On the Insecurity of Generalized (Rivest-Shamir-Adleman) - Advance and Adaptable Cryptosystem

On the Insecurity of Generalized (Rivest-Shamir -Adleman) - Advance and Adaptable Cryptosystem

M.A.M. Isa[1], N.N.A. Rahman[2], M.A. Asbullah[3,4], M.H.A. Sathar[3,4] and A.F.N. Rasedee[5]

[1]IT Exploration Technology, Universiti Teknologi MARA (UiTM), 40450 Shah Alam, Selangor, Malaysia
[2]Pusat GENIUSpintar Negara, Universiti Kebangsaan Malaysia, 43600 UKM Bangi, Selangor, Malaysia
[3]Institute for Mathematical Research, Universiti Putra Malaysia, Serdang, 43400, Malaysia
[4]Centre of Foundation Studies for Agricultural Science, Universiti Putra Malaysia, Serdang, 43400, Malaysia
[5]Faculty of Economics and Muamalat, Universiti Sains Islam Malaysia, Negeri, 78100, Malaysia

E-mail: [1]ma_asyraf@upm.edu.my

Abstract. This paper explores the security claims of the Generalized (Rivest-Shamir - Adleman) - Advance and Adaptable Cryptosystem, in short the GRSA-AA cryptosystem. In the GRSA-AA design proposal, the public key n is defined as the multiplication of two large prime numbers, while the values of encryption key E and decryption key D are relying on the result of multiplying 2^k large prime numbers called N where n divides N. The GRSA-AA claimed that the brute force is necessary to break the cryptosystem even if the integer n was factored. Nevertheless, this paper aims to show that this scheme is insecure once n is factored. The mathematical proof is presented to show that it is easy to generate an alternative value to the private key D without brute-forcing, yet successfully break the system.

1. Introduction

Integer factorization problem is a well-known problem in the field of number theory, computations and also cryptography. In general, the problem is defined as finding all the prime factors of an integer and it is the source of security for many cryptosystems most notably the first public-key encryption namely the RSA cryptosystem [11] and such as [3, 6, 9]. In the RSA cryptosystem, it is believed that to determine the private key is by factoring the integer n. However, due to current knowledge and computational power, it is widely known that factoring n is cannot be done in polynomial time [7]. Furthermore, many cryptanalysis results of the factoring work, for instance, see [2, 4] and [5], which does not contribute to a real-world threat to the RSA cryptosystem. As discussed in [7, 10], such attacks do not reflect the insecurity of any integer factorization-based cryptosystem in general.

The algorithms of the famous RSA cryptosystem which developed by [11] are divided into three components, namely the key generation algorithm, followed by the encryption and decryption algorithm, respectively. The details of the RSA algorithms are outlined as follows:

Algorithm 1 RSA Key Generation Algorithm

1: Generate two prime numbers p_1 and p_2 such that randomly
2: Calculate $n = p_1 p_2$ and $\phi(n) = (p_1 - 1)(p_2 - 1)$
3: Select e satisfies $3 \leq e < \phi(n)$ and $\gcd(e, \phi(n)) = 1$
4: Determine d satisfies $ed \equiv 1 \pmod{\phi(n)}$
5: Output the tuple (n, e) and (n, d) as the public private key, respectively.

Algorithm 2 RSA Encryption Algorithm

Input: The public key (n, e)
Output: A ciphertext c
1: Select integer m where $0 < m < n$ satisfies $\gcd(m, n) = 1$
2: Calculate $c \equiv m^e \pmod{n}$.
3: Output the ciphertext c

Algorithm 3 RSA Decryption Algorithm

Input: A ciphertext c and the private key (n, d)
Output: The plaintext m
1: Calculate $m \equiv c^d \pmod{n}$
2: Output the plaintext m

A new public-key cryptosystem which is called the Generalized (Rivest-Shamir-Adleman) or in short GRSA-AA using 2^k prime numbers were introduced by [8]. The authors of [8] proposed an improved version of the RSA cryptosystem with higher security measure. Suppose the public key n of the GRSA-AA is defined as the same as the original RSA design, i.e. the multiplication of two large prime numbers. However, compared to the original RSA cryptosystem, the public and the private parameters of encryption E and decryption D keys, respectively, are relying on the result of multiplying 2^k large prime numbers called N where n divides N. In that sense, [8] claimed that GRSA-AA is more secure than RSA.

