Rabin-p Key Encapsulation Mechanism
A Proposal for Public Key Encryption
for CyberSecurity Malaysia MySEAL Initiative

January 2019

COVER SHEET

1. Name of proposed algorithm : Rabin-p Key Encapsulation Mechanism
2. Principal submitter : Muhammad Asyraf Asbullah
3. Email : ma_asyraf@upm.edu.my
4. Organization : Institute for Mathematical Research, Universiti Putra Malaysia (UPM)
5. Postal Address : Institute for Mathematical Research (INSPEM), Universiti Putra Malaysia, 43400 UPM, Serdang, Selangor, MALAYSIA
6. Inventor : Muhammad Asyraf Asbullah, Muhammad Rezal Kamel Ariffin, Zahari Mahad
7. Owner : Universiti Putra Malaysia

Signature:

Muhammad Asyraf Asbullah

Contents

List of Tables

Abbreviations

CCA	Chosen Ciphertext Attack
CCA1	Non-adaptive CCA
CCA2	Adaptive CCA
CPA	Chosen Plaintext Attack
CRT	Chinese Remainder Theorem
gcd	Greatest Common Divisor
IFP	Integer Factorization Problem
IND	Indistinguishability
IND-CCA2	Indistinguishable against CCA2
LLL	Lenstra-Lenstra-Lovasz
OAEP	Optimal Asymmetric Encryption Padding
ROM	Random Oracle Model
RSA	Rivest-Shamir-Adleman

Abstract

The modular square root problem has a special property of the having computational equivalent to a well-known hard mathematical problem namely integer factorization problem. The proposed Rabin-p Key Encapsulation Mechanism is built upon the said problem as its source of security, aiming for efficient and practical modular square root-based cryptosystem of which accompanied with the following properties;

1. improves the performance without plaintext padding mechanisms or sending extra bits during encryption and decryption processes
2. the plaintext is uniquely decrypted without decryption failure
3. improve decryption efficiency by using only one modular exponentiation
4. a decryption key using only a single prime number
5. sufficiently large plaintext space
6. appropriate plaintext-ciphertext expansion ratio
7. implementable on software and hardware with ease
8. achieves IND-CPA security.

Acknowledgement

Universiti Putra Malaysia (UPM) would like to acknowledge the following sub-research group members for their contributions.

1. **Design, Cryptanalysis and Provable Security Personnel**

   Muhammad Asyraf Asbullah[1], Muhammad Rezal Kamel Ariffin[1]

2. **External dedicated cryptanalyst**

   Abderrahmane Nitaj [2]

3. **Implementation Personnel**

   Zahari Mahad[1]

[1]Universiti Putra Malaysia, Selangor, MALAYSIA
[2]Université de Caen, Normandy, FRANCE

MySEAL Requirements

This section will link the contents of this document with Section 7.1.B and Annex G as per requirements within MySEAL Submission and Evaluation Criteria document v1.0. It is intended to give evidence that this document has complied with all specifications and checklist needed for MySEAL evaluation process.

No hidden weaknesses in the algorithm design

We certify that, to the best of our knowledge, we have fully disclosed there are no hidden weaknesses in our algorithms.

Section 7.1.B

- Complete and unambiguous description of the basic algorithm in the most suitable form is shown in Chapter 2.
- Complete and unambiguous description of the Rabin-p Key Encapsulation Mechanism in the most suitable form is shown in Chapter 5.
- Method for key generation and parameter selection is shown in Chapter 4 [Section 4.1].

- Statement that there are no hidden weaknesses is as the mentioned above.
- Statement of the claim on security properties is discussed in Chapter 3.
- Expected security level is shown in Table 4.1.
- An analysis of the algorithm with respect to standard cryptanalytic attacks is discussed throughout Chapter 3.
- Statement giving the strengths and limitations of the algorithm is described in Chapter 4.
- Design rationale explaining design choices is described in Chapter 1 [Section 1.2].
- Statement of the estimated computational efficiency in software is mentioned in Chapter 6.
- Description of the basic techniques for implementers to avoid implementation weaknesses is mentioned in Chapter 7.

Annex G

- Hard mathematical problems and assumptions are described in Chapter 2, 3 and 5.
- Security model and its proof are explained throughout Chapter 5.

Chapter 1

Introduction

1.1 Background

The Rabin encryption scheme [30] is one of an existing workable asymmetric cryptosystem that comes with nice cryptographic properties. For instance, it has low-cost encryption of which the Rabin encryption is relatively fast to encrypt compared to the widely commercialized RSA cryptosystem [32], and it has been proven to be as difficult as the integer factorization problem. On the other hand, the decryption of Rabin's scheme produces four possible answers, which only one is correct. This four-to-one decryption setting of the Rabin decryption could lead to a decryption failure scenario since no indicator for selecting the correct plaintext is given.

Theoretically speaking, it is such a waste to abandon a cryptosystem that possesses nice features such as the Rabin cryptosystem. Hence attempts were made by numerous researchers with the objective to turn the Rabin cryptosystem to be as practical and implementable as the RSA cryptosystem. Broadly speaking, all the previous attempts made seem to employ one or

more additional features in order to obtain a unique decryption result, but at the same time may have a small probability for decryption failure. One of the ways to accomplish this is through manipulation of some mathematical objects such as the role of the Jacobi symbol or the Dedekind's sums theorem. Also, it can be done by designing an encryption function with a special message structure. Yet, at the same time all the designs lose the computational advantage of the original Rabin's encryption over the RSA cryptosystem.

In order to engage this problem and to overcome all the previous drawbacks of Rabin's original design and its variants, we propose the Rabin-p Key Encapsulation Mechanism, provided with theoretical analysis, perfomance measurement and robust implementation. We revisit the Rabin cryptosystem and then aspire to furnish a new design aiming for efficient, secure and practical Rabin-like cryptosystem. In our design, we use the modulus $N = p^2q$ and we restrict the plaintext to be less than p^2. Hence, to decrypt correctly, it suffices to apply an efficient algorithm that solves the square root of quadratic congruence modulo p instead of modulo $N = p^2q$.

1.2 Design Rationale

In designing the Rabin-p Key Encapsulation Mechanism, the following are the main criteria that were taken into consideration:

1. improves the performance without plaintext padding mechanisms or sending extra bits during encryption and decryption processes
2. the plaintext is uniquely decrypted without decryption failure
3. improve decryption efficiency by using only one modular exponentiation

4. a decryption key using only a single prime number
5. sufficiently large plaintext space
6. appropriate plaintext-ciphertext expansion ratio
7. implementable on software and hardware with ease
8. achieves IND-CPA security.

The design principle to overcome the drawbacks of the original Rabin cryptosystem and all its variants are outlined as follows. Firstly, we put the condition on the modulus to be used is of the type $N = p^2q$. We note that such modulus $N = p^2q$ is claimed to be no easier than factoring the conventional modulus of $N = pq$ [5]. We then impose restriction on the plaintext m and ciphertext c space as $m \in \mathbb{Z}_{p^2}$ and $c \in \mathbb{Z}_{p^2q}$, respectively. From the plaintext-ciphertext expansion, such restriction leads to a system that is not a length-preserving for the message.

Let m and c be the plaintext and ciphertext and $c(m)$ be the function of c taking m as its input. Say, for instance, the plaintext spaces and the ciphertext spaces in the RSA cryptosystem are the same. Thus we denote the mapping for the RSA cryptosystem as $c(m) : \mathbb{Z}_{pq} \longrightarrow \mathbb{Z}_{pq}$. Note that this situation could be an advantage for the RSA scheme since RSA encryption has no message expansion. This is, however, not true for all cryptosystems.

The size of a message m is determined by the size of its plaintext space. Suppose we put a restriction on the size of such m. If the intended plaintext m is merely the secret key needed for the use of a symmetric cryptosystem, then such key is indeed a short message. For example, the plaintext-ciphertext mapping for Okamoto-Uchiyama cryptosystem [26] is $c(m) : \mathbb{Z}_p \longrightarrow \mathbb{Z}_{p^2}$,

Pailier cryptosystem [28] and the cryptosystem proposed by [11] is $c(m) : \mathbb{Z}_{pq} \longrightarrow \mathbb{Z}_{(pq)^2}$, Rabin-Boneh [3] mapping is $c(m) : \mathbb{Z}_{\frac{pq}{2}} \longrightarrow \mathbb{Z}_{pq}$ and the Rabin variant introduced by [34] is $c(m) : \mathbb{Z}_{pq} \longrightarrow \mathbb{Z}_{p^2q}$.

Therefore, we note that the issue of losing the ability to encrypt a relatively longer m is insignificant. Hence, we reason that, even imposing restrictions on the plaintext space or to set a prefix message size would not be a hindrance for designing a considerable efficient cryptosystem.

Chapter 2

Rabin-p Cryptosystem: The Design

In this chapter, we provide the details of the proposed cryptosystem namely Rabin-p Cryptosystem. Rabin-p is named after the Rabin cryptosystem with the additional p symbolizing that the proposed scheme only uses a single prime p as the decryption key. This section is structured as follows. We first describe the Rabin-p key generation, encryption and decryption procedures. We then provide the explanation of the Rabin-p decryption process.

2.1 System Parameters

The key generation algorithm of the Rabin-p cryptosystem (Algorithm 1) produces two random and distinct primes p and q of the same length such that $p \equiv 3 \pmod 4$ and $q \equiv 3 \pmod 4$.

2.2 Rabin-p Key Generation Algorithm

The key generation algorithm then produces an integer N as a product $N = p^2q$, which is denoted as the public key. The private key is the prime p.

Algorithm 1 Rabin-p Key Generation Algorithm

Input: The size k of the security parameter

Output: The public key $N = p^2q$ and the private key 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 $N = p^2q$

3: Return the public key N and the private key p

2.3 Rabin-p Encryption Algorithm

To encrypt a plaintext, the Rabin-p encryption algorithm with the public key N does the following.

Algorithm 2 Rabin-p Encryption Algorithm

Input: The public key N

Output: A ciphertext c

1: Choose plaintext $0 < m < 2^{2k-1}$ such that $\gcd(m, N) = 1$

2: Compute $c \equiv m^2 \pmod N$

3: Return the ciphertext c

Remark 1. *The encryption algorithm (Algorithm 2) takes the plaintext* $m < 2^{2k-1}$ *and compute* $c \equiv m^2 \pmod N$. *We observe that the plaintext* m *is restricted to the range of* $m < 2^{2k-1} = \frac{2^{2k}}{2} < \frac{p^2}{2} < p^2$. *The output is the ciphertext* c.

2.4 Rabin-p Decryption Algorithm

To decrypt a ciphertext, the Rabin-p decryption algorithm with the private key p does the following.

