RF Simulations for *AA_β* Cryptosystem, an Asymmetric Encryption Scheme

Syed Farid Syed Adnan, Mohd Anuar Mat Isa, Habibah Hashim
Information Security and Trusted Infrastructures Laboratory (InSTIL),
Faculty of Electrical Engineering, Universiti Teknologi MARA, Shah Alam, Selangor, Malaysia

Article Info

Article history:

Received Feb 1, 2018
Revised Apr 21, 2018
Accepted Apr 27, 2018

Keywords:

Internet of Things
IoT
Encryption
Public key

ABSTRACT

Internet of Things (IoT) is a way of providing data with the physical thing that interconnected to the network, which is the Internet. The IoT devices connected to the internet, broadcast of the data to the broker or a server, becomes an open route for attackers to gain the data and making the data becomes vulnerable. Thus, the data could be altered or spoofed by an attacker which led to security issues especially on data integrity. Therefore, the data security collected from the sensors is as important as on the servers that eventually become the big data. However, most sensors are low powered devices in term of CPU, storage, memory and batteries that cryptographic algorithm computations might give overhead to the sensors and reduce the batteries even faster than it is supposed to be. Instead of looking at symmetric encryption scheme, this paper tries to explore the capabilities of the asymmetric scheme on resource constrained devices communications. Thus, this paper presents an RF communication analysis of a low consumption asymmetric encryption, the AAβ (AA-Beta) that promising to implement on the IoT devices to secure the IoT networks. The result shows only 14% size increased in ciphertext from plaintext and the RF simulation communications show a better result in Raspbian OS environment compare to windows environment even though with same configurations.

Corresponding Author:

Syed Farid Syed Adnan,
Information Security and Trusted Infrastructures Laboratory (InSTIL),
Faculty of Electrical Engineering,
Universiti Teknologi MARA, Shah Alam, Selangor, Malaysia.
Email: farrid85@gmail.com

1. INTRODUCTION

In the world of the Internet, a lot of tasks has been migrated to online such as online banking, online transport booking and even online shopping experience. As we change towards an autonomous world where all things are unified on the internet, this information is collected and exchanged, in a way we are unintentionally open to adversaries the likelihoods of data vulnerable. The attack could be by illegally gaining the data or by tampering the data integrity regardless of the data level significant or type of things are on the internet. This statement could be supported by Hewlett Packard (HP) research carried out by the IoT Research study [1] which found that 70 percent of IoT devices running on unencrypted network services and 80 percent of the devices elevated up privacy worries.

Internet of Things (IoT) enables interconnected of things equipped with embedded such as in vehicles, houses and even cities that allows the things autonomously share the data on the internet [2]. In the year 2025, 100 billion IoT devices connected is projected [3]. IoT application example are by Ortiz et al [4] that measure heart rates together with waveforms. The data is then transmitted to the website and database. Another example is by Subashini et al. [5] IoT application to monitor the crops humidity, temperature, light and soil moisture, and transmitted to server for analysis and database which can control the growth parameters.

However, the vulnerability of the *Things* issue might arise with these interconnected devices. In a study by [1], a large number of devices neglected the encryption while updating the software updates and no encryption in communication while connecting to the internet.

As stated by Rose et al. [3], IoT open to challenges such as security where IoT implementations will encounter new and unique security challenges. Additionally, there are several attacks on the sensor itself as mentioned by Anwar et. al. [6] such as replay attack, impersonation and compromise attacks.Therefore, it is a need for IoT devices to be trusted by users in order to prevent the data from being compromised in any way [3]. In another research, Mahmoud et al [7] said that the IoT framework can be attacked at each layer giving rise to tremendous security challenges and thereby it is crucial to examine new security requirements.

These statements are then supported in a report by one of the largest logistics company in the world, the DHL. The DHL article by Macaulay et al. [8] mentioned that automation opportunities can be occupied by IoT which lead to security vulnerabilities and it is vital to ensure IoT security agenda by the collaboration of industry and governments. Interconnection of IoT devices opens a path for cybercriminals, hackers and even terrorist may harm the IoT network for their own agenda.

Moreover, to get the IoT connected, the things must be connected to the network. One thing remains important is the mobility of the devices or things to be anywhere and scattered around the home, office, warehouse or even school resulted unsuitable to be connected on wired-based. Lots of wiring need to utilize and even designing the wired layout is another challenge to consider. Wireless technology could be applied to the connected things to offer the flexibility of the things. This wireless technology could be Bluetooth technology, Radio Frequency(RF), WIFI or even with 3G or LTE [7] [9]. However, the design is based on the location, distance and energy consumption limitation for the things when connected to the internet. RF could be the main player in IoT as mentioned by Mcadams [10] and the author advised to get the RF fundamental especially in designing the wireless to provide reliability to IoT networks.

