Malaysian Journal of Mathematical Sciences 13(2): 187–200 (2019)

MALAYSIAN JOURNAL OF MATHEMATICAL SCIENCES

Journal homepage: http://einspem.upm.edu.my/journal

Enhanced AA_β Cryptosystem - A Comparative Analysis

Mahad, Z.[1], Asbullah, M. A. *[1,2], and Kamel Ariffin, M. R.[1,3]

[1] *Laboratory of Cryptography, Analysis and Structure, Institute for Mathematical Research, Universiti Putra Malaysia, Malaysia*
[2] *Centre of Foundation Studies for Agricultural Science, Universiti Putra Malaysia, Malaysia*
[3] *Department of Mathematics, Faculty of Science, Universiti Putra Malaysia, Malaysia*

E-mail: ma_asyraf@upm.edu.my
* *Corresponding author*

Received: 31 August 2018
Accepted: 29 April 2019

Abstract

A major enhancement strategy of the AA_β cryptosystem is currently proposed which incorporates the Rabin-p decryption method upon its original design while maintaining the key generation and encryption procedures. Consequently, such strategy improved the decryption procedure of the AA_β cryptosystem compare to any previously proposed design. In this paper, the aim is to provide a comparative analysis of the new design of the AA_β cryptosystem with the original and the other enhancement methods in existence. The scope of this work is a comparative analysis upon the decryption procedure only. The results show that the enhanced version of the AA_β cryptosystem is efficient in term of faster running time and lower memory consumption.

Keywords: AA_β Cryptosystem, Internet of Things, Rabin-p Cryptosystem, Post-Quantum, Multivariate, Embedded System

1. Introduction

The AA_β public key cryptosystem was introduced earlier in 2012, as a visualization of an asymmetric encryption that utilized the concept of the Bivariate Function Hard Problem (Mahad and Ariffin, 2012). A completed workable version of the AA_β cryptosystem was introduced later in Ariffin et al. (2013). Since then, a considerable amount of research has been published on the AA_β public key cryptosystem.

Ariffin and Mahad (2012) performed an experimental analysis in term of encryption speed to show that the AA_β cryptosystem is faster than the RSA and the ECC cryptosystem. Later, Asbullah and Ariffin (2014) proposed to use the Garner's algorithm instead of Chinese Remaindering Theorem computation. As a result, the alteration influences the decryption process becomes faster than the original one. Furthermore, a comparative analysis against Rabin-Takagi (Takagi, 1998) and HIME(R) cryptosystem (Nishioka et al., 2002) was also provided in the work of Asbullah and Ariffin (2014).

In recent years, there has been an increasing amount of literature discussing the practicality of the AA_β cryptosystem. For instance, Adnan et al. (2016a) successfully implements the said algorithm as a lightweight asymmetric encryption scheme on an embedded system device. Additionally, based on the results of a timing analysis of the AA_β encryption in Adnan et al. (2016b) suggest that such cryptosystem can be for all intents and purposes installed for the Internet of Things (IoT). Furthermore, in term of energy analysis, the results provided in Adnan et al. (2017) positively demonstrate that the AA_β cryptosystem is appropriate for a lightweight encryption on an embedded device, thus reasonable likewise for IoT.

The first discussions of side channel attack on the AA_β cryptosystem emerged in the work in Ghafar and Ariffin (2014). Hypothetically, the result shows that such cryptosystem is susceptible to a timing attack. One of the limitations with this theoretical attacks is that the attack is possible if the attacker able to gather some leaked values of a specific parameter during the decryption process. Another kind of attacks illustrated by Asbullah and Ariffin (2016a) show that the AA_β cryptosystem is breakable if inappropriate keys are chosen, as analogous to the work by Asbullah et al. (2016). The reason was because of the algebraic nature implicitly dwell in the public and secret keys. Thus, they recommend that the parameters chosen during key generation for AA_β cryptosystem should be examined and chosen carefully. Later on, Ghafar and Ariffin (2016) demonstrate a simple power analysis upon the AA_β cryptosystem which show the possibility that the secret keys could be recovered by utilizing

such a strategy. Recently, Asbullah et al. (2019) remark that partial recovery of several most significant bit of the data is possible whenever the transmitted data larger than the specified. Nevertheless, we remark that all of the cryptanalytical results discussed are still impractical to break the AA_β cryptosystem.