Algorithm 3 Rabin-p Decryption Algorithm

Input: A ciphertext c and the private key p

Output: The plaintext m

1: Compute $w \equiv c \pmod{p}$

2: Compute $m_p \equiv w^{\frac{p+1}{4}} \pmod{p}$

3: Compute $i = \frac{c - {m_p}^2}{p}$

4: Compute $j \equiv \frac{i}{2m_p} \pmod{p}$

5: Compute $m_1 = m_p + jp$

6: If $m_1 < 2^{2k-1}$, then return $m = m_1$. Else, return $m = p^2 - m_1$

Remark 2. *We observe that the decryption algorithm needs only a single prime number as its key. Hence, only one modular exponentiation is taking place during the decryption process. Such computational advantage would positively affect the overall operations.*

Remark 3. *We reason that since our proposed scheme does not need to carry out any CRT computation, thus the Novak's attack is not applicable on the Rabin-p cryptosystem (i.e. resilient against Novak's attack).*

2.5 Proof of Correctness for Rabin-p Decryption

This section explain why the Rabin-p decryption procedure works.

Lemma 1. *[19]. Let p be a prime number such that $p \equiv 3 \pmod 4$ and c an integer such that $\gcd(c,p) = 1$. The congruence $c \equiv m^2 \pmod p$ has either no solutions or exactly two solutions. If m_1 is a solution, then $-m_1 \pmod p$ is the other solution.*

Lemma 2. *[19]. Let p be a prime number such that $p \equiv 3 \pmod 4$ and c an integer such that $\gcd(c,p) = 1$. The congruence $c \equiv m^2 \pmod{p^2}$ has exactly two solutions if $c \equiv m^2 \pmod p$ has exactly two solutions.*

Proof. Suppose that $c \equiv m^2 \pmod{p^2}$ has a solution m_1. Then any other solution m is such that $p^2 | m^2 - m_1^2$, that is $p^2 | (m+m_1)(m-m_1)$. Hence, we consider $p | (m+m_1)(m-m_1)$ as well. If $p | (m+m_1)$ and $p | (m-m_1)$, then p would divide $(m+m_1)+(m-m_1) = 2m$ and $(m+m_1)-(m-m_1) = 2m_1$. Since $p \equiv 3 \pmod 4$ is an odd prime, then $p \nmid 2$ so p would divide both m and m_1. Consider $c \equiv m^2 \pmod{p^2}$. If $p|m$ then $p|m^2$ therefore $p|c$, however $\gcd(c,p) = 1$ therefore $p \nmid c$. Hence $p \nmid m$. The same goes for $p \nmid m_1$. Now, consider in the case if $p | (m+m_1)$ or $p | (m-m_1)$ but not both. Since $p^2 | (m+m_1)(m-m_1)$, therefore either $p^2 | (m+m_1)$ or $p^2 | (m-m_1)$ implies that $m \equiv \pm m_1 \pmod{p^2}$. Since $-m_1 \pmod{p^2}$ is always a solution when $m_1 \pmod{p^2}$ is a solution, therefore the congruence has exactly two solutions. □

Lemma 3. *Consider Lemma 2. Let $c \equiv m^2 \pmod{p^2}$. Then $m_1 = m_p + jp$ is a solution to $c \equiv m^2 \pmod{p^2}$ where $m_p \equiv c^{\frac{p+1}{4}} \pmod p$, $j \equiv \frac{i}{2m_p} \pmod p$ such that $i = \frac{c - {m_p}^2}{p}$. Furthermore $m_2 \equiv -m_1 \pmod{p^2}$ is the other solution.*

Proof. Suppose we consider $c \equiv m^2 \pmod{p^2}$ as in Lemma 2. Let $m_p \equiv c^{\frac{p+1}{4}} \pmod p$ such that ${m_p}^2 \equiv c \pmod p$. Suppose that $m_1 = m_p + jp$ is a solution

for $c \equiv m^2 \pmod{p^2}$, then we have

$$\begin{aligned} c &\equiv m_1^2 \\ &\equiv (m_p + jp)^2 \\ &\equiv {m_p}^2 + 2m_p jp \pmod{p^2} \end{aligned} \tag{2.1}$$

Then, rearrange (2.1) as

$$2m_p jp \equiv c - {m_p}^2 \pmod{p^2} \tag{2.2}$$

Note that from ${m_p}^2 \equiv c \pmod{p}$, we have $c - {m_p}^2 \equiv 0 \pmod{p}$ which means that $c - {m_p}^2$ is a multiple of p. Let $ip = c - {m_p}^2$ for some integer i, then we obtain $i = \frac{c - {m_p}^2}{p}$. We then rewrite (2.2) as

$$2m_p jp \equiv ip \pmod{p^2},$$

of which such congruence implies that $2m_p j \equiv i \pmod{p}$. Hence, we obtain $j \equiv \frac{i}{2m_p} \pmod{p}$. To conclude, we have the solution $m_1 = m_p + jp$ such that $c \equiv {m_1}^2 \pmod{p^2}$. Furthermore, we observe that $m_2 \equiv -m_1 \pmod{p^2}$ is the other solution as in Lemma 2. □

Lemma 4. *Consider Lemma 3. If m_1 and m_2 are the two distinct integers solution for $c \equiv m^2 \pmod{p^2}$, then $m_1 + m_2 = p^2$.*

Proof. Suppose $m_1 \not\equiv m_2 \pmod{p^2}$ such that ${m_1}^2 \equiv {m_2}^2 \equiv c \pmod{p^2}$. Observe that, from Lemma 3 if m_1 is a solution for $c \equiv m^2 \pmod{p^2}$, then $m_2 \equiv -m_1 \pmod{p^2}$ is also a solution. Thus, $m_2 \equiv -m_1 \pmod{p^2}$ can be reinterpreted as $m_2 = p^2 - m_1$. Hence $m_1 + m_2 = p^2$. □

Lemma 5. *Let m_1 and m_2 be integers such that $m_1 + m_2 = p^2$ with p^2 is an odd integer. Then either m_1 or m_2 is less than $\frac{p^2}{2}$.*

Proof. Suppose p^2 is an odd integer, then by definition $\frac{p^2}{2}$ must not be an integer. Let $m_1 + m_2 = p^2$. Since that m_1 and m_2 are integers, therefore m_1 and m_2 must not be equal to $\frac{p^2}{2}$. Suppose we consider the following cases. If for both m_1 and m_2 are less than $\frac{p^2}{2}$, then we should have $m_1 + m_2 < p^2$. Therefore this case contradicts with the fact that $m_1 + m_2 = p^2$. On the other hand, if m_1 and m_2 are greater than $\frac{p^2}{2}$, then we should have $m_1 + m_2 > p^2$, which also contradicts with the fact that $m_1 + m_2 = p^2$. Hence, we consider the case where either m_1 or m_2 is less than $\frac{p^2}{2}$. Let $m_1 < \frac{p^2}{2}$, then there exists a real number ϵ_1 such that $m_1 + \epsilon_1 = \frac{p^2}{2}$. On the other hand, since $m_1 < \frac{p^2}{2}$, then m_2 must be greater than $\frac{p^2}{2}$. Therefore there exists a real number ϵ_2 such that $m_2 - \epsilon_2 = \frac{p^2}{2}$. If we add up these equations, we have

$$(m_1 + \epsilon_1) + (m_2 - \epsilon_2) = \frac{p^2}{2} + \frac{p^2}{2} = p^2$$

Since $m_1 + m_2 = p^2$, thus $\epsilon_1 - \epsilon_2$ should be equal to zero, meaning that $\epsilon_1 = \epsilon_2$. We conclude that only one of m_1 or m_2 is less than $\frac{p^2}{2}$. □

Theorem 1. *Let $c \equiv m^2 \pmod N$ be the Rabin-p ciphertext. Then Algorithm 3 is* ***correct****.*

Proof. Suppose $c \equiv m^2 \pmod N$ be the Rabin-p ciphertext where $N = p^2q$, thus we have $c - m^2 \equiv 0 \pmod N$. Since $p^2 \mid N$, then $p^2 \mid c - m^2$. Algorithm 2 show that $m < p^2$, therefore it is sufficient just solving for $c \equiv m^2 \pmod{p^2}$ which is efficiently solved using Lemma 3. In addition, according to Lemma 2, there are exactly two distinct solution m_1 and m_2 satisfies $c \equiv m^2 \pmod{p^2}$. From Lemma 4 we have $m_1 + m_2 = p^2$. We now show that the Algorithm 3 only produce a unique solution for $m < 2^{2k-1}$. Observe that the upper bound for $m < \frac{p^2}{2}$. Consider Lemma 5, then we have either m_1 or m_2 is less than $\frac{p^2}{2}$ such that $m_1 + m_2 = p^2$ satisfy $m < 2^{2k-1}$. Finally, we conclude that

only one of m_1 or m_2 are less than $\frac{p^2}{2}$ and will be outputted by Algorithm 3 as the unique $m < 2^{2k-1}$. ■ □

2.6 Toy Example

Suppose we have two communicating parties, namely Bob as the sender of a message and Alice as its corresponding receiver. Let the security parameter $k = 15$.

Key generation:

Alice generate two distinct primes $p = 32779, q = 40829$.

1. Compute $N = p^2q = 43869243335189$
2. Alice keeps her private key p
3. Alice publish her public key N

Encryption:

Bob receives Alice's public key. He would like to send a message $m = 479571937$.

1. Compute $c \equiv m^2 \pmod{N} = 26669194871231$
2. Bob send c to Alice as his ciphertext.

Decryption:

Alice receives a ciphertext $c = 26669194871231$ from Bob. To decrypt c, Alice then executes:

1. Compute $w \equiv c \pmod{p} = 27646$
2. Compute $m_p \equiv c^{\frac{p+1}{4}} \pmod{p} = 15167$
3. Compute $i = \frac{c-m_p^2}{p} = 24318$
4. Compute $j \equiv \frac{i}{2m_p} \pmod{p} = 14630$.
5. Compute $m_1 = m_p + jp = 479571937$.
6. Since the integer $m_1 < 2^{2k-1}$, then return the plaintext $m = m_1$.

Chapter 3

Rabin-p Cryptosystem: The Analysis

This chapter discusses the hard problem that becomes the source of security for the Rabin-p cryptosystem. In the following sections, we show that the problem of solving the Rabin-p ciphertext is reduced to factoring $N = p^2q$. Hence, in conclusion, it proves that breaking the Rabin-p cryptosystem is indeed equivalent to factoring $N = p^2q$. We then extend our security analysis by discussing some possible cryptanalysis, for instance; the continued fraction's attack, the Coppersmith's theorems and the Novak's attack.