Recently, there have been lots of research done on protecting the IoT devices and the communication of the connected devices. A heterogeneous signcryption scheme for the online and offline application has been suggested by Li and Xiong [11]. The authors proposed secured channel between the sensor node and the Internet which was also able to reduce the sensor nodes cost in terms of energy and time taken while integrating security._Singh et al. [12] have designed and implemented the secure MQTT protocols (SMQTT, SMQTTSN) with new secure publish command "*SPublish*". *SPublish* publishes encrypted data with lightweight ECC techniques through optimized parameters and computation algorithms over the elliptic curve (EC). The authors have compared their scheme and the results show that it is better in terms of performance against Wang et al. [13] which utilized the Java library for an asymmetric encryption scheme implemented on an android based phone called Attribute-Based Encryption (ABE). Apart from MQTT which utilizing the TCP protocols, there is also Constrained Application Protocol (CoAP) that utilized UDP protocols. A research that introduced "SecureSense" was done by Raza et al. in 2017 [14] by integrating the CoAP and security protocol DLTS directly between end to end (E2E) IoT devices. They tested the "SecureSense" on 6LowPAN networks side and their cloud platform side namely SicsthSense which concludes their "SecureSense" timing and energy is tolerable for IoT applications. These studies show that a lot of researchers have put an effort to secure the IoT for the waves of IoT.

2. *AA_β* CRYPTOSYSTEM

The findings to secure IoT things can be symmetric or asymmetric encryption or even hybrid scheme. Thus, this motivates this research to explore the competence of an asymmetric scheme, the AA-Beta (AA_β) scheme. AA_β is an alternative asymmetric crypto scheme that was introduced by Ariffin et al. in 2012 [15]. The authors demonstrated empirically that AA_β encryption shows faster runtime compare to other asymmetric encryption schemes, such as the Rivest Shamir Adleman (RSA) and Elliptic curve cryptography (ECC).

Additionally, the decryption performance was mentioned as fast as RSA and marginally behind ECC. In their assessment by Ariffin et al. [16] have shown their scheme on Maple 13 using Windows 7 operating system and demonstrated that AA_β encryption speed is faster even for large data sets with large key sizes. The authors continue their research in [17] which shows that AA_β could handle large message size and performs faster than RSA and ECC in Maple 13 environment.

The AA_β encryption scheme algorithm that revealed by Ariffin et al. in [16] consist of key generation, encryption and decryption can be described:

2.1 *AA*AA_β Key Generation

AA_β key generation steps can be described in Algorithm 1:

Algorithm 1: ***Key Generation for*** AA_β

```
BEGIN
Choose random primes p and q with size of n-bit until $p \equiv 3 (mod\ 4)$ and $q \equiv 3 (mod\ 4)$.
Get a random integer $d > (p^2 q)^{4/9}$
Get integer e such that e.d ≡1 mod pq
    • Add multiple p.q until $2^{3n+4} < e < 2^{3n+6}$
      (if necessary)
Get A1 = $p^2 q$ where $2^{3n} < A_1 < 2^{3n+3}$
Set A2 = e
Return (n, $A_1$, $A_2$) and (pq, d)
END
```

2.2 AA_β Encryption

AA_β encryption process can be defined in Algorithm 2:

Algorithm 2: AA_β **Encryption**

```
BEGIN
Get plaintext M
Divide plaintext M to get plaintext pair $m_1$ and $m_2$
Compute ciphertext, $C = A_1 m_1 + A_2 {m_2}^2$
Return ciphertext, C
END
```

2.3 AA_β Decryption

AA_β decryption procedure is shown in Algorithm 3:

Algorithm 3: AA_β **Decryption**

```
BEGIN
Read ciphertext, C
Compute $\boldsymbol{W} \equiv C\ (\boldsymbol{mod\ pq})$
Compute $V_{i=1} \equiv x_p M_1 q + (x_q M_2 p)(mod\ pq)$
Compute $V_{i=2} \equiv x_p M_1 q - (x_q M_2 p)\ (mod\ pq)$
Compute $V_{i=3} \equiv -x_p M_1 q + (x_q M_2 p)\ (mod\ pq)$
Compute $V_{i=4} \equiv -x_p M_1 q - (x_q M_2 p)\ (mod\ pq)$
Loop
        Compute $m_{1_i} = \frac{C - Vi^2 A_2}{A_1}$
While counter<4
If $m_{1_i}$ > 0, pick the pair($m_{1_i}$, $V_i$)
Set $m_{2_i} = V_i$
Return the plaintext $M = 2^{4n} m_1 + m_2$
END
```

The AA_β encryption scheme overview is described in Figure 1. The public key *(n, A_1, A_2)* are publicly available to all devices. Along and Busu are an example of two entities to establish a secure communication channel. Along can be described as the server, while Busu is the Client.