The aim of this paper is to prove that the need for brute force is unnecessary to break the system once n is factored. Moreover, this work proves that it is easy to generate an alternative value to the private key D therefore able to successfully attack the system. As a result, the security level GRSA-AA has deemed no better than the classical RSA.

This paper has been organized in the following way. Firstly, this paper gives a brief introduction of the RSA cryptosystem and overview for the GRSA-AA which was established by [8] in Section 1. Section 2 begins by laying out the details of the GRSA-AA key generation mechanism, along with the procedure of encryption and decryption. Section 3 describes the theorem that will be useful for the cryptanalysis results while in Section 4 the execution of the attack upon the GRSA-AA is presented. The last section concludes the paper.

2. Review of the GRSA-AA Cryptosystem

This section dedicated for the GRSA-AA cryptosystem, as presented in [8]. We summarized it in a simplified version and provides with security analysis as follows.

Algorithm 4 GRSA-AA Key Generation Process for 2^3 prime numbers

1: Choose eight (i.e 2^3) random and distinct primes p_i where $i = 1, 2, ..., 8$
2: Compute $n = p_1 p_2$
3: Compute $N = \prod_{i=1}^{8} p_i$
4: Determine $\phi(n) = (p_1 - 1)(p_2 - 1)$
5: Determine $\phi(N) = \prod_{i=1}^{8}(p_i - 1)$
6: Choose at random e_1 such that $1 \leq e_1 < \phi(n)$ and $\gcd(e_1, \phi(n)) = 1$
7: Choose at random e_2 such that $1 \leq e_2 < \phi(N)$ and $\gcd(e_2, \phi(N)) = 1$
8: Compute E' such that $E' \equiv e_1^{e_2} \pmod{N}$
9: Choose E such that $1 < E < \phi(n) \cdot E'$ and $\gcd(E, \phi(n) \cdot E') = 1$
10: Compute D such that $ED \equiv 1 \pmod{\phi(N) \cdot E'}$
11: Return the public key (n, E) and the private key (n, D)

Note that Algorithm 4 is the algorithm for GRSA-AA to generate its public and private keys of 2^3 prime numbers, while the following Algorithm 5 is the algorithm for GRSA-AA to generate its public and private keys of 2^4 prime numbers (rewritten in this paper as a simplified version of the original GRSA-AA [8]).

Algorithm 5 GRSA-AA Key Generation Process for 2^4 prime numbers

1: Choose sixteen (i.e 2^4) random and distinct primes p_i where $i = 1, 2, ..., 16$
2: Compute $n = p_1 p_2$
3: Compute $N = \prod_{i=1}^{16} p_i$
4: Determine $\phi(n) = (p_1 - 1)(p_2 - 1)$
5: Determine $\phi(N) = \prod_{i=1}^{16}(p_i - 1)$
6: Choose at random e_1 such that $1 \leq e_1 < \phi(n)$ and $\gcd(e_1, \phi(n)) = 1$
7: Choose at random e_2 such that $1 \leq e_2 < \phi(N)$ and $\gcd(e_2, \phi(N)) = 1$
8: Compute E' such that $E' \equiv e_1^{e_2} \pmod{N}$
9: Choose E such that $1 < E < \phi(n) \cdot E'$ and $\gcd(E, \phi(n) \cdot E') = 1$
10: Compute D such that $ED \equiv 1 \pmod{\phi(N) \cdot E'}$
11: Return the public key (n, E) and the private key (n, D)

The next two algorithms which labelled as Algorithm 6 and Algorithm 7 are for the GRSA-AA encryption and decryption algorithm, respectively. Observed that, the procedure is analogous to the original RSA cryptosystem except for the public and parameters are generated using GRSA-AA key generation mechanisms as presented as above.

Algorithm 6 GRSA-AA Encryption Algorithm

1: Select M as an integer satisfies $0 < M < n$ and $\gcd(M, n) = 1$
2: Calculate $c \equiv M^E \pmod{n}$.
3: Output the ciphertext C

Algorithm 7 GRSA-AA Decryption Algorithm

1: Calculate $M \equiv C^D \pmod{n}$
2: Output the plaintext M

In [8], the GRSA-AA was claimed that brute force is required despite whether somebody able to factor the public key n, to get the rest of the set of primes that constructs N. Therefore, the private key D. Hence, in [8] affirmed that the security level considerably strengthened, in contrast to the original design of the RSA cryptosystem. In this work, we tend to argue that the said claim in [8] is often not necessarily true and will justify the argument in the following section.