Our contribution. In this work, we provide a comparative analysis of the new design of the AA_β cryptosystem, namely the enhanced AA_β cryptosystem (Asbullah et al., 2018) with of any of version of AA_β cryptosystem prior to this work. Note that this work is focussed to compare the efficiency analysis on the decryption procedure only. The results show that the enhanced version of the AA_β cryptosystem is efficient in term of faster running time and lower memory consumption.

This paper has been divided into four sections, begins with a brief overview of the AA_β cryptosystem in Section 1. Section 2 laying out the background and important materials for this research. Section 3 describes the comparative analysis between three version of AA_β cryptosystem. Finally, we conclude the paper in Section 4.

2. Preliminaries

In this section, we describe all the three version of the AA_β cryptosystem in consideration. Next, we also provide a brief introduction to single-precision multiplication and memory cost evaluation technique that we used throughout this work.

2.1 AA_β Cryptosystem - The original Version

We begin with a description of the key generation, encryption and decryption procedure of the original AA_β cryptosystem Ariffin et al. (2013) as follows.

2.2 Fast AA_β variant with Garner's Algorithm

We now describe decryption procedure of the AA_β cryptosystem of Asbullah and Ariffin (2014) version as follows. Remark that the same decryption techniques for this version is analogous to Mahad et al. (2017) .

Remark 2.1. *The key generation and encryption procedure for the fast AA_β variant with Garner's algorithm are the same as the original version of AA_β, i.e. Algorithm 1 and Algorithm 2, respectively.*

Algorithm 1: AA_β Key Generation

Input : The size k-bit of the security parameter
Output: The public key (A_1, A_2) and the private key (d', p, q)

1 Generate at random two distinct primes $p, q \equiv 3 \pmod 4$ such that $2^k < p, q < 2^{k+1}$
2 Set $A_2 = p^2 q$
3 Choose a random integer A_1 such that $2^{3k+4} < A_1 < 2^{3k+6}$ and $\gcd(A_1, A_2) = 1$
4 Compute an integer d' such that $A_1 d' \equiv 1 \pmod{pq}$
5 Return the public key (A_1, A_2) and the private key (d', p, q)

Algorithm 2: AA_β Encryption

Input : The public key A_1 and A_2)
Output: A ciphertext c

1 Choose a plaintext m such that $2^{2k-2} < m < 2^{2k-1}$
2 Choose a plaintext t such that $2^{4k} < t < 2^{4k+1}$
3 Compute $c = A_1 m^2 + A_2 t$
4 Return the ciphertext c

2.3 Enhanced AA_β-cryptosystem

This section introduces a new design of public key encryption scheme namely the Enhanced AA_β-cryptosystem (Asbullah et al., 2018), which incorporates the Rabin-p (Asbullah and Ariffin, 2016b) decryption techniques upon the AA_β decryption procedure.

Remark 2.2. *In reference to the Algorithm 1, we note that the key generation here (i.e. Algorithm 5) is slightly modified from the original version, in term of the generation of the integer d and removal of the prime q in the private keys tuple. Whilst the encryption procedure is exactly the same as the describe by Algorithm 2.*

Algorithm 3: AA_β Decryption

Input : A ciphertext, c, the private key d',p and q
Output: The value m, t as the plaintext

Calculate $w \equiv cd' \pmod{pq}$
Calculate $m_p \equiv M^{\frac{p+1}{4}} \pmod{p}$
Calculate $m_q \equiv M^{\frac{q+1}{4}} \pmod{q}$
Calculate $j_1 \equiv q^{-1} \pmod{p}$
Calculate $j_2 \equiv p^{-1} \pmod{q}$
Calculate $m_1 \equiv m_p j_1 q + m_q j_2 p \pmod{pq}$
Calculate $m_2 \equiv m_p j_1 q - m_q j_2 p \pmod{pq}$
Calculate $m_3 \equiv -m_p j_1 q + m_q j_2 p \pmod{pq}$
Calculate $m_4 \equiv -m_p j_1 q - m_q j_2 p \pmod{pq}$
for $i \leftarrow 1$ **to** 4 **do**
    Compute $t_i = \frac{c - A_1 m_i^2}{A_2}$
end
for $i \leftarrow 1$ **to** 4 **do**
    if (m_i, t_i)=*INTEGER* **then**
        Return the plaintext (m, t)=(m_i, t_i)
    else
        Reject
    end
end