3.1 Reduction to Factoring $N = p^2q$

In this section, we show that if there exists an algorithm that can decrypt message m from any random Rabin-p ciphertext, then such algorithm also be able to factor $N = p^2q$. We observe the following.

Theorem 2. *Let* $N = p^2q$, $m < 2^{2k-1}$ *and* $2^{2k-1} < \hat{m} < p^2$ *such that* $m + \hat{m} = p^2$. *Then* $\gcd(m + \hat{m}, N) = p^2$.

Proof. Suppose $2^k < p < 2^{k+1}$, then $2^{2k} < p^2 < 2^{2k+2}$, and $2^{2k-1} < \frac{p^2}{2} < 2^{2k+1}$. Suppose $m < 2^{2k-1}$, then from Lemma 5 there exists another integer $\hat{m} > 2^{2k-1}$ such that $m + \hat{m} = p^2$. Thus this implies $p^2 - \hat{m} = m < 2^{2k-1}$. Now, we determine the range of the $\hat{m}$ such that $p^2 - \hat{m} < 2^{2k-1}$. Then we obtain the lower bound for $\hat{m}$, of which

$$\begin{aligned} \hat{m} &> p^2 - 2^{2k-1} \\ &> 2^{2k} - 2^{2k-1} \\ &> 2^{2k-1} \end{aligned}$$

and upper bounded by $\hat{m} < p^2$. Take the $\gcd(m + \hat{m}, N)$, then we obtain p^2. Hence $q = \frac{N}{p^2}$. □

Remark 4. *Theorem 2 implies that if there exists someone or an algorithm that can decrypt the message m from the Rabin-p's ciphertext, then that someone must also be able to factor $N = p^2q$.*

3.1.1 Algorithm for Factoring $N = p^2q$

Note that the Algorithm 3 will output only the integer $m < 2^{2k-1}$. Hence, if we generate an integer $\hat{m}$ such that $2^{2k-1} < \hat{m} < 2^{2k}$, then we can build a factoring algorithm for N, according to Theorem 2 and the Algorithm 3. The factoring algorithm is defined as follows.

Algorithm 4 Algorithm for Factoring $N = p^2q$

Input: A ciphertext c and the modulus N

Output: The prime factors p,q

1: Choose an integer $2^{2k-1} < \hat{m} < 2^{2k}$

2: Compute $\hat{c} \equiv \hat{m}^2 \pmod{N}$

3: Ask the decryption of $\hat{c}$ from Algorithm 3

4: Algorithm 3 output $m < 2^{2k-1}$, else reject

5: Compute $\gcd(\hat{m} + m, N)$

6: If $\gcd(\hat{m} + m, N) = 1$, then reject

7: If $\gcd(\hat{m} + m, N) \neq 1$, then return p^2

8: Compute $\frac{N}{p^2} = q$

9: Return the prime factors p, q

3.1.2 A Toy Example

Let the security parameter $k = 15$. Consider the Algorithm 4 and the key generation from Example 2.6. **Factoring $N = p^2q$ according to Theorem 2:**

1. Generate an integer $\hat{m} = 786696491$ such that $2^{2k-1} < \hat{m} < 2^{2k}$
2. Compute $\hat{c} \equiv \hat{m}^2 \pmod{N} = 27953222201858$
3. Ask the decryption of ciphertext $\hat{c}$ from Algorithm 3
4. Receive the output $m = 287766350$
5. Compute $\gcd(\hat{m} + m, N) = 1074462841$
6. Compute $\frac{N}{1074462841} = 40829$
7. Return the prime factors $p = \sqrt{1074462841} = 32779$ and $q = 40829$

3.2 Computational Equivalent

If a new cryptosystem is designed, we are expected to provide a comparison of the relative difficulty of breaking the scheme into the solving any existing hard problems. Now, we show that breaking the Rabin-p cryptosystem is indeed reducible to factoring the modulus $N = p^2q$. Furthermore, the converse of such statement is also true.

Lemma 6. *Breaking the Rabin-p cryptosystem is reducible to factoring* $N = p^2q$.

Proof. Suppose there exists an algorithm $\mathcal{A}_1$ with the ability to factor the modulus $N = p^2q$, then we obtain the primes p and q. Thus, we can solve the Rabin-p's ciphertext $c \equiv m^2 \pmod{N}$ directly by using the Algorithm 3. □

Lemma 7. *Factoring* $N = p^2q$ *is reducible to breaking the Rabin-p cryptosystem.*

Proof. Conversely, suppose there exists an algorithm $\mathcal{A}_2$ that breaks the Rabin-p cryptosystem. Then such algorithm is able to find the message m from the ciphertext $c \equiv m^2 \pmod{N}$. By using the same approach as Theorem 2, hence $\mathcal{A}_2$ can proceed to compute $\hat{m}$. Finally, with the help of Algorithm 4, $\mathcal{A}_2$ can easily factor the modulus $N = p^2q$. □

Theorem 3. *Breaking the Rabin-p cryptosystem is equivalent to factoring the modulus* $N = p^2q$.

Proof. This assertion is a consequence from Lemma 6 and Lemma 7. □

3.3 Analysis via Continued Fraction's Method

We begin with the definition of the continued fractions, which serves as a very useful mathematical tool and has been applied in many cryptanalytic works.

Definition 1 (Continued Fractions)**.** *[14]. The continued fraction of a real number $R \in \mathbb{R}$ is an expression of the form*

$$R = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cdots}}} \tag{3.1}$$

where $a_0 \in \mathbb{Z}$ and $a_i \in \mathbb{N} - \{0\}$ for $i \geq 1$. The numbers $a_0, a_1, a_2, \ldots$ are called the partial quotients. The equation (3.1) can be denoted as $R = [a_0, a_1, a_2, \ldots]$ and are called the convergents of the continued fraction expansion of R. If R is a rational number then the continued fraction expansion of R is finite.

Following this definition is an important theorem of the continued fraction which be used widely throughout this proposal. This theorem simply says, the unknown integers x and y can be recovered from the list of continued fraction expansion of a rational number R satisfying the given inequality.

Theorem 4 (Legendre's Theorem)**.** *[14] Let R is a rational number. Let $x, y \in \mathbb{Z}, y \neq 0$ and $\gcd(x, y) = 1$. Suppose*

$$\left| R - \frac{x}{y} \right| < \frac{1}{2y^2}$$

Then $\frac{x}{y}$ is a convergent of the continued fraction expansion of R.

We outline the analysis by continued fraction's method as follows. Suppose c and N are the parameters from the Rabin-p cryptosystem. Since we have the ciphertext $c \pmod N$, thus $c < N$. Therefore c can be written as $c = a + bpq$ or $c = a' + b'p^2$ for some integer a, a', b, b'.

Theorem 5. *Let $c = a + bpq$ for some positive integer a and b. If $a < \frac{q}{2}$ and $b < p$, then $\frac{b}{p}$ is a convergent of the continued fraction expansion of $\frac{c}{N}$.*

Proof. Consider the value $c = a + bpq$ and $N = p^2q$. If we divide $c = a + bpq$ by N, then we obtain

$$\begin{aligned} \frac{c}{N} &= \frac{a + bpq}{N} \\ &= \frac{a}{N} + \frac{bpq}{N} \\ &= \frac{a}{N} + \frac{b}{p} \end{aligned} \tag{3.2}$$

Rearranging (3.2),

$$\frac{c}{N} - \frac{b}{p} = \frac{a}{N}$$

Thus, to show that

$$\left| \frac{c}{N} - \frac{b}{p} \right| < \frac{1}{2p^2}$$

it suffices if $\frac{a}{N} < \frac{1}{2p^2}$, which implies that $a < \frac{N}{2p^2} = \frac{q}{2}$. Hence, by Theorem 4, $\frac{b}{p}$ is a convergent of the continued fraction expansion of $\frac{c}{N}$ if $a < \frac{q}{2}$. This leads to finding p and then q. □

Theorem 6. *Let $c = a' + b'p^2$ for some positive integer a' and b'. If $a' < \frac{p^2}{2q}$ and $b' < q$, then $\frac{b'}{q}$ is a convergent of the continued fraction expansion of $\frac{c}{N}$.*

Proof. Consider the value $c = a' + b'p^2$ and $N = p^2q$. If we divide $c = a' + b'p^2$ by N, then we obtain

$$\begin{aligned} \frac{c}{N} &= \frac{a' + b'p^2}{N} \\ &= \frac{a'}{N} + \frac{b'p^2}{N} \\ &= \frac{a'}{N} + \frac{b'}{q} \end{aligned} \tag{3.3}$$

Rearranging (3.3),

$$\frac{c}{N} - \frac{b'}{q} = \frac{a'}{N}$$

Thus, to show that

$$\left|\frac{c}{N} - \frac{b'}{q}\right| < \frac{1}{2q^2}$$

it suffices if $\frac{a'}{N} < \frac{1}{2q^2}$, which implies that $a' < \frac{N}{2q^2} = \frac{p^2}{2q}$. Hence, by Theorem 4, $\frac{b'}{q}$ is a convergent of the continued fraction expansion of $\frac{c}{N}$ if $a' < \frac{p^2}{2q}$. This leads to finding q and then p. □

3.4 Analysis via Coppersmith's Method

[6] invented a significantly powerful method for finding small roots of modular polynomial equations. This method has found many different applications in the area of cryptography and vastly useful tool for cryptanalysis [10]. We now reproduce Coppersmith's theorems for the benefit of the reader.

Theorem 7. *[6] Let N be an integer of unknown factorization. Let $f_N(x)$ be a univariate, a monic polynomial of degree δ. Then we can find all solutions x_0 for the equation $f_N(x) \equiv 0 \pmod{N}$ with $|x_0| < N^{1/\delta}$ in polynomial time.*

Theorem 8. *[20] Let N be an integer of unknown factorization, which has a divisor $b > N^{\beta}$. Furthermore, let $f_b(x)$ be a univariate, a monic polynomial of degree δ.Then we can find all solutions x_0 for the equation $f_b(x) \equiv 0 \pmod{b}$ with $|x_0| < \frac{1}{2}N^{\beta^2/\delta}$ in polynomial time.*

We now analyze the Rabin-p cryptosystem based on the Theorem 7 and Theorem 8 and obtain the following results. Suppose c, m and N are the parameters from the Rabin-p cryptosystem.