3. Results and Discussion

This section will now explain the argument such that whenever the modulus n is factored, then the GRSA-AA already insecure and broken (i.e. easily attacked). The argument starts as follows.

Suppose that the prime factors of n is obtained; i.e. its corresponding primes p_1 and p_2. Thus, from here the value $\phi(n) = (p_1 - 1)(p_2 - 1)$ easily computatable. Note that the public key E of GRSA-AA cryptosystem is determine satisfying the $\gcd(E, \phi(N) \cdot E') = 1$ for the generated modulus N as described in Algorithm 4 and Algorithm 5, respectively. Now, the modulus N is a multiple of n, hence $\phi(N)$ is a multiple of $\phi(n)$, therefore the $\gcd(E, \phi(n)) = 1$. Thus, E has a modular inverse in mod $\phi(n)$. Suppose such integer (i.e. modular inverse) is labeled as D^*. Consider the following classical theorem by Euler;

Theorem 1. *(Euler's Theorem). Let $n = p_1 p_2$ and $\phi(n) = (p_1 - 1)(p_2 - 1)$. For every integer M such that such that $0 < M < n$ and $\gcd(M, n) = 1$, then $M^{\phi(n)} \equiv 1 \pmod{n}$.*

Hence Euler's Theorem is an important tool to justify the argument, which written as the following proposition.

Proposition 1. *Let (n, E) denoted as the public tuple and (n, D) as its private key counterparts of the GRSA-AA cryptosystem. For any message M such that $0 < M < n$, and for $D^* \neq D$ such that $ED^* \equiv 1 \pmod{\phi(n)}$, then $M \equiv C^{D^*} \pmod{n}$.*

Proof. Consider $D^* \neq D$ such that $ED^* \equiv 1 \pmod{\phi(n)}$. Since $D^* \neq D$ is the multiplicative inverse of $E \pmod{\phi(n)}$, thus $ED^* \equiv 1 \pmod{\phi(n)}$ can be rewritten as $ED^* = 1 + \phi(n)j$ for some integer j. Now, for any message M such that $0 < M < n$, Theorem 1 confirmed that $C^{D^*} \equiv M^{ED^*} \equiv M^{1+\phi(n)j} \equiv M^1 \cdot M^{\phi(n)j} \equiv M \pmod{n}$. □

Hence, given only the parameter of n, E and its ciphertext C, the above result shows that any message $M < n$ easily recovered by using the newly introduce private key D^*.

Note that the private key D generated from Algorithm 4 (or Algorithm 5) indicate that it has at least as the same size as of the modulus N. Surprisingly, the Proposition 1 shows that it is not necessarily to find the exact private key D, yet it is sufficient only to have the prime factors of the modulus n, contradicts to the original claim of [8]. Hence, the GRSA-AA cryptosystem is broken once the prime factors of n are available, without using brute force. In the next subsections, the successful attacks on the examples given in [8] using Proposition 1 will be presented.

3.1. Attack 1: on the illustration of the GRSA-AA with 2^3 (eight) prime numbers

The following is the illustration of the GRSA-AA with eight prime numbers, which replicated directly from the Example 2.5 given in [8]. Take eight prime numbers and proceed the key generation procedure of Algorithm 4 and obtaining the following parameters;

Public keys	$n = 10403, E = 239$
Private modulus	$N = 31705684556450851$
Private key	$D = 183935154785333957559167984061 59$
Message	$M = 786$
Ciphertext	$C = 9614$

Table 1. Relevance parameters displayed in ([8], Section 2.5)

Successful Attack 1: Now, the following steps will illustrate the attack using Proposition 1 to obtain M easily without the need to obtain all the prime factors of $N = 31705684556450851$ nor the exact value of decryption key $D = 183935154785333957559167984061 59$. Assume that the factorization of the modulus $n = 10403$ are given, i.e. obtain its corresponding prime factors $p1 = 101, p2 = 103$, respectively. Next, compute $\phi(n) = 10200$. Since $E = 239$ is publicly available therefore it is easy to determine the private value $D^* = 6359 \neq D$ where $D^* \equiv E^{-1} \pmod{\phi(n)}$. Finally, compute $C^{D^*} \equiv 9614^{6359} \equiv 786 \equiv M \pmod{n}$, which is the intended message M.