2.4 Single-Precision Multiplication

A single-precision multiplication (*spm*) is referred to as the multiplication of the two integers, which in cryptography, often stated as the binary representation of any digits. We suppose the running time for an addition and a subtraction is operations that can perform very quickly compared to a multiplication or a division. Furthermore, a division is the most complex and expensive amongst basic arithmetic operations.

The running time for an algorithm is possibly measured in numerous ways such as the number of steps, the machine instructions or the clock cycles (Menezes et al., 1997). Running time is important since it can show the performance of an algorithm, and could be used as a method of comparison between algorithms. The 'single-precision multiplication' measurement (i.e. *spm*) is selected in this work to determine the running time. The reason behind the selection of the *spm* measurement because it was widely used to analyzed crypto-

Algorithm 4: AA_β with Garner's Algorithm

Input : A ciphertext, c, the private key d',p and q
Output: The plaintext (m, t)

Calculate $w \equiv cd' \pmod{pq}$
Calculate $m_p \equiv M^{\frac{p+1}{4}} \pmod{p}$
Calculate $m_q \equiv M^{\frac{q+1}{4}} \pmod{q}$
Calculate $j \equiv p^{-1} \pmod{q}$
Calculate $h_1 \equiv (m_q - m_p)j \pmod{q}$
Calculate $h_2 \equiv (-m_q - m_p)j \pmod{q}$
Calculate $m_1 = m_p + h_1 p$
Calculate $m_2 = m_p + h_2 p$
Calculate $m_3 = pq - m_2$
Calculate $m_4 = pq - m_1$
for $i \leftarrow 1$ **to** 4 **do**
    Compute $t_i = \frac{c - A_1 m_i^2}{A_2}$
end
for $i \leftarrow 1$ **to** 4 **do**
    if (m_i, t_i)=*INTEGER* **then**
        Return the plaintext (m, t)=(m_i, t_i)
    else
        Reject
    end
end

Algorithm 5: Enhanced AA_β Key Generation

Input : The security parameter of k-bit size
Output: The public key (A_1, A_2) and the private key (d, p)

Choose two distinct primes at random; $p, q \equiv 3 \pmod{4}$ such that $2^k < p, q < 2^{k+1}$
Calculate $A_2 = p^2 q$
Select an integer A_1 at random satisfy $2^{3k+4} < A_1 < 2^{3k+6}$
Calculate the value $d \equiv {A_1}^{-1} \pmod{p^2}$
Output the value A_1, A_2 as the public keys and d, p as the private key

graphic algorithms. For instance, this method was used by CRYPTREC (i.e. a cryptographic evaluation community established by the Japanese Government) to conduct their analysis on several cryptographic algorithms for government and industrial use. Therefore, the comparative analysis conducted in this work

Algorithm 6: Enhanced AA_β Decryption

Input : A ciphertext c and the private key (d, p)
Output: The plaintext (m, t)

Calculate $w \equiv cd \pmod{p^2}$
Calculate $m_p \equiv M^{\frac{p+1}{4}} \pmod{p}$
Calculate $i = \frac{M - {m_p}^2}{p}$
Calculate $j \equiv \frac{i}{2m_p} \pmod{p}$
Calculate $m_1 = m_p + jp$
if $m_1 < 2^{2k-1}$ **then**
  $m = m_1$
else
  $m = p^2 - m_1$
end
Calculate $t = \frac{c - A_1 m^2}{A_2}$
Output the value m, t as the plaintext

to measure the running time of each scheme will be based on the *spm* method of evaluation. All notations used for this analysis can be retrieved from (Asbullah and Ariffin, 2014).

2.5 Memory Cost Evaluation

We highlight that this work uses the similar method as used in (Asbullah and Ariffin, 2014) for the purpose of evaluating the memory cost consumption. We present the definition of system parameters and accumulators, respectively as follows.

