International Journal of Cryptology Research 5(2): 1- 14 (2015)

Provably Secure Randomized AA_β Cryptosystem

Muhammad Asyraf Asbullah[1] and **Muhammad Rezal Kamel Ariffin**[2]

[1,2]*Al-Kindi Cryptography Research Laboratory, Institute for Mathematical Research, Universiti Putra Malaysia*
[2]*Mathematics Department, Faculty of Science, Universiti Putra Malaysia*

E-mail: ma_asyraf@upm.edu.my[1], *rezal@upm.edu.my*[2]

ABSTRACT

The security of a modern public key cryptosystem is usually viewed from their security goal and attack models, with the intention to come out with a provably secure cryptosystem. In this paper, we propose a randomized encryption setting algorithm based on the AA_β cryptosystem. We also present provable security elements for the randomized AA_β cryptosystem with emphasis given to the standard security against strongest attack model, namely the chosen-ciphertext attack. This randomized AA_β cryptosystem is projected in the random oracle model.

Keywords: AA_β cryptosystem, provable security, chosen ciphertext attack, random oracle model.

1 INTRODUCTION

Basically, a public key cryptosystem is not practical for sending long messages, but rather is frequently used to transmit a short and temporary key for the use of

symmetric cryptosystem (Abe et al., 2008). Nevertheless, some cryptographic protocols such as Secure Electronic Transaction (SET) do not only encrypt the secret symmetric key but also attaches other information together. For instance, information on user's identification or user account authentication (Hitachi, 2002). As a result, Nishioka et al. (2002) took this as a motivation to construct a new encryption scheme known as the HIME(R) cryptosystem.

As taken in the report in Hitachi (2002), the plaintext space of the HIME(R) scheme is large enough to carry the secret encryption key with the attached data. Henceforth, it is significant to design a public key encryption scheme that manifests such purposes. So, with the same spirit, a new Rabin-like encryption scheme known as the AA_{β} cryptosystem was designed that can actually transmit the information in a single encrypted data set, which its security based on the hardness for factoring $N = p^2q$ (Ariffin et al., 2013). This cryptosystem has proven to have the quality for encrypting and securing large size data (Asbullah and Ariffin, 2014), however there is no provable security proof is given.

In proposing a public key cryptosystem, it is conventional to claim that the public key cryptosystem has the strongest security by showing that it is secure in the sense of secure against chosen-ciphertext attack (CCA). For instance see Bellare and Rogaway (1995) and Cramer and Shoup (2003). It is largely agreed upon that security against CCA is one of the most important attributes of any public key cryptosystem (Müller, 2001). Hence, formalizing and proving a public key cryptosystem is secure under CCA is of very important. Therefore, throughout this paper, we will give the desired provable security proof upon the AA_{β} cryptosystem. We consider the security goal under CCA in the random oracle setting.

The paper is organized as follows. Section 2 reviews the AA_{β} cryptosystem and provide several definitions related to the study. In section 3, we describe our proposed randomized AA_{β} scheme and accompany it with the security proof in section 4. The conclusion appears in the final section.

2 PRELIMINARIES

In this section, we begin with a description of the AA_β cryptosystem, which is proposed earlier by Ariffin et al. (2013). Then, we provide all the necessary definitions that particularly used in this paper. We then further with the definition of chosen ciphertext attack (CCA).

2.1 AA_β Cryptosystem

In this section, we review the AA_β cryptosystem. We begin by describing the key generation, encryption and decryption procedure of AA_β cryptosystem as follows.