3.2. Attack 2: on the illustration of the GRSA-AA with 2^4 *(sixteen) prime numbers*
The following is the Example 2.6 (attack on the illustration of the GRSA-AA with sixteen prime numbers) given in [8]. Take sixteen prime numbers and proceed the key generation procedure of Algorithm 5 and obtaining the following parameters;

Public keys	$n = 129593387513, E = 769$
Private modulus	$N = 42624624772085204961203849243873728696103577690$ $708177642812230936409520655643273544114569$
Private key	$D = 14004465899537074532492622505905541002426419424 6$ $645270734820063515907116459162019549611244132869$ $701459598144704223767446645079612794540802545128 54$ $880472624562132900791910144332 9$
Message	$M = 786786$
Ciphertext	$C = 115334483704$

Table 2. Relevance parameters displayed in ([8], Section 2.6)

Successful Attack 2: Now, the following steps will illustrate the second attack using Proposition 1 to obtain M easily without the need to obtain all the prime factors of N nor the exact value of decryption key D. Suppose the factorization of the modulus $n = 129593387513$ i.e. its corresponding prime factors $p_1 = 317159, p_2 = 408607$ are given. Next, compute $\phi(n) = 129592661748$. Using the public key $E = 769$ hence compute $D^* = 28311530785 \neq D$ where $D^* \equiv E^{-1} \pmod{\phi(n)}$. Finally, compute $C^{D^*} \equiv 115334483704^{28311530785} \equiv 786786 \equiv M \pmod{n}$, which is the intended message M.

4. Conclusion
This paper has argued that the brute force in unnecessary to break the GRSA-AA cryptosystem once the public key n is factored. Moreover, this work proves that it is easy to generate an alternative value to the private key D, namely the attacking private value D^*, therefore able to successfully attack the system. In conclusion, the security level GRSA-AA has deemed no better than the classical RSA.

Acknowledgments
The present research was partially supported by the Putra Grant with Project Number GP/2017/9552200.

References

[1] Abubakar S I, Ariffin M R K and Asbullah M A 2018 A new simultaneous diophantine attack upon RSA moduli $N = pq$ *Proceedings of the 6th International Cryptology and Information Security Conference 2018, CRYPTOLOGY 2018* pp 119–138.
[2] Asbullah M A and Ariffin M R K 2019 Another Proof Of Wiener's Short Secret Exponent. *Malaysian Journal of Science*, **1** 62
[3] Asbullah M A, Ariffin M R K and Mahad Z 2018 Enhanced AA_β cryptosystem: The design *Proceedings of the 6th International Cryptology and Information Security Conference 2018, CRYPTOLOGY 2018* pp 94–102.
[4] Asbullah M A, Ariffin M R K and Mahad Z 2016 Analysis on the Rabin-p cryptosystem *AIP Conference Proceedings* **1787** 080012.
[5] Asbullah M A and Ariffin M R K 2016 Analysis on the AA_β cryptosystem *Proceedings of the 5th International Cryptology and Information Security Conference 2016, CRYPTOLOGY 2016* pp 41–48.
[6] Asbullah M A and Ariffin M R K 2016 Design of Rabin-like cryptosystem decryption failure *Malaysian Journal of Mathematical Sciences* **10(S)** pp 1–18.
[7] Ghafar A H A, Ariffin M R K and Asbullah M A 2018 Extending pollard class of factorable RSA modulus *Proceedings of the 6th International Cryptology and Information Security Conference 2018, CRYPTOLOGY 2018* pp 103–118.
[8] Lone A H and Khalique A 2016 Generalized RSA Using 2^k Prime Numbers with Secure Key Generation. *Security and Communication Networks* **9(17)** pp.4443–4450.
[9] 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)** pp 9–20.
[10] Rahman N N A, Ariffin M R K, Asbullah M A and Yunos, F 2018 New vulnerability on system of $Ni = p_i^2 q_i$ using good approximation of $\phi(N)$. *Proceedings of the 6th International Cryptology and Information Security Conference 2018, CRYPTOLOGY 2018* pp 139–150.
[11] Rivest R L, Shamir A and Adleman L 1978 A Method for Obtaining Digital Signatures and Public-Key Cryptosystems *Communications of the ACM* **21(2)** pp.120–126.