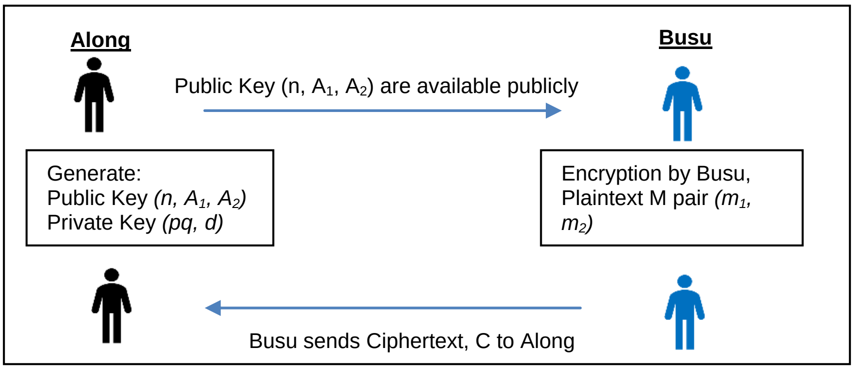
Along
Busu
Public Key (n, A1, A2) are available publicly
Generate:
Public Key (n, A1, A2)
Private Key (pq, d)
Encryption by Busu,
Plaintext M pair (m1, m2)
Busu sends Ciphertext, C to Along

Figure 1. AA_β Encryption

For the AA_β decryption scheme, the process can be summarized as in Figure 2. The private *key (pq, d)* only available to Along and the private key is used to decrypt Busu's message.

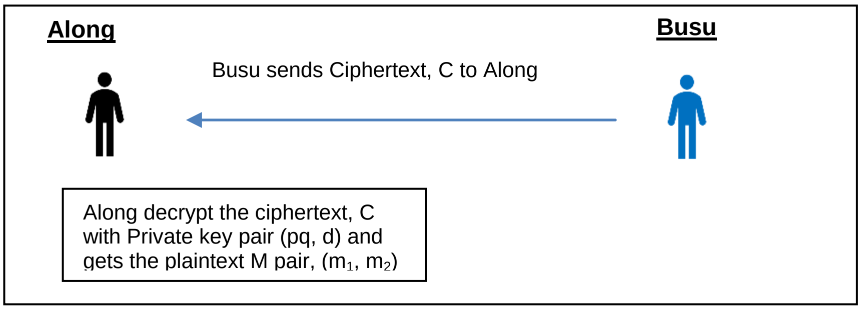
Along
Busu
Busu sends Ciphertext, C to Along
Along decrypt the ciphertext, C
with Private key pair (pq, d) and
gets the plaintext M pair, (m1, m2)

Figure 2. AA_β Decryption

3. EXPERIMENTAL SETUP

The experiments had been conducted on two types of environment. The first setup is on Windows 10 operating system (OS), which is usually for an end user. On another hand, the second environment is on Raspbian OS. The Raspbian OS is a Linux OS that has been enhanced for Raspberry Pi, embedded devices that are based on the ARM processor.The platforms are:

a) Toshiba L745 Notebook. The processor is Intel Core i5 2450M (2nd generation) with 2.5 GHz base frequency and 4Gb of DDR3 1333Mhz memory. Running with Windows 10.
b) Raspberry Pi 2, a SoC that is based on 32-bit Arm Cortex-A7 CPU at 900MHz quad-core and 1Gb of RAM shared with GPU. Running on Raspbian "Jessie" version.

On the communications between devices, the codes are written in Python scripting. The communications included transmission (Tx) and receive (Rx) while the serial speed chosen is 115200 baud rate.

On the other hands, the codes were written using the C programming language for AA_β encryption process on Linux and Raspbian environment. To handle multiple precision numbers and optimize the encryption process, the codes were written with Bignum library (GMP). GMP library is usually used for arbitrary precision arithmetic that supports the calculations up to the available memory of the system rather than having a limitation on the types of the variable itself [18].