Algorithm 1 AA_β Key Generation Algorithm

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

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

Algorithm 2 AA_β Encryption Algorithm

Input: The plaintext m, t and the public key A_1, A_2
Output: A ciphertext c

1: Choose a plaintext $2^{2k-2} < m < 2^{2k-1}$ such that $\gcd(m, A_2) = 1$
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

Algorithm 3 AA_β Decryption Algorithm

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

1: Compute $w \equiv cd \pmod{A_2}$
2: Compute $m_p \equiv w^{\frac{p+1}{4}} \pmod{p}$
3: Compute $m_q \equiv w^{\frac{q+1}{4}} \pmod{q}$
4: Compute $j \equiv p^{-1} \pmod{q}$
5: Compute $h_1 \equiv (m_q - m_p)j \pmod{q}$
6: Compute $h_2 \equiv (-m_q - m_p)j \pmod{q}$
7: Compute $m_1 = m_p + h_1 p$
8: Compute $m_2 = m_p + h_2 p$
9: Compute $m_3 = pq - m_2$
10: Compute $m_4 = pq - m_1$
11: Compute $t_i = \frac{c - A_1 m_i^2}{A_2}$ for $m_i < 2^{2k-1}$ for $i = 1, 2, 3, 4$
12: Sort the pair (m_i, t_i) for integer t_i, else reject
13: Return the plaintext m, t

2.2 Useful Definitions

Definition 2.1. *(Cryptographic Hard Problem). A cryptographic hard problem is defined as a function that can be computed easily, however very hard to find the inverse.*

Definition 2.2. *(Negligible Function). For every polynomial $f(\cdot)$, if there exists an $N > 0$ such that for all integers $n > N$ it holds that $\epsilon(n) < \frac{1}{f(n)}$, then such $\epsilon(n)$ is called as a negligible function.*

Definition 2.3. *(AA_β Function). Let k be the security parameter. Choose two random and distinct primes $p, q \equiv 3 \pmod{4}$ where $2^k < p, q < 2^{k+1}$. Let $A_2 = p^2 q$ and choose a random $A_1 \in \mathbb{Z}^+_{(2^{3k+4}, 2^{3k+6})}$ such that $\gcd(A_1, A_2) = 1$. Let $m^2 \in \mathbb{Z}^+_{(2^{2k-2}, 2^{2k-1})}$ and $t \in \mathbb{Z}^+_{(2^{4k}, 2^{4k+1})}$. Suppose (A_1, A_2) be the public parameters and (m, t) be private parameters. Let*

$$c = A_1 m^2 + A_2 t \tag{1}$$

We define (1) as the AA_β function.

Definition 2.4. *(AA_β Function Hard Problem). Let the AA_β Function Hard Problem is defined as the problem to find (m, t) from c such that $c = A_1m^2 + A_2t$ where the parameter m, t, A_1, A_2 are as described by Definition 2.3. Suppose $[\mathcal{A}_{(AA_\beta)} = 1]$ is an event of a probabilistic polynomial time algorithm $\mathcal{A}$ that given (A_1, A_2) and $c = A_1m^2 + A_2t$ is successfully obtained (m, t), otherwise $[\mathcal{A}_{(AA_\beta)} = 0]$. We say that the AA_β Function Hard Problem is hard, if for all probabilistic polynomial time algorithm $\mathcal{A}$ there exists a negligible function ϵ such that*

$$\Pr[\mathcal{A}_{(AA_\beta)} = 1] \leq \epsilon$$

Definition 2.5. *(Random Oracle Model). A random oracle is a function $H(\cdot) : \{0,1\}^n \longrightarrow \{0,1\}^n$ that maps an input value to a true random output value.*

2.3 Chosen Ciphertext Attack

Throughout this paper, we consider the notions of security against the chosen ciphertext attack as our security goal.

Definition 2.6. *(CCA Game). Suppose Π be a public key encryption scheme and Λ be a polynomial time adversary that attempts to break such scheme. We describe the CCA via the following game played between Π and Λ.*

1. *Π generate (e, d) and e is given to the adversary Λ.*
2. *Λ is allowed to make decryption queries for any ciphertext of his choice to Π, which decrypts the given ciphertexts.*
3. *Λ will output two messages m_0 and m_1 and send these messages to Π. During this stage, Π chooses $r \in \{0,1\}$ at random and sets $c = E(e, m_r)$ as the challenge ciphertext.*
4. *Λ is allowed to continue the decryption queries for any ciphertext $c^* \neq c$ to Π.*
5. *Λ outputs the value of r' and wins the game if $r' = r$.*

The goal of the adversary Λ is to answer whether the challenge ciphertext c is the encryption of m_0 or m_1. The scheme which denoted by Π is said to be adaptive chosen ciphertext secure if for all efficient polynomial time adversaries Λ, the probability of such adversary are winning the CCA game is not greater than $\frac{1}{2}$, which means to show that the adversary Λ cannot acquire any useful information about a message from its ciphertext. We now formally defined the CCA-secure for public key encryption scheme as the following definition.