Theorem 9. *Let $c \equiv m^2 \pmod{N}$ and $N = p^2q$. If $m < 2^{3k/2}$ then m can be found in polynomial time.*

Proof. Suppose $c \equiv m^2 \pmod{N}$ and $N = p^2q$. Consider the univariate, monic polynomial $f_N(x) \equiv x^2 - c \equiv 0 \pmod{N}$. By applying Theorem 7 we set $\delta = 2$. Hence the root $x_0 = m$ can be recovered if $m < N^{1/\delta} = N^{1/2} \approx 2^{3k/2}$. □

Theorem 10. *Let $c \equiv m^2 \pmod{p^2}$ such that p^2 is an unknown factor for N. If $m < 2^{2k/3}$ then m can be found in polynomial time.*

Proof. Suppose $c \equiv m^2 \pmod{p^2}$ such that p^2 is an unknown factor for N. Consider $f_{p^2}(x) \equiv x^2 - c \equiv 0 \pmod{p^2}$ with $p^2 \approx N^{2/3} \approx 2^{2k}$. We can find a solution $x_0 = m$ if $m < \frac{1}{2}N^{\beta^2/\delta} < N^{(2/3)^2/2} = N^{2/9} \approx 2^{2k/3}$. □

Remark 5. *Therefore in order to avoid both attacks, we would set $m > 2^{3k/2}$ in the Rabin-p encryption algorithm.*

3.5 Resistant to Novak's Attack

In general, the decryption algorithm of a Rabin-like cryptosystem consists of two parts. The first part is for the modular exponentiation operation of which in order to obtain the message in the form of m modulo p and m modulo q from its corresponding ciphertext c. The second part then would be the recombination process using the Chinese Remainder Theorem (CRT) algorithm to recover the proper message m. Most side channel attacks deal with the first part. For instance, the work by [18], [33] and [4] which uses the timing attack approach or the result in [22] enables side channel attack using the power analysis approach.

Alternatively, [25] proposed a very efficient side channel attack upon the CRT computation (i.e. the second part of the Rabin-like decryption). We observe that all variants of the Rabin-like cryptosystem (except Rabin-Williams scheme) involves a process that hardly depends on the CRT or Garner's algorithm (i.e. the process to recover all the modulo square roots). Therefore, Novak's attack is indeed applicable for such computation, of which can result in the insecurity of the cryptosystems [27].

Remark 6. *We reason that since our proposed scheme does not need to carry out any CRT computation, thus the Novak's attack is not applicable on the Rabin-p cryptosystem (i.e. resilient against Novak's attack).*

3.6 Resistant to Chosen Ciphertext Attack

Notice that the factoring algorithm mentioned by the Algorithm 4 could provide a way to launch a chosen ciphertext attack upon the proposed scheme in polynomial time, hence resulting in the system totally insecure in this sense. Therefore, to provide security against this kind of attack, we could consider implementing as a Key Encapsulation Mechanism (KEM) following the KEM framework for Rabin cryptosystem as proposed in [9]. We will discuss this issue further in details in Chapter 5.

Chapter 4

Comparative Analysis

This chapter gives comparison of the basic scheme of Rabin-p cryptosystem and other existing implementable, standardized public key encryption (basic) schemes that are based on the intractibility of the integer factorization problem; namely the HIME(R), Rabin-SAEP+ and RSA-OAEP.

4.1 Security Level and Key Lengths

For the primes p, q of the Rabin-p cryptosystem should be chosen to be intractable to factor the modulus of $N = p^2q$. We choose the NIST Recommendations (2016) [12] for factoring modulus which present the appropriate key length for user's desired level of protection, as follows. Note that for good protection against quantum computers, the modulus size of 15360-bit is sufficent, unless Shor's algorithm applies [12].

Date	Security Level	Modulus Size (bits)	Prime Size (bits)
2016 - 2030 (& beyond)	128	3072	1024
2016 - 2030 (& beyond)	192	7680	2560
2016 - 2030 (& beyond)	256	15360	5120

Table 4.1: Recommendation modulus length for Rabin-p cryptosystem

We suppose that the bit-length k of the modulus $N = p^2q$ for Rabin-p and HIME(R) and the bit-length K of the modulus $N = pq$ for Rabin-SAEP+ and RSA-OAEP have been selected so that the security level of these moduli against integer factorization attacks is the same. The bit-length of the prime factors of a Rabin-p or HIME(R) k-bits modulus is denoted by t (so $t = \frac{k}{3}$), while the bit-length of the prime factors of an RSA-OAEP or Rabin-SAEP+ K-bits modulus is denoted by T (so $T = \frac{K}{2}$). Hence we have the comparative tables as follows.

Algorithm	Modulus length	Public key	Private key
Rabin-p	$N = p^2q$	N	p
HIME(R)[15]	$N = p^2q$	N	p, q
Rabin-SAEP+[35]	$N = PQ$	N	P, Q
RSA-OAEP[2]	$N = PQ$	N, e	P, Q, d_P, d_Q

Table 4.2: Key bit length vs HIME(R), Rabin-SAEP+ and RSA-OAEP

Algorithm	Modulus length	Public key	Private key
Rabin-p	3072	3072	1024
	7680	7680	2560
	15360	15360	5120
HIME(R)[15]	3072	3072	2048
	7680	7680	5120
	15360	15360	10240
Rabin-SAEP+[35]	3072	3072	3072
	7680	7680	7680
	15360	15360	15360
RSA-OAEP[2]	3072	3072∼6144	6144
	7680	7680∼15360	15360
	15360	15360∼30720	30720

Table 4.3: Modulus, Public key(s) and Private key(s) of Rabin-p, HIME(R), Rabin-SAEP+ and RSA-OAEP

4.2 Performance Efficiency

In this section, we compare the speed of Rabin-p when compared to HIME(R), Rabin-SAEP+ and RSA-OAEP through its most fundamental complexity order (i.e. basic textbook operation speed without any enhancement). As a note, any enhancement for the benchmark algorithms will result also in Rabin-p cryptosystem using the enhanced operation mechanism.

4.2.1 Encryption

The computational steps that dominate the execution time of the encryption process for the Rabin-p, HIME(R), Rabin-SAEP+ and RSA-OAEP are:

1. Rabin-p: $m^2 \pmod N$. That is, a modular squaring operation with a k-bit modulus.

2. HIME(R): $m^2 \pmod N$. That is, a modular squaring operation with a k-bit modulus.

3. Rabin-SAEP+: $m^2 \pmod N$. That is, a modular squaring operation with a K-bit modulus.

4. RSA-OAEP: $m^e \pmod N$. That is, a modular exponentitation operation with a K-bit modulus.

4.2.2 Decryption

The computational steps that dominate the execution time of the decryption process for the Rabin-p, HIME(R), Rabin-SAEP+ and RSA-OAEP are:

1. Rabin-p: $c^{\frac{p+1}{4}} \pmod p$. That is, one modular exponentiations with t-bit modulus.

2. HIME(R): $c^{\frac{p+1}{4}} \pmod p$ and $c^{\frac{q+1}{4}} \pmod q$. That is, two modular exponentiations with t-bit moduli.

3. Rabin-SAEP+:$c^{\frac{P+1}{4}} \pmod P$ and $c^{\frac{Q+1}{4}} \pmod Q$. That is, two modular exponentiations with T-bit moduli.

4. RSA-OAEP: $c^{d_P} \pmod P$ and $c^{d_Q} \pmod Q$. That is, two modular exponentiations with T-bit moduli.

4.2.3 Complexity Comparison

Algorithm	Encryption Complexity	Decryption Complexity
Rabin-p	$O(n^2)$	$O(n^3)$
HIME(R)	$O(n^2)$	$O(n^3)$
Rabin-SAEP+	$O(n^2)$	$O(n^3)$
RSA-OAEP	$O(n^3)$	$O(n^3)$

Table 4.4: Performance efficiency between the Rabin-p, HIME(R), Rabin-SAEP+ and RSA-OAEP

4.3 Plaintext to Ciphertext Ratio

Message expansion is another angle where comparison can be made. This area is closely related to bandwith overhead. The larger the expansion the more bandwith is utilized. We provide a table for comparison against the HIME(R), Rabin-SAEP+ and RSA-OAEP. Plaintext to ciphertext ratio is denoted as $m : c$.

Algorithm	$m : c$
Rabin-p	2 : 3
HIME(R)[15]	$\sim$ 3 : 4
Rabin-SAEP+[35]	1 : 4
RSA-OAEP[2]	$\sim$ 3 : 4

Table 4.5: Plaintext to Ciphertext Ratio vs HIME(R), Rabin-SAEP+ and RSA-OAEP

4.4 Conclusion

The ability of Rabin-p cryptosystem to have the following characteristics:

1. Key length comparable to currently deployed public key encryptions algorithms;
2. Fast performance during encryption and decryption;
3. Fair message expansion rate;
4. Does not have decryption failure [refer Chapter 2];

makes Rabin-p cryptosystem a possible candidate for a secure national encryption scheme. Moreover, with the beneficial features that the Rabin-p has, the possibility of seamless deployment within current public key infrastructure cannot be ruled out. Additionally, for good protection against quantum computers, the modulus size of 15360-bit is sufficent, unless Shor's algorithm applies.

Chapter 5

Rabin-p Key Encapsulation Mechanism: The Proposal

The security of a modern public key cryptosystem is usually viewed from their mathematical hard problem and its security model. In this chapter, we propose the design for Rabin-p cryptosystem in the setting of Key Encapsulation Mechanism (KEM) following the KEM framework for Rabin cryptosystem as proposed in [9].

5.1 Preliminaries

In order to facilitate fundamental flow of knowledge, we lay down some definitions. We begin with important definitions concerning with the material of related cryptographic hard problems. Secondly, we outline our security model.

5.1.1 Related Cryptographic Hard Problem

Definition 2 (Cryptographic Hard Problem)**.** *[21]. A cryptographic hard problem is defined as a concrete mathematical object which is easily to compute in one direction, but very hard to invert.*

Definition 3 (Negligible Function)**.** *[17]. A function ϵ is negligible if for every polynomial with integer coefficients $f(\cdot)$ there exists an $N > 0$ such that for all integers $n > N$ it holds that $\epsilon(n) < \frac{1}{f(n)}$.*

Let $\mathcal{A}$ be a probabilistic polynomial time algorithm and a probability denoted as Pr. Then we have the following definitions.