In these experiments, the size of plaintext pair, m_1 and m_2 are based on the condition where:

$a)\ 2^{4n} < m_1 < 2^{4n+1}$
$b)\ 2^{2n-2} < m_2 < 2^{2n-1}$

Where n is the length of prime and m is the plaintext. The plaintext range used is based on Ariffin et al. definition in [15]. Meanwhile, this research assumed the key generation is generated earlier and the public key is preshared with the client and the server.

4. RESULTS AND ANALYSIS

This section presents the results of the experiments from the previous section. The results include the RF simulation transmission and receive runtime while transmitting the ciphertext.

Table 1 shows the simulation of AA_β RF transmission in Toshiba Notebook L745. While Table 2 shows the RF simulation for AA_β on Raspberry Pi 2 platform. The data size is referring to the ciphertext generated from the encryption section. The data size is represented in bits. Both results simulate the condition where a client transmitting data to the server. This could be a transmission from a sensor to a server, where low runtime and low energy is required especially for a low powered device.

Table 1. Toshiba L745 Notebook RF Transmission Simulation Runtime

Key Size	Data Size (bits)	Client Transmit(s)	Server Received(s)
1536	3587	0.068979222	0.99908226
3072	7172	0.147948718	1.037374765
6144	14340	0.304948253	1.082143804

Table 2. Raspberry Pi 2 RF Transmission Simulation Runtime

Key Size	Data Size (bits)	Client Transmit(s)	Server Received(s)
1536	3587	0.001234667	0.056090333
3072	7172	0.001349667	0.120447333
6144	14340	0.001716667	0.239137333

Figure 3 shows the comparison of RF simulation on Raspberry Pi 2(Rpi2) and Toshiba L745 Notebook which shows the transmit (TX) on Rpi2 is less than 0.05 seconds across all data size. And the runtime rises exponentially for Toshiba L745 Notebook from 0.069 seconds with 3587-bits data size and highest at 0.305 seconds with 14340-bits data size.

Furthermore, Figure 4 shows the simulation on AA_β RF receive (Rx). It is clearly seen that Rpi2 dominates the lowest Rx runtime even for large data size at 14340-bits at 0.239 seconds. While Toshiba L745 Notebook has a steady Rx runtime at 1 seconds across all data size. This might be due to windows limitation on simulation using serial communications compared to Raspberry Pi where Raspbian, a Linux OS were installed. Even though the same configuration were configured such as baud rate at 115200 baud on both platform. Figure 5 summarizes the total RF transmission on both Rpi2 and Toshiba L745 Notebook. At the highest key size of 6144-bit and 2585-bits of data, the Rpi2 dominated Toshiba L745 Notebook on total transmission time at 5.8 times, resulting reduced runtime and energy consumption to secure a plaintext and transmit using solely the AA_β encryption scheme.

Table 3 shows the AA_β encryption scheme process on Rpi2. This includes from encrypting the plaintext, simulation to transmit the ciphertext, simulation to receive the ciphertext and decrypted the ciphertext to gain the original plaintext. This process can be summarized in

Figure *6* for AA_β cryptosystem while the public key is considered pre-shared earlier. The total runtime exponentially increased from key size 1536-bits to 6144-bits and worst time for total runtime shows less than 1 seconds to for largest key size, 6144-bits and largest plaintext size at 12287-bits. On the other hand, the plaintext size was increased around 14.2% to 14.3% to ciphertext when encrypted to AA_β for all key size.

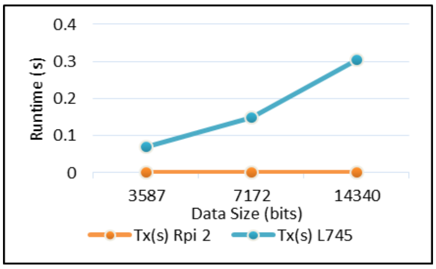

Figure 3. AA_β RF Transmit (Tx) simulation comparison of Raspberry Pi 2(Rpi2) and Toshiba

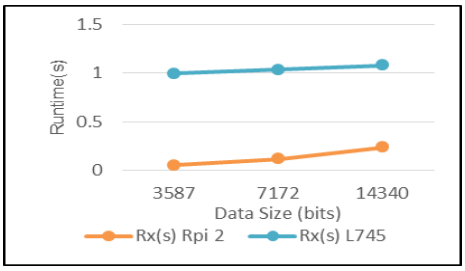

Figure 4. AA_β RF Receive (Rx) simulation comparison of Raspberry Pi 2(Rpi2) and Toshiba

