. **Authentication Technologies**, **Trusted Information Management**, in *Handbook of Research on Information Security and Assurance*, J. N. D. Gupta, and S. K. Sharma, New York, USA: Information Science Reference, 2009, pp. 270-275.
5. K. Skračić, P. Pale, and Z. Kostanjčar. **Authentication approach using one-time challenge generation based on user behavior patterns captured in transactional data sets,** *Computers & Security*, Vol. 67, pp. 107-121, June 2017.
6. H. N. Noura, A. Chehab, and R. Couturier**. Efficient & Secure Cipher Scheme with Dynamic Key-Dependent Mode of Operation**, *Signal Proccessing: Image Communication*, Vol. 78, pp. 448-464, October 2019.

7. M. H. Barkadehi, M. Nilashi, O. Ibrahim, A. Z. Fardi, and S. Samad. **Authentication systems: A literature review and classification**. *Telematics and Informatics*, Vol. 35, Issue 5, 1491-1511, 2018 doi:10.1016/j.tele.2018.03.018.
8. A. R. L. Reyes, E. D. Festijo, and R. P. Medina. **Securing one time password (OTP) for multi-factor out-of-band authentication through a 128-bit blowfish algorithm**, *International Journal of Communication Networks and Information Security*, Vol. 10, Issue 1, 242-247, 2018.
9. A. Sharma, and S. K. Lenka. **Analysis of QKD multifactor authentication in online banking systems**, Polska Akademia Nauk. *Bulletin of the Polish Academy of Sciences*, Vol. 63, Issue 2, 545-548, 2015. doi:http://dx.doi.org/10.1515/bpasts-2015-0062.
10. A. Bani-hani, A. Majdalweieh, and A. AlShamsi. **Online Authentication Methods Used in Banks and Attacks Against These Methods**, in *The 6th International Workshop on Machine Learning and Data Mining for Sensor Networks (MLDM-SN), Procedia Computer Science*, Issue 151, pp 1052-1059, 2019. doi: http://dx.doi.org/10.1016/j.procs.2017.04.149.
11. D. Kora´c, and D. Simic. **Design of fuzzy expert system for evaluation of contemporary user authentication methods intended for mobile devices**, *Journal of Control Engineering Application and Information*, Vol. 19, Issue 4, pp. 93–100, 2017.
12. H. Guo, Y. Gao, T. Xu, X. Zhang, and J. Ye. **A Secure and Efficient Three-Factor Multi-gateway Authentication Protocol for Wireless Sensor Networks**, *Ad Hoc Networks*, Vol. 95, pp. 101965, August 2019, doi: 10.1016/j.adhoc.2019.101965.
13. P. M. Shafi, V. S. Bidve, V. V. Kimbahune, and Y. V. Gurav. **Group-based Authentication Methodologies,** *International Journal of Advanced Trends in Computer Science and Engineering*, Vol. 9, No. 2, April 2020.
14. J. H. Q. Celis, and A. C. Pagatpatan. **Predicting the Poverty Alleviation in the Province of Eastern Samar using Data Mining Techniques**, *International Journal of Recent Technology and Engineering*, Vol. 8, Issue 3, pp. 7140-7145, September 2019.
15. Microsoft. **Configuring Kerberos authentication for load-balanced Client Access servers,** available at https://docs.microsoft.com/en-us/exchange/configuring-kerberos-authentication-for-load-balanced-client-access-servers-exchange-2013-help, retrieved January 21, 2018.
16. L. Xiang-Yang. **Cryptography and Network Security,** available at http://www.cs.iit.edu/~cs549/lectures/CNS-1.pdf retrieved December 21, 2017.
17. K. G.Nida, L. N. Dayanand, B. N. Chaithanya, K. Geetha, and K. B. Manikanta. **A Virtual Machine Introspection in Cloud Computing for Intrusion Detection**, *International Journal of Advanced Trends in Computer Science and Engineering*, Vol. 9, No. 3, June 2020.
18. K. S. Kumar, R. Sukumar, P. Asrin Banu. **An Experimental Study on Energy Consumption of Cryptographic Algorithms for Mobile Hand Held Devices**, *International Journal of Computer Applications*, Volume 40, Issue 1,pp. 1-7, February 2012.
19. A. S. Wander, N. Gura, H. Eberle, V. Cupta, and S. C. Shantz. **Energy Analysis of Public-Key Cryptography on Small Wireless Devices,** in *Proc. 3rd IEEE Int'l. Conf. Pervasive Computing and Communication*, March 2005.
20. N. R. Potlapally, S. Ravi, A. Raghunathan, and N. K. Jha. **Analyzing the Energy Consumption of Security Protocols**, ISLPED, August 25-27, 2003, Seoul Korea, ACM 1-58113-682-X/03/0008, available at http://palms.ee.princeton.edu/PALMSopen/potlapally03analyzing.pdf, retrieved January 7, 2018.
21. J. Flinn, and M. Satyanarayanan. **PowerScope: a tool for profiling the power usage of mobile applications**. *Proc. Second IEEE Workshop on Mobile Computer Systems and Applications*, pp. 2–10, February 1999.
22. B. Ndibanje, Hoon-Jae Lee, Sang-Gon Lee. **Security Analysis and Improvements of Authentication and Access Control in the Internet of Things**, *Journal Sensors*, Vol. 14, Issue 8, pp. 14786-14805, August 13, 2014. doi:10.3390/s140814786.