Definition 4 (Integer Factorization Problem)**.** *[16]. Let N be a positive integer. Then, the integer factorization problem (IFP) is defined as the problem to find the prime factorization of N such that, $N = p_1^{r_1} p_2^{r_2} p_3^{r_3} \dots p_s^{r_s}$ where p_i are distinct primes and $r_i \geq 1$. For our case, the problem is to find the prime factors p and q from $N = p^2 q$.*

Definition 5 (IFP Hard Problem)**.** *[17]. Let the IFP is defined as in Definition 4 with the particular modulus such that $N = p^2 q$. Suppose $[\mathcal{A}_{(IFP)} = 1]$ is an event such that $\mathcal{A}$ is successfully factor p and q given $N = p^2 q$, otherwise $[\mathcal{A}_{(IFP)} = 0]$. We say that IFP (i.e. factoring $N = p^2 q$) is hard if for all probabilistic polynomial time algorithm $\mathcal{A}$ there exists a negligible function ϵ such that*

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

Definition 6 (Rabin-p Hard Problem)**.** *Let the Rabin-p cryptosystem is as defined as in Chapter 2. Suppose $[\mathcal{A}_{(Rabin-p)} = 1]$ is an event such that $\mathcal{A}$ successfully invert the Rabin-p cryptosystem and obtained the correct message m, otherwise $[\mathcal{A}_{(Rabin-p)} = 0]$. As proven in Theorem 3 that breaking*

Rabin-p cryptosystem is equivalent to factoring the modulus $N = p^2q$, thus $\Pr[\mathcal{A}_{(Rabin-p)} = 1] = \Pr[\mathcal{A}_{(IFP)} = 1]$. *We say that breaking the Rabin-p cryptosystem is hard relative to IFP (i.e. Definition 5) if for all probabilistic polynomial time algorithm $\mathcal{A}$ there exists a negligible function ϵ such that*

$$\Pr[\mathcal{A}_{(Rabin-p)} = 1] \leq \epsilon$$

5.1.2 Security Goals and Attack Models

The security of public key cryptosystem is usually categorized from the point of view of their goals and attack models. The currently known standard goals of public key cryptosystems are defined as follows.

Definition 7 (Indistinguishability)**.** *][13]. Indistinguishability (IND) refers to the situation of given a ciphertext of one of the two plaintexts (i.e. both plaintexts known to the adversary), and then any adversary cannot distinguish which one is encrypted. This notion is rather artificial, but in considering provable security of a public key cryptosystem it is usually convenient to employ this notion as the goal of the system.*

On the other hand, the currently known standard attack models upon a public key cryptosystem are as follows.

Remark 7 (Chosen Plaintext Attacks (CPA))**.** *. In this model, an adversary has access to an encryption oracle. That is, such adversary can choose a set of plaintexts and obtain the corresponding ciphertexts.*

Remark 8 (Non-adaptive Chosen Ciphertext Attacks (CCA1))**.** *. In this model, an adversary has, in addition to the ability to the CPA adversary, access to a decryption oracle before obtains a challenge ciphertext. That is, the adversary can choose a set of ciphertexts and obtain the corresponding plaintexts during this period [24].*

Remark 9 (Adaptive Chosen Ciphertext Attacks (CCA2))**.** . *In this model, an adversary has, in addition to the ability of the CCA1 adversary, access to a decryption oracle even after obtaining the challenge ciphertext. However, this kind of adversary is prohibited from asking the oracle to decrypt the challenge ciphertext itself [31].*

Several security notions can be constructed by combining these goals and attack models, and, of course, there are relations between some of these notions. In fact, the following facts on such relations have been known so far [37]. First, regarding the attack models, the power of the adversaries gets stronger in the order CPA, CCA1, and CCA2, so does the strength of the security notions. It is largely agreed upon that security against CCA2 is one of the most important attributes of any public key cryptosystem [23].

Secondly, 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 indistinguishability against chosen ciphertext attacks (IND-CCA2). For instance see [2], and [8]. Hence, formalizing and proving for any designated public key cryptosystem resilient to such stronger attack model is very important.

5.1.3 Deterministic Encryption

We will start by considering deterministic encryption schemes.

Definition 8 (A Deterministic Encryption Scheme [9])**.** *A deterministic encryption scheme is a triple* (G, E, D) *where:*

1. *a key generation algorithm,* G*, which takes as input a security parameter* 1^k *and outputs a public/secret key-pair* (pk, sk)*,*

2. *an encryption algorithm, E, which takes as input a message $m \in \mathcal{M}$ and the public key pk and outputs a ciphertext $C \in \mathcal{C}$,*
3. *the decryption algorithm, D, which takes as input a ciphertext $C \in \mathcal{C}$ and and the secret-key sk and outputs either a message $m \in \mathcal{M}$ or the error symbol $\perp$.*

The weakest notion of security for a deterministic encryption scheme is one-way security.

Definition 9. *A deterministic encryption scheme (G, E, D) is said to be one-way if the probability that a polynomial time attacker A can invert a randomly generated ciphertext $C = E(m, pk)$ (where m is chosen at random from $\mathcal{M}$ is negligible as a function of k. Such a cryptosystem is often said to be secure in the OW-CPA model.*

5.1.4 Key Encapsulation Mechanism - KEM

Technically, to use the public key systems in sending long messages is not practical. Instead, they are frequently applied to exchange, symmetric keys, which are comparatively short [1]. The symmetric key is then employed to encrypt the longer messages. The public key cryptosystem is somehow relatively slow compared to its symmetric counterpart; thus it is not suited for encrypting large bulk of information.

Essentially, [9] gives a generic construction method to allow an algorithm designer to construct a KEM from almost any cryptographic problem. As a result, we propose a Rabin-p KEM, that is as secure as factoring, in the random oracle model. Firstly, we recall the definition of the random oracle model as follows.

Definition 10 (Random Oracle Model [17]). *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.*

In the random oracle model (ROM), one assumes that some hash function is replaced by a random function accessible to the public. This means that the adversary cannot calculate the result of the hash function itself, instead he must query the random oracle. This also means that anyone, including the adversary has access to the random oracle [7].

Definition 11 (Key Encapsulation Mechanism [9]). *A KEM is a triple of algorithms:*

1. *a key generation algorithm,* $KEM.Gen$*, which takes as input a security parameter* 1^k *and outputs a public/secret key-pair* (pk, sk)*,*
2. *an encapsulation algorithm,* $KEM.Encap$*, that takes as input a public key* pk *and outputs an encapsulated key-pair* (K, C) *(i.e.* C *is sometimes said to be an encapsulation of the key* K*),*
3. *a decapsulation algorithm,* $KEM.Decap$*, that takes as input an encapsulation of a key* C *and a secret-key* sk*, and outputs a key* K*.*

We choose to approach provable security from an asymptotic point of view and suggest that a scheme is secure if the probability of breaking that scheme is negligible as a function of the security parameter.

5.2 The Proposal for Rabin-p KEM

5.2.1 The Security of Rabin-p Encryption

Clearly, Rabin-p does not achieve IND-CPA because Rabin-p encryption algorithm as shown in Chapter 2 is deterministic. Next we discuss the onewayness (OW) and unbreakability (UB) of Rabin-p.

As described and discussed in Chapter 2, the onewayness for Rabin-p scheme or the Rabin-p decryption problem is: Given public key N and ciphertext c, find m such that $E(N, m) \equiv m^2 \pmod{N} \equiv c$. Section 3 have proven that under CPA the **Rabin-p decryption problem** is reduced to the integer factorization problem (IFP). The proof includes an algorithm (See Section 3.2) which chooses and encrypts a message which is larger than p^2 and queries it to the OW adversary. The adversary then returns a message less than p^2. Utilizing the Euclidean algorithm on the two distinct messages enable the factoring of the public key N. Let this algorithm (i.e. Algorithm 4) be named Rabin-p factoring algorithm. By the proofs of Theorem 3 and by Definition 9, hence the Rabin-p encryption achieves OW-CPA assuming that integer factorization is hard.

Furthermore, from the public key of Rabin-p, which is in the form of $N = p^2q$ where p and q are k-bit primes and $p, q \equiv 3 \pmod{4}$. The private key is the prime p. Hence the **Rabin-p private key problem** can be stated as: Given the public key, N, find the private key, a k-bit prime p such that p^2 divides N. As such, the Rabin-p private key problem is exactly the integer factorization problem under CPA and this is correctly proven in the previous section. Hence, Rabin-p is UB-CPA, assuming integer factorization is hard.

5.2.2 Generic construction of secure KEM

[9] propose a simpler construction for designing a KEM based on a deterministic encryption scheme with weak security assumptions.In other words, a secure KEM is build from a deterministic encryption scheme that is secure in the OW-CPA model. The following Algorithm 5, Algorithm 6 and Algorithm 7, gives a construction of a KEM based on a deterministic asymmetric encryption scheme (G, E, D). The scheme makes use of a key derivation function KDF and a hash function $Hash$. These functions will be modelled as random oracles and so care must be taken that their outputs are suitably independent.

Algorithm 5 Key Generation of a KEM derived from an OW-CPA secure, deterministic encryption scheme

1: Key-generation is given by G, i.e. $KEM.Gen = G$

Algorithm 6 Encapsulation of a KEM derived from an OW-CPA secure, deterministic encryption scheme

1: Generate a suitably large bit-string $x \in \mathcal{M}$.
2: Set $C_1 := \mathcal{E}(x, pk)$
3: Set $C_2 := Hash(x)$
4: Set $C := (C_1, C_2)$
5: Set $K := KDF(x)$
6: Output (K, C)

Algorithm 7 Decapsulation of a KEM derived from an OW-CPA secure, deterministic encryption scheme

1: Parse C as (C_1, C_2).

2: Set $x := \mathcal{D}(C_1, sk)$. If $x = \perp$ then output $\perp$ and halt.

3: Check that $C_2 = Hash(x)$. If not, output $\perp$ and halt.

4: Set $K := KDF(x)$

5: Output K

This construction also has the advantage that the decryption algorithm need not return a unique solution but need only return a small subset of the message space that includes the original message, as, with high probability, the original message will be the only message in the subset that hashes to give the correct value of C_2. We will make heavy use of this fact in the specifIcation of Rabin-p KEM.

Theorem 11 ([9]). *Suppose that (G, E, D) is a deterministic encryption algorithm that is secure in the OW-CPA model. Then the KEM derived from (G, E, D) in Table 4 is, in the random oracle model, IND-CCA2 secure.*

Proof. Appendix B of Theorem 4 in [9] □

5.2.3 The Design of Secure Rabin-p KEM

In this work, we will view the Rabin-p as a KEM-DEM framework, and study only the KEM component. Security analysis for Rabin-p KEM instead of a hybrid scheme is more elegant because the KEM-DEM framework has specified the required security level for KEM relating directly to security of Rabin-p scheme. This section presents the security of Rabin-p as a KEM, following the KEM framework for Rabin as proposed in [9].

Now we are ready to present our KEM design for the Rabin-p cryptosystem. The same procedure is retained for the key generation as described in Algorithm 1 and output the public key $N = p^2q$ and the private key p. We begin with the key generation algorithm as follows.

Algorithm 8 Rabin-p KEM Key Generation

Input: The size k of the security parameter.

Output: The public key N and the private key p.

1: Generate two random and distinct primes p and q such that $p, q \equiv 3 \pmod 4$ where $2^k < p, q < 2^{k+1}$.

2: Compute $N = p^2q$.