Definition 2.7. *Let CCA^{Λ}_{Π} be an event of a public key encryption scheme Π and a polynomial time adversary Λ playing the CCA game as described in Definition 2.6. Let $[CCA^{\Lambda}_{\Pi} = 1]$ is the event for Λ winning the CCA game over Π, otherwise $[CCA^{\Lambda}_{\Pi} = 0]$. Then a public key encryption Π is considered has the CCA-security if for all probabilistic polynomial-time adversaries Λ there exists a negligible function ϵ such that*

$$\Pr[CCA^{\Lambda}_{\Pi} = 1] \leq \frac{1}{2} + \epsilon$$

3 RANDOMIZED AA_{β} CRYPTOSYSTEM

We retain the same procedure for the AA_{β} key generation as described in Algorithm 1 and output the public keys A_1, A_2 and the private keys d, p, q.

Remark 3.1. *Suppose we have a random oracle $G(\cdot) : |m^2| \mapsto |v|$ such that $m \in \mathbb{Z}^{+}_{(2^{2k-2}, 2^{2k-1})}$ and $v \in \mathbb{Z}^{+}_{(2^{4k}, 2^{4k+1})}$. We then start the encryption as follows.*

Remark 3.2. *In the randomized AA_{β} encryption setting, the random integer m is not part of the plaintext (or any intended information) but instead it only acts as an ephemeral value.*

Algorithm 4 Randomized AA_β Encryption Algorithm

Input: The value m^2, t and the public keys A_1, A_2
Output: A ciphertext c

1: Choose a random integer $m \in \mathbb{Z}^+_{(2^{2k-2},2^{2k-1})}$
2: Compute m^2
3: Query $u = G(m^2)$ from the random oracle
4: Choose a plaintext $v \in \mathbb{Z}^+_{(2^{4k},2^{4k+1})}$
5: Compute $t = v \oplus u$
6: Compute $c = A_1 m^2 + A_2 t$
7: Return the ciphertext c

Remark 3.3. *After receiving the ciphertext c, we first decrypt c as mentioned in Algorithm 3 in order to obtain the integer m, with additional operation as follows.*

Algorithm 5 Randomized AA_β Decryption Algorithm

Input: A ciphertext c and the private key d, p
Output: The message v

1: Decrypt c as Algorithm 3, and obtain m and t
2: Compute m^2
3: Query $u = G(m^2)$ from the random oracle
4: Compute $v = t \oplus u$
5: Return the message v

4 SECURITY PROOF FOR THE RANDOMIZED AA_β CRYPTOSYSTEM

Theorem 4.1. *The randomized AA_β cryptosystem is secure against CCA, if the AA_β function is hard and $G(\cdot)$ is modeled as a random oracle.*

Proof sketch. Suppose $G(\cdot)$ is modeled as a random oracle such that $G(m^2) : |m^2| \mapsto |v|$. Then we will follow the argument that to distinguish

between the case that the adversary does not query m^2 to the random oracle $G(\cdot)$ and the case of when it does. Note that the construction of the randomized AA_β cryptosystem does not involve any symmetric key cryptosystem.

Observe that, in the first case, the adversary learns nothing about the key m^2. Hence we can say that the success probability of such adversary to break the randomized AA_β cryptosystem is no better than to the success probability of a random guessing, which is exactly $\frac{1}{2}$. For the latter case, the probability that the adversary is able to query m^2 to the random function $G(\cdot)$ is negligible if the AA_β function is hard. Once again, in order to prove Theorem 4.1, we need to show simulation (i.e a reduction algorithm) for the query to the decryption oracle and the query to the random oracle $G(\cdot)$, which made by the adversary.