Definition 2.1. *(System parameter). A system parameter is a constant or a variable; which was pre-registered and conceivably was settled preceding any computational action. The system parameter will be stored permanently in memory space. For example, the secret keys that were implanted in the equipment (i.e. hardwired with Integrated Circuit).*

Definition 2.2. *(Accumulator) An accumulator is a constant or a variable value of any current computational activity, and its resultant value is temporarily being stored in the memory. Once all of the required computations are finished, the parameters will be deleted. In other words, the memory space for an accumulator is dynamically being stored and deleted.*

3. Comparative Analysis

This section gives a comparative analysis between the three schemes that was mentioned earlier which are the original AA_β (Ariffin et al., 2013), the AA_β fast version (Asbullah and Ariffin, 2014) and the new enhanced AA_β (Asbullah et al., 2018). We provide a comparative analysis of the decryption procedures only. This is due to that all of these cryptosystems using the same key generation and encryption procedures.

3.1 Enhancement from the Original Version

Table 1 below compares the number and the size of public and private keys of all three version of the AA_β considered in this work. In addition, Table 1 provides the number of mathematical operations needed to complete the decryption procedures, respectively.

Table 1: Comparison between three versions of AA_β Cryptosystem.

	Original AA_β	Fast AA_β	Enhanced AA_β
Public Keys	$\lvert A_1\rvert, \lvert A_2\rvert = 3k$	$\lvert A_1\rvert, \lvert A_2\rvert = 3k$	$\lvert A_1\rvert, \lvert A_2\rvert = 3k$
Private Keys	$\lvert d^{'}\rvert = 2k$, $\lvert p\rvert, \lvert q\rvert = k$	$\lvert d^{'}\rvert = 2k$, $\lvert p\rvert, \lvert q\rvert = k$	$\lvert d\rvert = 2k$, $\lvert p\rvert = k$
Mod Exponent	2	2	1
Mod Inverse	2	1	1
Mod Reduction	5	3	2
Division	4	4	2
Novak's Attack	Yes	Yes	No

As can be seen from the Table 1, the enhanced AA_β version reported only use two values for the private keys (i.e. d and p) while the original AA_β and fast AA_β variant use d', p and additional parameter q. Meaning that the enhanced AA_β have several advantages compared to other AA_β variant in term of smaller size of the decryption key, which apparently affects the overall computational operations.

As shown in Table 1, the mathematical operations needed for the enhanced AA_β to complete the decryption process are minimal in comparison to the original version and the fast variant.

3.2 Running Time Evaluation

Table 2 presents comparison on decryption running time in term of *spm*. Evidently, the Enhanced AA_β runs near to 50% faster than the other version. The contributing factor to this finding because the Enhanced AA_β only uses a single prime number p during decryption.

Table 2: Comparison on Decryption Running Time.

Algorithm	Running Time (*spm*)
Original AA_β	$14k^3 + 413k^2 + 376k + 7$
Fast AA_β	$12k^3 + 274k^2 + 251k + 9$
Enhanced AA_β	$6k^3 + 193k^2 + 175k + 5$

Figure 1 presents the concrete benchmark test performed for all three of the cryptosystems compared in this work. The horizontal axis on the graph shows the security parameter size. We use the concrete running time using the standard security parameter k of 341, 682 and 1365 bits size, respectively. The vertical axis shows the running time taken by each scheme to complete their decryption process, measured in *spm* unit, respectively. The result shows that the amount of the running time for the enhanced AA_β to complete the decryption is faster than the other two schemes. As the security parameter increases, the running time for the enhanced AA_β affected insignificantly. This implies that the increment of the security parameter size will not render the performance of the enhanced AA_β decryption.

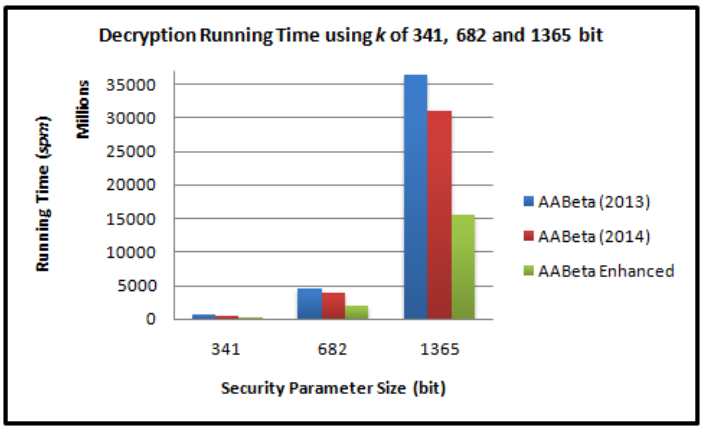

Figure 1: Decryption Running Time using k of 341, 682 and 1365 bits.