3: Return the public key N and the secret key p.

Algorithm 9 Rabin-p KEM Encapsulation Algorithm

Input: The public key N.

Output: A ciphertext tuple (K, C).

1: Choose a random integer $2^{3k/2} < x < 2^{2k-1}$.

2: Compute $C_1 \equiv x^2 \pmod N$.

3: Compute $C_2 = Hash(x)$.

4: Set $C := (C_1, C_2)$

5: Set $K := KDF(x)$

6: Output (K, C).

Algorithm 10 Rabin-p KEM Decapsulation Algorithm

Input: A ciphertext C and the private key p.

Output: The value K.

1: Parse C as (C_1, C_2)

2: Compute $w \equiv C_1 \pmod{p}$.

3: Compute $x_p \equiv w^{\frac{p+1}{4}} \pmod{p}$.

4: Compute $i = \frac{c - x_p^2}{p}$.

5: Compute $j \equiv \frac{i}{2x_p} \pmod{p}$.

6: Compute $x_1 = x_p + jp$.

7: If $x_1 < 2^{2k-1}$, then return $x = x_1$. Else set $x = p^2 - x_1$.

8: Check that $C_2 = Hash(x)$. If not, output $\perp$ and halt.

9: Let x be the unique square root of C_1 modulo N for which $Hash(x) = C_2$.

10: Set $K := KDF(x)$

11: Output K.

5.2.4 Security Proof for Rabin-p KEM

We proposing a new KEM whose security is equivalent to factoring, that is the Rabin-p KEM. The Rabin-p KEM construction will be based on the generic construction given in Section 5.2.2 and the Rabin-p encryption from Chapter 2. The algorithms of Rabin-p KEM is described by Algorithm 8, Algorithm 9 and Algorithm 10, respectively. The provable security proof of the proposed Rabin-p KEM can be summed up in the following theorem.

Theorem 12. *Providing the factoring problem is hard, Rabin-p KEM is IND-CPA secure in the random oracle model.*

Proof. It is proven in Theorem 3 that the Rabin-p function is one-way providing that the factoring assumption is hard. Therefore, given that the factoring problem is intractable, by Theorem 11 the proposed Rabin-p KEM

is IND-CPA secure in the random oracle model. □

Remark 10. *Observe that, the Rabin-p cryptosystem falls prey to the integer factorization based-encryption security incompatibility in the same way as Rabin cryptosystem [30]. This incompatibility is first found by [38] in Rabin cryptosystem and was formally stated and proven in [29]. A simplified statement of the security incompatibility is: If an encryption scheme OW-CPA implies integer factorization problem, then the scheme is totally broken under CCA. Therefore, particularly in our case, it is necessary to reduce the security claims of Theorem 11 which originally proved for IND-CCA2 security to only achieve IND-CPA secure.*

Chapter 6

Implementation Reports

This chapter discuss the computational running time for both the encryption and decryption process for Rabin-p cryptosystem.

6.1 Encryption Computational Running Time

The Rabin-p encryption process involves a squaring and a modular reduction process. Its total running time is $O(14k^2 + 4k)$.

6.2 Decryption Computational Running Time

The Rabin-p decryption process involves 1 modular exponentiation, 2 modulo reduction, 1 division over the integers, 1 modular inverse and 1 addition process. Its total running time is $O(3k^3 + 142k^2 + 154k + 3)$.

6.3 Empirical Performance Data

These experiments were conducted using Microsoft Visual Studio 2010 on ASUS Model G551J, Windows 8.1 with Intel(R) Core(TM) i7-4710HQ CPU 2.50GHz and 4.00GB RAM.

6.3.1 Rabin-p Encryption

Table 6.1 shows the computational time of Rabin-p encryption algorithm when executing on specific numbers of data.

Number of data encrypted	Time (ms)
100	13
500	76
1000	138
5000	717
10000	1430

Table 6.1: Rabin-p encryption algorithm execution time.

6.3.2 Rabin-p Decryption

Table 6.2 shows the computational time of Rabin-p decryption algorithm when executing on specific numbers of data.

Number of data decrypted	Time (ms)
100	21
500	83
1000	156
5000	842
10000	1538

Table 6.2: Rabin-p decryption algorithm execution time.

Chapter 7

Suggested Implementation Practices

7.1 Key Generation Procedure

A practical key generation methodology for factoring based cryptosystems are already established and well-developed. In implementing the Rabin-p key generation procedure properly, we suggested the implementers to utilize the key generation mechanism provided in [12] and [36] satisfying the condition within Section 2.2 in Chapter 2.

7.2 Rabin-p Encryption Procedure

Chapter 2(in Section 2.3) and Chapter 3 (in Section 3.3, Section 3.4) lists out strict conditions for variables within Rabin-p encryption procedures. These conditions have to be satisfied in order for Rabin-p security properties to be realized.

7.3 Rabin-p Decryption Procedure

For implementers that wish to optimize the decryption procedure, we suggest the the implementers to follow the mechanism as described in Chapter 2(in Section 2.4) and in Chapter 3.

Chapter 8

Intellectual Property Rights

I, Muhammad Asyraf bin Asbullah, of Institute for Mathematical Research, Universiti Putra Malaysia, do hereby declare that the cryptosystems that I have submitted, known as Rabin-p Key Encapsulation Mechanism, are my own original works, or if submitted jointly with others, are the original work of the joint submitters.

I further declare that the Rabin-p Key Encapsulation Mechanism are properties of Universiti Putra Malaysia (UPM). It has been filed for copyright as according to Universiti Putra Malaysia (Research) Rules 2012. Their copyright application reference are belonged to the thesis of Muhammad Asyraf bin Asbullah entitled 'Cryptanalysis on the Modulus $N = p^2q$ and Design of Rabin-like Cryptosystem without Decryption Failure'. UPM hopes that all parties interested with Rabin-p Key Encapsulation Mechanism will endeavor to communicate with the owners as well as to cite this document in all future works regarding Rabin-p Key Encapsulation Mechanism.

In addition, I do hereby declare that the cryptosystems that I have submitted,

known as Rabin-p Key Encapsulation Mechanism, are already in publications prior to this proposal submission, as follows;

1. M A Asbullah & M R K Ariffin. Design of Rabin-like Cryptosystem Without Decryption Failure (2016). Malaysian Journal of Mathematical Science, 10(S), 1-18.

2. M A Asbullah, M R K Ariffin & Z Mahad (2016). Analysis on the Rabin-p cryptosystem. AIP Conference Proceedings 1787, 080012.

3. M A Asbullah & M R K Ariffin (2016). Provably Secure Rabin-p Cryptosystem in Hybrid Setting. AIP Conference Proceedings 1739, 020001.

4. M A Asbullah & M R K Ariffin. Algebraic Analysis of a Rabin-Like Cryptosystem and Its Countermeasures (2017). Indian Journal of Science and Technology, 10(1), 1-5.

5. M A Asbullah, Z Mahad & M R K Ariffin, Efficient Programming Deployment Strategy for Rabin-p Cryptosystem in C/C++, MyIPO Copyright Filing No. LY2018004528, 27 September 2018.

6. M A Asbullah, Z Mahad & M R K Ariffin, Efficient Programming Deployment Strategy for Rabin-p Cryptosystem in Java, MyIPO Copyright Filing No. LY2018004528, 27 September 2018.

Finally, I will undertake to update the MySEAL project when necessary.

Chapter 9

Full Size Test Vectors

We now provide the interested reader 3 full size test vectors for Rabin-p Key Encapsulation Mechanism. These test vectors work on the prime number p with 1024-bits (equivalent to 128-bit security level). We remark that all parameters are as defined as in Section 5.2.3. The hash value $C_2 = Hash(x)$ is using SHA-256 as the symmetric hash function algorithm.

9.1 Sample Key 1

The public key N:

60991B16F64E4BBB902562B527E4CCD3FE1181D88BF36301A7CA8DEF
CB29295E7C2583B84AB2F5BA7BD1E420CA88961B2975EE797F222AC14
655394C2F643A62FD63736EE57CFE0FD36E29E2C1195D676DAD1582887
C82D99F701A23A6497C09BF4B46ECDE12F4ED815A0FFD922210B27FF
2C4A51646A9264CF5B08985EBD81AE12493551783681F3106BCA24906FA
C6F7F149EC0609022A7C80CDE49D630E9ECC81842703B22A2ED946D21
B4A30A74E0D343CB9C8DF6F768E2D1FB3DF65C7CAF13BB4D7CBC1A
F00FA475817C738B843CD152AF059427CD62DEFF0E696208B1B7487F6E

9DD7D4E389A187300A8BC2F3900F02E0DAAEE618F5C298F23B7E1E60
0D227450009042AA1D4AEA81F6BFD6723B46F008CB63D9C9C731D38D1
CC66A37B75D65EEB73429801F658A730C85EDE2BB31BA768D84D27681
730F692F53291CEC0B3DB994D8C4977804404A9DA2DAC94F551DE1F82
FFE6A6D76C8F067005814758EEA549B94E7515C76B93B4EC52340E5BA9
C1DC3B6CF84E7F9357CF169F0E2B

The private key p:

9F68093FE147E282A55A37D32AA1DB511C50C2EC0791584BC738AF47A
47A5EEF3A41DE8E40B4C4ED893631758194B51F58E3AAC460585A47205
F1615D5C30185331667392AF029DCD448255CE86D6BBF1F9BCBB7AF62
BD9767D90D6B95CABAB102A6600B036BC5F8DEF4634A2BAD2EE5D62
B63819A5DC13140AEF19AFF0105FF

9.1.1 Test Vector 1

The random integer x:

0718A9E9D65C570D2DD30DBA0B365C4C10547626064EBB8B4A6591E50
AFDEC6F384D9EF33A12CEDF32ED721890A958FE9EAADA6AB87FDF9
63A05AC6B72A63AF0BF8063DB2AC6DCF168C6CEEFADB2178B846775
0E04CDE9D00827C1EFCF7FDB3789F0D8CDD881886626567D84FD1645F
357ED5B686E8CC4FF0F4B8387FA987DA8B3031D535B72754618D64D393
44E83BBE6538A91B18A5493BC31D916D297FDD58B59002B32FEC917272
C45BD0C41804285EDC7DBF33962DD651633A41D3F923834AACB9B5F6
684CAED005420CD8B72E6F70401D7B574FFC8CAFD1422AA13C2078921
C4465961522A193E34C368FC522E3877BE6F067034E976071CDD3966FAC
2

The ciphertext C_1:

53C1E2231BEF8E1DBE117E44A7C2C1CAB220926AB02F0999E6901D977
D421A183426CEA9A60AC69E8234617BAB4310067C6E9240A296A6F31FC
9A0F4364D38E32BB444D8CCFB03F0DBCE77F723ACAA39D6D31702FE
02A932FE4CA4908329F242BAFB1B5D701C701FEC1F691F3A569E2C91C
CA6903E025745F9C4AD90262627A73051FCE91B407E1BB11C57125A9B0
2F26907B2390C79A8185398CF2146B906DF3B346C0DECB93B1C98A2234
B06E1EE6E2D36DFF2ED66DDADFB1D63A69535CFA8DB4CC4A792D5
4DCF92EF278EE9244BE11A6229554426E2282FA07655FF4AED4285D7A
0E493536069C35C5C1BF6E679CE26C5F92CEB2930462CB99D125E10E45
2A89F62F05FF7162C589A87AD7D76C250D07937917A9387A8A523D7E77
68723062756DB82CB573F40C3AE5EC7D3B2446CA123A74441BD9CE0E3
3A4620CEDFE040147D8675826E7C6A0DBA751F4D49EF22E701CFBBA8
44FAAE61A70FC2A9C0E5538C04E72FE55DA61B851A5DB53D4E7A6CD
D50821BF9A5C753F816C6668DBF4A31

The hash value $Hash(x)$:

91F624BF6A5C421AB6BB28F5079AEB69A1072917BC0A727ABDC690DB
6BE3F61C

9.1.2 Test Vector 2

The random integer x:

1ECD4B6D611937F67C6B42FBE26C473A1A403ABDFADD86F5A0BF677
069FE644A8FFF9A5E47DF44A1F3C3340F7C8B22EA2CA2B075AA0E208
2EEA3E659B049B8B8B8C6294B9A35719B29E685F3B461489AA3DB6F06
F40D5E9C569E98BFF5D1B4F944AE8493D88AAC6F5EC49441D122ECE3

D916FCF04FDCA847BBCB0C05DCC328974EB3D2692EE164DF570A956
F0A33E7F5D7E54856A1768D8F6BDC714295D6088C5465EA02C217CFAF
2F20A7F4E92B15D689F048CB951CD6012B0B5B3719DD020598299096E64
FAA23106248C1AA586455A91BC642E5CB9197A18584545F1AF3894CAA6
0BD44CE0DACE0D21EA71C58AEE24C39CA30C3E663D075763A8EE523
92E7

The ciphertext C_1:

14EE1803219D42B523F0A165DBF476C0314037C6DD4479A128D803FCEA
8025F533868B220C8B8D4FD50423BEE64015239C3249A8CF9591CB1212A
BE2D32FAD51C1AFB8B6D1EEE8F9B870B94A961A37D83E94E8CF2990
11ACDBD136F3D8DE250FFE16426CE666A4637444BBD557C1F565EAF4
5F52051ABD1A29065985F48E334BF13AC383FC972E20F8779DD9DD463
75A94A80CE0F5ED8483B149BA24615E7F29448DCA99953C678E75D7DF
CE4BB5016F83FB11FB0C9B3F96BC73E906AC013CBE636F31A2C424B7
637970260356662F885721539198E1C2ECC7F897F129E1E9F93CDC10E3E7
C7859AF0233BB704368B5F34F61EC2AB1A893ED6EB896008792A4C60A
4B509E2D3FDDD74204DE4AE2DCCF2FA038278FD4BCC8F5026E799AA
399B310FE780515F2CD562C753DB8F86EFD31C4F7B0F9AECF7B6D43E
0BB9BA3682CAD274478A375E98E56B7263F91E5307A00CCA8ABC63DF
5DFE08A03B7964FAAB01845ECC687385BF0BFCBDC7CFE8EC0F7A477
38D56DF2DB7533509C600EEAA185C24

The hash value $Hash(x)$:

89341F698854C013481922C0C32C40174C5362148F89274FA57B7D7919615
990

9.2 Sample Key 2

The public key N:

8361303AC8602B85150B82E37D004ABD36E6E69DC7B8C1CA3DD46728B
0E57B9E954107CF1999C1E8AC5C2EE2F2B21A864546377B1672300491B0
4E304D77A308D3ACAA7C97B8D112D7D5AEA290C9C7068DC2BCE45C
E65BB51E202BCBEB27C37EA371D969CFB0EAEFBC1E15AD1CD331275
F73BCD411C583666A8F3760BFD752D14DEC54DE089924E51660FC78DA
5EAC3507AA545E6C7E96B378D792D2B02B70A377F222190C7572870FC2
DF4DA9012890B66DB64DD95D713085D5667C2EC199C5830C78458EBA7
C49E3603CFA0FDAE1F790CD59C5639435DF7BEF76FAF16D567DEAA4
4C7D8E6958414B855B8CC5230C308F41ACFF756CCC150B8962493E1352
69BB58858E7C3CC96C9DFF35579F310B0910B8E9E5104EAFA4FAFEA8
DA8F4DB9AF452CEC653FBC1703BAD6101607DE1961ECB4EE4EC19EE
B60F549C0D731DFBABD273BEDC08EA5C65D92E7992D06303BCB4C20
562530ECDF9145450F474E669385EE27B0F1C947C92870B91B361FADBD
36C95432F669100C00E1867396F0331917

The private key p:

DEEE7964B8C4A8AEF5CD09213E6BD63977E38330E4A536A2EFC927E9
2BDA0B5C637C31F7AA089B4C743C7175998F7C973372EB85639934C4032
E5F9026798F8690BCA71D8EA7DEF19AD4CBA64CF766A7B067CA64C76
A2CBBF948CA170280C376077FDBEFBBA1A758A76D94A8E9DFCAA5F
FB2100BFFB914355E589EB02B494B97

9.2.1 Test Vector 1

The random integer x:

18C5975532F6EBC7AB76A6C459228684C3A5835EA2C622E71D5BF43205
F24E24E007E59B2D66EC10C159008259FA3182FD62DD4947F6D8B61F05
B8F6F926F04B8B428B7498BBB9D1666A6829A75C488C85C607F99A46C9
300EBFCBE0E910C05248A1C0F24C7090358BDEFD035D7F32F2F6D0F1
780323FD3A8E123B7BBAA3F8A793F0CA4E45AFF2592D40798CED3ED8
857CAB473587A17B59145B4DDAFEED544DCD130D8449524087BD37FD
6CFA015BB33424172853273473A9FF3AA91F6FC295D2617F83466A1A778
BB64EBC3E4DBB6A11F6FA247CD2E12753E9608BF3B67AC2F657CD70
4148E43FF25FA212402CAD56BD54307CE20ECC78E43513ED6DB7BD6D

The ciphertext C_1:

6C3C53147D1495BB8048A91C45E37C9F99A1D1AFFDA3D4476E2D8A30
E1C216844234A272DF551226BA05FF6A37FD28766370C75B918432A835A
3DA818EAE00855A4E18165CCE601A58BEE62B636D4A94CB73CFFD33
D5B8D7D6A1A11F90BD08DAE9E9EA75A94A2D81858E8BEF36B4E92B
B47A1A88F55A90FB7903438EA78C13E8ADB49A2FACA1236EBA94C859
9DD44753818EA34773B301DF64EBB9938664CAAA42501243C7C008A10C
A99D75F77EC8D016DB20A7849BEA2A8D378C3D25FFAEC0443877B9E5
8E8F0636EFAA439E54DFD02ED3D84C775E3BB4A31997700EE5CDB0B8
03AF0CEC4660F298A650DA0E83ED044801AF43A57B9B04A0778262D33
C992B0E6AB09E2A2368856E540C5AD9E8DBF1DA97426BD1610CE8DC
FFA080A4A31F8A39542EE6D1600D09F475ADE5161E369169B1DB2CBA
61EF8FD1E2028EE592E19523B6C0001E196DD2E54131A7B3871BD0CBC
D3B48156C7DF133AEB143538B588C8EEC45F75A78E2B528D453112C60
ED9D6AED7265B91FF599461BDF6F24BB7371

TThe hash value $Hash(x)$:

FAF9FB73F42420D1EA11D3990A51657AAE0F647F5CEF2CAEFF1D1BF
6C3E5581D

9.2.2 Test Vector 2

The random integer x:

1642175D484D9F1E076D7B31159E79EE299197C3EC330AB9333AB903608
04013351C58282C2AC314DE108FC69CA7A2D37BC5A86D0C87607A0E98
40019B98C8692E1E667C73059EAF84DA0911CC8FEC92433078F572A7976
DB00F14D649DCEB4511CC6053E5A5B8B1A1A38AE995C977A892D2178
4145BEF38B3986A10A2483E2E03F323C0B5AD4301061C57FE70848564D4
C9A82A39C8747E62CDD08A3FD8C68190479EF81541F755160A8CADBB
EE62C802D37571813FBEDB4ABDFD832449C0C12D21F070A5A35D5C10
59F4C4F31864475A51C78136A873285E4810E2B6C6045F694A1238181E052
BB5F524E681AA4A5FA59ED32107F94BDDF3184E91291092E5

The ciphertext C_1:

4E7AE2E7AC86CBDB15B072311C5B997715E66AB47D94B3154B05FDA2
00B1D3F4E86A347A40DA183AA2BEF538FB4B37835B5F54F845384ECD
94E7D0502597DAF11D0A9C9D149AFC464DDDE26D7AA4111BFC037D0
AAB6CC9985C7AB2ED9BAFC0143D354C919D57DA507857097A53CA3B
E8D80960BD813A6D3B958B0D8A7B169F038FFFBEAE60216F8D0895851
ABA313E0CD87F0C99F337B8EC6F963725C26385FB2528AB2B076D53CD
1E035A8408282F8F9BE589CDF66ACB57BAEE3739828C463EE9258BA7A
E6B1920CE93747D1B608E34BB9A200958509D08A8529BC77B0F2B89ED8

16F5D8B8D645461EF0ED82B1C832FB2D32BEF939254B20DF07C0B02F5
7C9CD372152E003ACEABEF266D37F853E1BEE56CC573E4DB408DD3B
69574B65A57C10186EAA1446989CDC6F9025C05FE6649601EAF4FFD46D
E195C6B3C7F342C6FAFF7D48F08F9DDD5C5720E07067C68C9A759F588
9AC335A0B88FC6D427DBC1EC075839A5A95A312DA42BC72FD4CE0AC
833E05ECEE182E09E14DF91F54B1C29701E

The hash value $Hash(x)$:
C6E4B3585E07BA91D60A4F4CDBB39A09093072F6F20AC6425BA963A0
A8FB8888