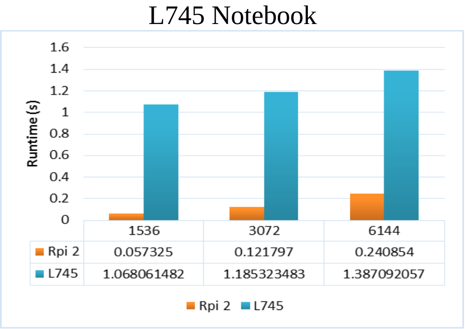

Figure 5. AA$_\beta$ Comparison on Total of RF Transmission runtime simulation from Client to Server on Raspberry Pi 2 (Rpi2) and Toshiba L745 Notebook platform

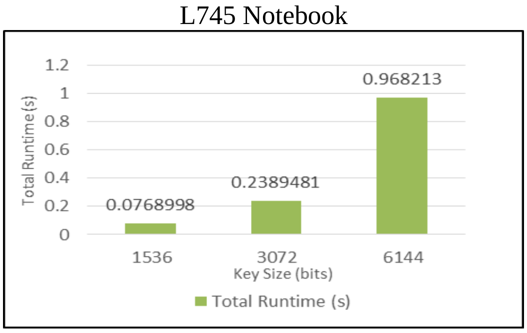

Figure 6. *AA$_\beta$* Total Runtime for Encryption, Decryption and Transmission on Raspberry Pi 2

Table 3. *AA$_\beta$* Encryption, Decryption and Transmission Runtime on Raspberry Pi 2

Key Size	Plaintext Size	Encryption (s)	Decryption (s)	Client Transmit(s)	Server Received(s)
1536	3073	0.0009587	0.0186161	0.001234667	0.056090333
3072	6143	0.001755	0.1153961	0.001349667	0.120447333
6144	12287	0.0039217	0.7234373	0.001716667	0.239137333

Given the scenario above, given the measured current on Rpi2 is 0.258 Amp at 5 Volt, the energy can be calculated from the total runtime above where energy is:

$$Energy\ (E) = Power\ (P) x\ Time\ (t)$$

Thus, for encrypting a message, transmit through RF simulator and decrypt the ciphertext back with a key size of 6144-bit, the energy is taken 4.84 Joules. Let say a 2200maH battery pack is connected to the Rpi2, the Rpi2 can stand up to 30,697.2 hours if the encryption is performed continuously. Which is equivalent to 1,279.05 days.

5. CONCLUSION

This research presented an investigation of an asymmetric encryption scheme, the *AA$_\beta$* encryption scheme in terms of encryption and decryption time and in addition, the RF transmission runtime simulation. The simulation on Raspberry Pi 2 running on Raspbian OS has resulted in 82.6% RF simulation transmission runtime enhancement compared to Toshiba L745 Notebook with Windows 10 OS. The reason for slower RF transmission runtime simulation in Toshiba L745 Notebook might due to default configuration on windows OS used even with same baud rate speed for both platforms. Nevertheless, this demonstrated that it is viable to explore the *AA$_\beta$* ability in real RF communications for IoT due to faster encryption time achieved in embedded devices and only 14.2% ciphertext size increased after plaintext encrypted. This RF simulation have zero error as it is in a controlled environment and does not have data loss or delay even up for 200 times data transmission as suggested by Putra et al. in transmitting the data in wireless condition [19]. In a real IoT device, the RF communication between devices might have challenges in transmission configuration. Nonetheless, this scheme may be beneficial to low powered devices to encrypt the data with asymmetric encryption scheme offering simple math calculation on encryption, addition, multiplication and square compared to RSA which required modulus operation, resulting additional overhead in encrypting a message. The future works on this research can be testing of *AA$_\beta$* encryption scheme on real RF transmitter, Wifi or Bluetooth connection, depending on the environment for the IoT taking place. Furthermore, the scheme can be compared with another asymmetric scheme such as RSA and ECC on RF communications.

ACKNOWLEDGEMENT

The authors would like to acknowledge the Ministry of Higher Education (MOHE) Malaysia for providing the grant 600-RMI/NRGS 5/3 (5/2013) and Universiti Teknologi MARA (UiTM) for supporting this research work.

REFERENCES

[1] Hewlett Packard Enterprise, “Internet of Things Research Study 2015 Report,” no. July, p. 4, 2015.