Proof. (Full proof) Let the randomized AA_β cryptosystem be described as in Section 3 denoted as Υ. Let Λ be a probabilistic polynomial-time adversary that succeeds in breaking such Υ with non-negligible probability. First of all, we should describe the scenario of the actual CCA game that's being played between Λ and the Υ as follows.

1. Υ publish the public keys A_1 and A_2 to Λ.
2. Λ has given the ability to make queries to the random function $G(\cdot)$ for certain values, and to make decryption queries for any ciphertext of his choice to Υ.
3. Λ output two messages v_0 and v_1 such that $v_0, v_1 \in \mathbb{Z}^+_{(2^{4k}, 2^{4k+1})}$ and send it to Υ.
4. Υ choose the integer $\mathring{m} \in (2^{2n-2}, 2^{2n-1})$ and choose $r \in \{0, 1\}$ at random
5. Υ query for the value $\mathring{u} = G(\mathring{m}^2)$ and compute $\mathring{t} = v_r \oplus G(\mathring{m}^2)$
6. Sets $\mathring{c} = A_1\mathring{m}^2 + A_2\mathring{t}$ as the challenge ciphertext.
7. Λ is still allowed to make queries to the random function $G(\cdot)$ and to ask for decryption of any ciphertext, except for the challenge ciphertext $\mathring{c}$ itself.

8. Λ output a value of r' and wins the game if $r' = r$.

Let $Q_{(\mathring{m}^2)}$ denote the event that Λ successfully queries the correct $\mathring{m}^2$ to the random oracle $G(\cdot)$ occurs. We also use Win as a notation of the event Λ correctly outputs $r' = r$. Then,

$$\begin{aligned} \Pr[Win] &= \Pr[Win \wedge \overline{Q_{(\mathring{m}^2)}}] + \Pr[Win \wedge Q_{(\mathring{m}^2)}] \\ &\leq \Pr[Win \wedge \overline{Q_{(\mathring{m}^2)}}] + \Pr[Q_{(\mathring{m}^2)}] \end{aligned}$$

where all probabilities are taken over the randomness used in the CCA game. □

We are now proceed to show that it is indeed $\Pr[Win \wedge \overline{Q_{(\mathring{m}^2)}}] \leq \frac{1}{2}$ and there exists a negligible function ϵ such that $\Pr[Q_{(\mathring{m}^2)}] \leq \epsilon$. Consider the following lemma.

Lemma 4.1. *If the AA_β function is hard and $G(\cdot)$ is modeled as random oracle, then $\Pr[Win \wedge \overline{Q_{(\mathring{m}^2)}}] \leq \frac{1}{2}$ and $\Pr[Q_{(\mathring{m}^2)}] \leq \epsilon$.*

Proof. We first proof the first assertion. Suppose $\Pr[Win \wedge \overline{Q_{(\mathring{m}^2)}}]$ means that the adversary correctly outputs $r' = r$ without making any query to the random oracle $G(\cdot)$. Therefore, in the first statement, the adversary learns nothing about the value of $\mathring{m}^2$. Then we can reduce the success probability is similar to the event of random guessing, hence

$$\Pr[Win \wedge \overline{Q_{(\mathring{m}^2)}}] \leq \frac{1}{2}$$

Now, we further to prove the second assertion, as follows. Suppose we are given a AA_β function hard problem as follows. Suppose we are provided with $\mathring{c}$ suct that $\mathring{c} = A_1\mathring{m}^2 + A_2\mathring{t}$ and are given the known parameters A_1, A_2. The task is to determine the exact integer $\mathring{m}^2$ and $\mathring{t}$. If $\Pr[Q_{(\mathring{m}^2)}]$ is not negligible, then we can manipulate the adversary Λ as a subroutine in order to answer the given a AA_β function hard problem.