3.3 Memory Consumption

Table 3 and Table 4 summarizes the memory consumption of the system parameters and the accumulators for original AA_{β} decryption, respectively.

Table 3: System Parameters Memory for Original AA_{β} Decryption.

Register Name	**Number of Register**	**Bits**
p, q	$2 \times k$	$2k$
pq, d'	$2 \times 2k$	$4k$
A_1, A_2	$2 \times 3k$	$6k$
c	$1 \times 7k$	$7k$
	Total	$19k$

Table 4: Accumulators' Memory for Original AA_{β} Decryption.

Register Name	**Number of Register**	**Bits**
m_p, m_q, j_1, j_2,	$4 \times k$	$4k$
w, m_1, m_2, m_3, m_4	$5 \times 2k$	$10k$
t_i	$1 \times 4k$	$4k$
	Total	$18k$

Table 5 and Table 6 summarizes the memory consumption of the system parameters and the accumulators for fast AA_{β} decryption, respectively.

Table 5: System Parameters Memory for Fast AA_{β} Decryption.

Register Name	**Number of Register**	**Bits**
p, q	$2 \times k$	$2k$
pq, d'	$2 \times 2k$	$4k$
A_1, A_2	$2 \times 3k$	$6k$
c	$1 \times 7k$	$7k$
	Total	$19k$

Table 6: Accumulators' Memory for Fast AA_β Decryption.

Register Name	Number of Register	Bits
m_p, m_q, h_1, h_2, j	$5 \times k$	$5k$
w, m_1, m_2, m_3, m_4	$5 \times 2k$	$10k$
t_i	$1 \times 4k$	$4k$
	Total	$19k$

Table 7 and Table 8 summarizes the memory consumption of the system parameters and the accumulators for enhanced AA_β decryption, respectively.

Table 7: System Parameters Memory for Enhanced AA_β Decryption.

Register Name	Number of Register	Bits
p	$1 \times k$	$1k$
p^2, d	$2 \times 2k$	$4k$
A_1, A_2	$2 \times 3k$	$6k$
c	$1 \times 7k$	$7k$
	Total	$18k$

Table 8: Accumulators' Memory for Enhanced AA_β Decryption.

Register Name	Number of Register	Bits
m_p, i, j	$3 \times k$	$3k$
w, m_1	$2 \times 2k$	$4k$
t	$1 \times 4k$	$4k$
	Total	$11k$

Table 9 compares the amount of memory consumption during decryption stage of the original AA_β, the fast AA_β and the enhanced AA_β.

Table 9: Memory Consumption during Decryption Stage.

	SP	Ac	Total
Original AA_β	$19k$	$18k$	$37k$
Fast AA_β	$19k$	$18k$	$38k$
Enhanced AA_β	$18k$	$11k$	$29k$

4. Conclusion

To summarize, the comparative analysis indicates that the enhanced AA_β cryptosystem is faster running time to completing decryption process compare to the other two versions. Evidently, the enhances version gain advantages since utilizing only a single prime p instead of two primes p and q. In addition, the number of other operations such as modular reduction and division operations also are minimal. Moreover, enhanced AA_β decryption uses less memory for systems parameters and accumulators as compared to the original and fast AA_β. In conclusion, when we consider selecting amongst AA_β-variants cryptosystem, the enhanced version is the best choice.

Acknowledgements

The present research was partially supported by the UPM Grant under Putra Grant-IPM with Project Number GP-IPM/2017/9519200.

References

Adnan, S., Isa, M., and Hashim, H. (2016a). Implementation of the AA_β lightweight asymmetric encryption scheme on an embedded system device. *Advanced Science Letters*, 22(10):2910–2913.