[2] Margaret Rouse and Ivy Wigmore, “Internet of Things (IoT),” *Tech Target*. [Online]. Available: http://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT. [Accessed: 29-Feb-2016].

[3] K. Rose, S. Eldridge, and C. Lyman, “The internet of things: an overview,” *Internet Soc.*, no. October, p. 53, 2015.

[4] K. J. P. Ortiz, J. P. O. Davalos, E. S. Eusebio, and D. M. Tucay, “IoT: Electrocardiogram (ECG) monitoring system,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 10, no. 2, pp. 480–489, 2018.

[5] M. S. M, S. Das, S. Heble, U. Raj, and R. Karthik, “Internet of Things based Wireless Plant Sensor for Smart Farming,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 10, no. 2, pp. 456–468, 2018.

[6] K. N. Q. Raja Waseem Anwar, Majid Bakhtiari, Anazida Zainal, “Security in Wireless Sensor Network : Approaches and Issues,” *TELKOMNIKA Indones. J. Electr. Eng.*, vol. 15, no. 3, pp. 584–590, 2015.

[7] R. Mahmoud, T. Yousuf, F. Aloul, and I. Zualkernan, *“Internet of things (IoT) security: Current status, challenges and prospective measures,”* 2015 10th International Conference for Internet Technology and Secured Transactions (ICITST). pp. 336–341, 2015.

[8] J. Macaulay, L. Buckalew, and G. Chung, “Internet of Things in Logistics,” *DHL Trend Res.*, vol. 1, no. 1, pp. 1–27, 2015.

[9] F. K. Santoso and N. C. H. Vun, *“Securing IoT for smart home system,”* Proc. Int. Symp. Consum. Electron. ISCE, vol. 2015–Augus, pp. 5–6, 2015.

[10] B. Mcadams, “RF FUNDAMENTALS for the INTERNET of THINGS,” 2017. [Online]. Available: https://oleumtech.com/wp-content/uploads/downloads/references/rf-fundamentals-for-iot-iiot-oleumtech.pdf.

[11] F. Li and P. Xiong, “Practical secure communication for integrating wireless sensor networks into the internet of things,” *IEEE Sens. J.*, vol. 13, no. 10, pp. 3677–3684, 2013.

[12] M. Singh, M. A. Rajan, V. L. Shivraj, and P. Balamuralidhar, *“Secure MQTT for Internet of Things (IoT),”* Proc. - 2015 5th Int. Conf. Commun. Syst. Netw. Technol. CSNT 2015, pp. 746–751, 2015.

[13] X. Wang, J. Zhang, E. M. Schooler, and M. Ion, *“Performance evaluation of Attribute-Based Encryption: Toward data privacy in the IoT,”* 2014 IEEE Int. Conf. Commun. ICC 2014, pp. 725–730, 2014.

[14] S. Raza, T. Helgason, P. Papadimitratos, and T. Voigt, “SecureSense: End-to-end secure communication architecture for the cloud-connected Internet of Things,” *Futur. Gener. Comput. Syst.*, vol. 77, pp. 40–51, 2017.

[15] M. R. K. Ariffin, M. A. Asbullah, N. A. Abu, and Z. Mahad, “A New Efficient Asymmetric Cryptosystem Based on the Integer Factorization Problem of N=P^2 .q,” *Malaysian J. Math. Sci. 7(S) 19-37 Spec. Issue 3rd Int. Conf. Cryptol. Comput. Secur. 2012*, vol. 7, pp. 1–6, 2012.

[16] Z. Mahad and M. R. K. Ariffin, *“AABeta public key cryptosystem - A new practical asymmetric implementation based on the square root problem,”* in Computing and Convergence Technology (ICCCT), 2012 7th International Conference on, 2012, pp. 584–588.

[17] M. R. K. Ariffi and Z. M, “*AAβ Public Key Cryptosystem – A Comparative Analysis Against RSA and ECC,*” Comput. Converg. Technol. (ICCCT), 2012 7th Int. Conf., pp. 589–594, 2012.

[18] The GNU MP Bignum Library, “The GNU Multiple Precision Arithmetic Library (GMP) Library.” [Online]. Available: https://gmplib.org/. [Accessed: 25-Mar-2016].

[19] G. D. Putra, A. R. Pratama, A. Lazovik, and M. Aiello, *“Comparison of energy consumption in Wi-Fi and bluetooth communication in a Smart Building,”* in 2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC), 2017, pp. 1–6.