Observe the following; suppose Λ is given the public keys A_1, A_2 and a ciphertext c from Υ during the execution of the CCA game. Then we watch

over all the queries made to the random oracle $G(\cdot)$ by Λ. Suppose $[\mathcal{A}_{(AA_\beta)} = 1]$ is an event as in Definition 2.4. If the event $Q_{(\mathring{m}^2)}$ is taking place, then one of the queries $\mathring{m}^2$ to the random oracle satisfies $\mathring{c}$, therefore $\Pr[Q_{(\mathring{m}^2)}]$ has the same probability as $\Pr[\mathcal{A}_{(AA_\beta)} = 1]$. Hence, by definition we should have $\Pr[Q_{(\mathring{m}^2)}] \leq \epsilon$.

Formally, we need to set a reduction algorithm, denoted as Δ'_R that solve the AA_β function hard problem. This reduction algorithm Δ'_R should have the ability to respond to both; the **decryption queries** and the **random oracle queries**, as the same as Υ, however without the prior knowledge of the decryption key d and p. Once again, the usefulness of the ROM methodology allowed for such response to be made.

Suppose the reduction algorithm Δ'_R is given A_1, A_2 and $\mathring{c}$ as the input. The objective is for Δ'_R to output the value of $\mathring{m}^2$ and $\mathring{t}$. Initially Δ'_R will prepare a table that contains a tuple of $(\cdot, \cdot, \cdot, \cdot, \cdot)$ with each '$\cdot$' indicates that its corresponding c, m^2, u, t and v, respectively. Note that the value of m^2 and v are unknown at the moment, so does for $u = G(m^2)$.

Remark 4.1. *When Λ make **a decryption query** for a ciphertext c, then Δ'_R will look up through the table and search if there is a tuple that contains c as its initial entry.*

1. *If the value c is found from the table (in the form of (c, m^2, u, t, v)), then Δ'_R will return the decryption result v to Λ.*
2. *Otherwise, if the value c is nowhere to be found in the table, then Δ'_R will store c, generate u and t at random. It then compute $m^2 = \frac{c - A_2 t}{A_1}$ and set $v = t \oplus u$. Thus, the table is refreshed with new entry (c, m^2, u, t, v). Hence Δ'_R will return v as the result of the decryption query to Λ.*

Remark 4.2. *When Λ make **a random oracle query** for a value m^2, then Δ'_R will search through all the entries in the table and search if there is a tuple that contains m^2.*

1. *If there is an entry that contain m^2 (i.e. in the form (c, m^2, u, t, v)), then Δ'_R will return the value u to Λ*

2. *If there is an entry that contain m^2 (i.e. in the form $(\cdot, m^2, u, \cdot, \cdot)$), then Δ'_R will return the value u to Λ*

3. *If the query value m^2 is nowhere be found in the table, then Δ'_R will generate u at random, then Δ'_R store the query value m^2 and the newly generated u into the table as the tuple $(\cdot, m^2, u, \cdot, \cdot)$ and return u in response*

Let the reduction algorithm Δ'_R be given A_1, A_2 and $\mathring{c}$ as inputs, of which the objective is to output its corresponding $\mathring{m}$ and $\mathring{t}$. Hence, we begin the simulation of the CCA game between the reduction algorithm Δ'_R and the adversary Λ as follows.

Simulation of the CCA game between Δ'_R and Λ

1. Δ'_R is given A_1, A_2 and $\mathring{c}$ as an instance from AA_β function hard problem

2. Δ'_R send the A_1, A_2 as its public keys to Λ

3. Λ make a decryption query for any of its chosen value c

   At this point, Δ'_R must be able to answer any queries of Λ on the value of c when is needed. Suppose Λ submit its decryption query on the value of c, then Δ'_R will respond to the value v as follows.

   i. If such c is from an entry (c, m^2, u, t, v), then return the decryption result v to Λ

   ii. Otherwise, Δ'_R will store the query value c and then generate u and t at random. It then compute $m^2 = \frac{c - A_2 t}{A_1}$ and set $v = t \oplus u$. Thus, Δ'_R will return v as the result of the decryption query to Λ store all the newly entries into the table

4. Λ make a random oracle query for any of its chosen value m^2

   At this point, Δ'_R must be able to answer any queries of Λ on the value of m^2 when is needed. Suppose Λ submit its query on the value of m^2 to the random oracle, then Δ'_R will respond with the value u as follows.