Adnan, S., Isa, M., and Hashim, H. (2016b). Timing analysis of the lightweight AA_β encryption scheme on embedded Linux for Internet of Things. In *ISCAIE 2016 - 2016 IEEE Symposium on Computer Applications and Industrial Electronics*, pages 113–116. IEEE.

Adnan, S., Isa, M., and Hashim, H. (2017). Energy analysis of the AA_β lightweight asymmetric encryption scheme on an embedded device. In *IEACon 2016 - 2016 IEEE Industrial Electronics and Applications Conference*, pages 116–122. IEEE.

Ariffin, M. R. K., Asbullah, M. A., Abu, N. A., and Mahad, Z. (2013). A New Efficient Asymmetric Cryptosystem Based on the Integer Factorization Problem of $N = p^2q$. *Malaysian Journal of Mathematical Sciences*, 7(S):19–37.

Ariffin, M. R. K. and Mahad, Z. (2012). AA_β public key cryptosystem - A comparative analysis against RSA and ECC. In *Proceedings - 2012 7th In-*

ternational Conference on Computing and Convergence Technology (ICCIT, ICEI and ICACT), ICCCT 2012, pages 589–594. Cited By :4.

Asbullah, M., Kamel Ariffin, M., Mahad, Z., and Daud, M. (2019). (In)security of the cryptosystem for transmitting large data. In *8th International Conference on Software and Computer Applications*, volume Part F147956, pages 91–94.

Asbullah, M. A. and Ariffin, M. R. K. (2014). Comparative Analysis of Three Asymmetric Encryption Schemes Based Upon the Intractability of Square Roots Modulo $N = p^2q$. In *The 4th International Cryptology and Information Security Conference 2014 (Cryptology2014)*, pages 86–99.

Asbullah, M. A. and Ariffin, M. R. K. (2016a). Analysis on the AA_β Cryptosystem. In *The 5th International Cryptology and Information Security Conference 2016 (Cryptology2016)*, pages 41–48.

Asbullah, M. A. and Ariffin, M. R. K. (2016b). Design of Rabin-like Cryptosystem without Decryption Failure. *Malaysian Journal of Mathematical Sciences*, 10(S):1–18.

Asbullah, M. A., Ariffin, M. R. K., and Mahad, Z. (2016). Analysis on the Rabin-p cryptosystem. In *The 4th International Conference on Fundamental and Applied Sciences (ICFAS2016)*, pages 080012–1–080012–8. AIP Conf. Proc. 1787.

Asbullah, M. A., Ariffin, M. R. K., and Mahad, Z. (2018). Enhanced AA_β Cryptosystem: The Design. In *Cryptology and Information Security Conference 2018 (Cryptology2018)*, pages 94–102.

Ghafar, A. H. A. and Ariffin, M. R. K. (2014). Timing Attack Analysis on AA_β Cryptosystem. *Journal of Computer and Communication*, 2:1–9.

Ghafar, A. H. A. and Ariffin, M. R. K. (2016). SPA on Rabin variant with public key $N = p^2q$. *Journal of Cryptographic Engineering*, 6(4):339–346.

Mahad, Z. and Ariffin, M. R. K. (2012). AA_β public key cryptosystem - A new practical asymmetric implementation based on the square root problem. In *Proceedings - 2012 7th International Conference on Computing and Convergence Technology (ICCIT, ICEI and ICACT), ICCCT 2012*, pages 584–588. Cited By :1.

Mahad, Z., Asbullah, M. A., and Ariffin, M. R. K. (2017). Efficient methods to overcome Rabin cryptosystem decryption failure. *Malaysian Journal of Mathematical Sciences*, 11(S2):9–20.

Menezes, A., Oorschot, P., and Vanstone, S. (1997). *Handbook Of Applied Cryptography*. CRC Press.

Nishioka, M., Satoh, H., and Sakurai, K. (2002). Design and Analysis of Fast Provably Secure Public-Key Cryptosystems Based on a Modular Squaring. *In Information Security And Cryptology - ICISC 2001*, pages 81–102.

Takagi, T. (1998). Fast RSA-Type Cryptosystem Modulo $p^k q$. *Advances in Cryptology-CRYPTO'98*, 1462:318–326.