   i. If there exists an entry m^2 of the form (c, m^2, u, t, v) in the table, then return u to Λ

ii. If there exists an entry m^2 of the form $(\cdot, m^2, u, \cdot, \cdot)$ in the table, then return u to Λ

iii. Otherwise, if the query value m^2 is nowhere be found in the table, then Δ'_R will generate u at random, then Δ'_R store the query value m^2 and the newly generated u into the table as the tuple $(\cdot, m^2, u, \cdot, \cdot)$ and return u in response

5. At some point, Λ output two messages v_0 and v_1
6. Δ'_R choose $r \in \{0, 1\}$ at random
7. Δ'_R set $\mathring{c}$ as the challenge ciphertext

   At this point, Δ'_R send $\mathring{c}$ as the challenge ciphertext to Λ.
8. Δ'_R continues to answer the decryption queries and the random oracle queries from Λ as before
9. Suppose Λ query for $\mathring{m}^2$ the random oracle

   At this point Δ'_R compute $t = \frac{\mathring{c} - A_1 \mathring{m}^2}{A_2}$

   i. If $t = \frac{\mathring{c} - A_1 \mathring{m}^2}{A_2} \in \mathbb{Z}$, then set $\mathring{t} = t$. Return $\mathring{u} = \mathring{t} \oplus v_r$ to Λ

   ii. Otherwise, return to Step 4.
10. At the end of the CCA game, Λ will output it guesses r'
11. Δ'_R output $\mathring{m}^2$ and $\mathring{t}$.

The CCA game ends upon Δ'_R output $\mathring{m}$ and $\mathring{t}$. It is immediate that the reduction is done by Δ'_R runs in polynomial time. We notice that every given input during the CCA game played between Δ'_R and Λ is distributed uniformly as in the actual CCA game played between Υ and Λ. Thus, the view of Λ of either in the actual CCA game with Υ or being used as subroutine by the Δ'_R in this reduction is indistinguishable.

Let $Q_{(\mathring{m}^2)}$ denote the event that Λ successfully queries the correct $\mathring{m}^2$ to the random oracle occurs. Let $[\mathcal{A}_{(AA_\beta)} = 1]$ as define in the Definition 2.4. We say that the reduction algorithm Δ'_R solve the instance $\mathring{c}$ of the AA_β function

hard problem whenever the event $Q_{(\mathring{m}^2)}$ happens. Since the $\Pr[\mathcal{A}_{(AA_\beta)} = 1]$ is negligible, then we obtain

$$\Pr[Q_{(\mathring{m}^2)}] = \Pr[\mathcal{A}_{(AA_\beta)} = 1] \leq \epsilon$$

The proof is now complete. □

5 CONCLUSION

In this work, we design an efficient and provably secure cryptosystem which is proven to resilient to the stronger adversarial model, namely the chosen ciphertext attack. The design is done by randomized setting implemented to the original AA_β cryptosystem from its deterministic form that is proposed earlier in Ariffin et al. (2013). This randomized AA_β cryptosystem is also shown to be secure against chosen ciphertext attack and is projected in the random oracle model.

REFERENCES

Abe, M., Gennaro, R., and Kurosawa, K. (2008). Tag-KEM/DEM: A New Framework For Hybrid Encryption. *Journal Of Cryptology*, 21(1):97–130.

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.

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 Proceeding of the 4th International Cryptology and Information Security Conference 2014*, pages 86–99.

Bellare, M. and Rogaway, P. (1995). Optimal Asymmetric Encryption. In *Advances In Cryptology-EUROCRYPT'94*, pages 92–111. Springer.

Cramer, R. and Shoup, V. (2003). Design And Analysis Of Practical Public-Key Encryption Schemes Secure Against Adaptive Chosen Ciphertext Attack. *SIAM Journal On Computing*, 33(1):167–226.

Hitachi. HIME(R) Public-Key Cryptosystem. *http://www.hitachi.com/rd/yrl/crypto/hime/*.

Müller, S. (2001). On The Security Of Williams Based Public Key Encryption Scheme. In *Public Key Cryptography*, pages 1–18. Springer.

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. Springer.