9.3 Sample Key 3

The public key N:
457601732E1F91E8673E15982C7190F2CAC121BC24D4590BF8837C444F9
7A47CFC36A47D46EBBC5C012F1250E0AA379CBF1D25E4CD0CD5C2F
6D41607A8BFB17BAD9B81D0C658C56AAC79AFA51E5D9DEE3C832248
1F9A4B7277DCA8FFBEB18218CACE9D8487FEE7B91FC5054D3F5DE1B
B1A7838F45D3770EA3D6C9E5D01AA685CC3B5DE077BED842EEA176D
340225E02E32E41587F30E8E8FB139F32A1099276F34324EFFD0C8FEAF
FE1E414B17F765043B171342B5AE40DF024D3A56D95867E1FD8545BB2F
9790FC579BCE5AEE30CC95A4CA81B1A48CEFB5435F6BEFE54A6ABA
95BDEC6C04A8F8C2787EAD0533B37B48D113AD2972832E93139DC521C
B9210584A32E6DDA985E3194F185CED66CABD064CBB6748E7407E6BC
762732FE3ECB7663B6A24BE2BB2BE522A808473F0689921888CF20C189
AA9EA77ADB1E873EDED9BFD823D235BAC2AC49ECBCDA9B594B8F2

274A58F75299ECDC87E83E72AEC141C6AB44229CF24C280C1B16259ED
704D30955CB9FF1561EBC8E51A4C8B06E8D1F7F

The private key p:

ADE0FE9E24C7FBC63C702B8853DD98D3DFCA054E8CC7E0B2B680FE0
A8B79576327CC2612FD2DC986967F06D1CF20AB745DBEEB6781287B2A
98B3CCE9B5105F11ACD86D95FC1F128DC7E4E0C82BBB9FC2615FC709
96F1F6105D0AE36920E4511A4C087C0DE2F350EE40813E7B34DC55BEF
568C4714D613A0CF16EA145B7152FE3

9.3.1 Test Vector 1

The random integer x:

19DAD4AD323624CD281AF1770B8F9C8337E94D5AF8056B7E07AB4A42
B0B09CADF5CED628DD4719A3F47485A83DFF2EA0A37BFE072BBCC8
453E6EBD39CE71016310062E0244962045FA67761FB6259E758D650F7159
AA73C51C9C451136C13552287C7A06A6CC7C46DA95E41EC5358505670E
8C53B24DDC097C9A894FC948F4FB8EB67173D147D2D59209082ACF46E
2C1C8CAEC5C8BD4C268DBC99549A030874906EAAC4097F0017FCFE42
88CB0E8FC53D331CF2EEFDCBC51FFFD0021E8C5CF8302CC422A65A
BBEE9D281F3D71D534C96A23195034AF9FAFCF5EE506B5C3120F7AB7
3243877662C92E13F9D7314B185501FAB8991D7D094E511BDE5AD06135F
80

The ciphertext C_1:

281EF58B2F124A3A5F6CB474242CE1C7D095A73DE8F62594A8940C6FB
856D0AB8FDA63560A4C9509E1D8A8354A2C96F907C6D4942991C45599D
DD122CCE292F645063A3C7425582FD64033A68B7155D7829CAF664DA202

E246B2C7522B1DB14A7869B4DB558AFE516A3A1CF6E2BD15C17342B1
5B01F802B8071CE4F3C4BA779147AF12A449DBFF4FDAE9F995D28AD
61E68584E5DD9CCB1DEC47B046F1742773B4047247B168E990BF0F3E65
4AA9A4C0F7E403947EC50051A30EAA5710C951151B244F544BF39AA77
0F1AA6BEDF239B3EFE0251AFD67CE4D59409728DDD12F37AB5BFA84
A10DB8D4BBB27598386C5F4C2AFBE444AA5379E862554E26740917CE4
237C72EE8D8B25EAAB1724B1DFA4F7F19CDB4B7BC469070776255CE6
F20A0A1450F0068C384503093B6D96E11648E3AF316EEF1CE9C4C1F451
BB1BECFFEF8BCF210EBF169A8615E4819B32C0906D08166DD324FCA6
33421C7B9D5A76E81CD6B0C1712EC2E3D2D213734C7FE36ECCB9717B
6768A000AE7E5BAE27D2988EC0B50C6

The hash value $Hash(x)$:
62DB0258794ECFB08722A3EE866A43314DA7C1CBEEE0102584A1B5196
37FE779

9.3.2 Test Vector 2

The random integer x:
06B3E6C8FA44E0AD7682B724866A3B8018F83C2A250BC6ED44760B9B8
F668B9D13B84FC6F7D1B2A145F9D5C435A0A5A3AC77DAC5B79EB5FC
04EF58E15C0355FF040046EA8633E66CB6354268B9E1481AA60217264E3
37FE86027E6EC547BB4FE5C4AAF80DFF560F6D1757D088EF4658987748
29664361A0E0108A2434EB7C1B8CCD76F257A14C4663610E457C08318E0
03961BE8A4D5342D31D5931ED54965D49B93E3D8C266302F4860AED1E7
AEAEF1D344CF418BFA53A5470DA7DEE1F4B8FC7299C867F621961394
151F4AE7561FFCF8B07EF7B8A29177D460D18EEAC0677486BAD3B9D3

845EB0B392600BCFB6030F8A01F7362EB3ABCBD2B6DCC5552329A2

The ciphertext C_1:

3DF509D9BA9E25F5D34D29138C28AADD8A327C9C2B9BD9DC32089251
A74AA31775FED7A6BAF437E6CE7B9ACB178E3D2EA2E85483E2165184
0400E3302056B4F2B97A79A231CF85A6E29128F6FEE18D98D4636985AE
B4A111DF248B43067A5B75E76C3BB5C6F0F78535F6C9E880737B3CD312
144568CE7017D569415BDAF40CDCBED6FCBFC8BE5403744D1DAACC
ACD47EF10F7C4FE65E86572B9182CD84E2F00463BC1640C0661E23B729
D5472DF78D0F2357B8015A7CA1654D36C6F5540BB7124B17D65FE824C8
3E1F2D3AE8DCB09317ACDECB4D38304FFBEBE06C4826558DAEEEC8
7271488208826FBBC4AB21176F33CD57287BA6C67FBD99DD6C193E888
802D09F36FFA32696CF6951484AF2DFD0116FF3EBAFB8CC2E970672A1
E16F4AC09C95CBD3D291453ECBE372B79651C3B80F7D43F37C9C76101
0CC842100CC18FDF2F46E18A8C2BE4D27C27EFC173BEA15287CFF8E6
BDC998AF012F633FBFBF0A24186187F9B9969895E73E2BA119BE2C2545
28A9D1EBB5EC3A047268F66C51E8834

The hash value $Hash(x)$:

5415BBB263767E9320616BA95C3A566D8402EDAE9A4E8189C6EBAF162
193AB4A

Bibliography

[1] Masayuki Abe, Rosario Gennaro, and Kaoru Kurosawa. Tag-KEM/DEM: A New Framework for Hybrid Encryption. *Journal Of Cryptology*, 21(1):97–130, 2008.

[2] Mihir Bellare and Phillip Rogaway. Optimal Asymmetric Encryption. In *Advances In Cryptology - EUROCRYPT'94*, pages 92–111. Springer, 1995.

[3] Dan Boneh. Simplified OAEP For The RSA And Rabin Functions. In *Advances In Cryptology-Crypto 2001*, pages 275–291. Springer, 2001.

[4] David Brumley and Dan Boneh. Remote Timing Attacks Are Practical. *Computer Networks*, 48(5):701–716, 2005.

[5] Guilhem Castagnos, Antoine Joux, Fabien Laguillaumie, and Phong Q Nguyen. Factoring pq^2 With Quadratic Forms: Nice Cryptanalyses. In *Advances In Cryptology - ASIACRYPT 2009*, pages 469–486. Springer, 2009.

[6] Don Coppersmith. Small Solutions to Polynomial Equations, and Low Exponent RSA Vulnerabilities. *Journal Of Cryptology*, 10(4):233–260, 1997.

[7] Jean-SÉBastien Coron, Jacques Patarin, and Yannick Seurin. The Random Oracle Model and the Ideal Cipher Model are Equivalent. In *Advances In Cryptology–Crypto 2008*, pages 1–20. Springer, 2008.

[8] Ronald Cramer and Victor Shoup. Design and Analysis of Practical Public-Key Encryption Schemes Secure Against Adaptive Chosen Ciphertext Attack. *SIAM Journal On Computing*, 33(1):167–226, 2003.

[9] AW Dent. A designers guide to kems. cryptography and coding, lncs 2898: 133–151, 2003.

[10] Steven D Galbraith. *Mathematics Of Public Key Cryptography*. Cambridge University Press, 2012.

[11] David Galindo, Sebastiá Martỳn, Paz Morillo, and Jorge L Villar. A Practical Public Key Cryptosystem from Paillier and Rabin Schemes. In *Public Key Cryptography - PKC 2003*, pages 279–291. Springer, 2002.

[12] Damien Giry. NIST Recommendations on Key Length (2016). *https://www.keylength.com/en/4/*, 2017.

[13] Shafi Goldwasser and Silvio Micali. Probabilistic Encryption. *Journal Of Computer And System Sciences*, 28(2):270–299, 1984.

[14] G.H. Hardy and E.M Wright. *An Introduction to the Theory of Numbers*. Oxford University Press, London, 1965.

[15] Hitachi. HIME(R) Public-Key Cryptosystem. *http://www.hitachi.com/rd/yrl/crypto/hime/*, 2002.

[16] Jeffrey Hoffstein, Jill Pipher, and Joseph H Silverman. *An Introduction To Mathematical Cryptography*. Springer, 2008.

[17] J. Katz and Y. Lindell. *Introduction To Modern Cryptography: Principles And Protocols* . Chapman And Hall/ CRC Press, 2008.

[18] Paul C Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In *Advances In Cryptology - Crypto'96*, pages 104–113. Springer, 1996.

[19] Ramanujachary Kumanduri and Cristina Romero. *Number theory with Computer Applications*. Prentice Hall New Jersey, 1998.

[20] Alexander May. *New RSA Vulnerabilities Using Lattice Reduction Methods*. PhD thesis, University Of Paderborn, 2003.

[21] A.J. Menezes, P.C.V. Oorschot, and S.A. Vanstone. *Handbook Of Applied Cryptography*. CRC Press, 1997.

[22] Thomas S Messerges, Ezzy A Dabbish, and Robert H Sloan. Power Analysis Attacks of Modular Exponentiation in Smartcards. In *Cryptographic Hardware And Embedded Systems - CHES'99*, pages 144–157. Springer, 1999.

[23] Siguna Müller. On the Security of Williams Based Public Key Encryption Scheme. In *Public Key Cryptography*, pages 1–18. Springer, 2001.

[24] Moni Naor and Moti Yung. Public-Key Cryptosystems Provably Secure Against Chosen Ciphertext Attacks. In *Proceedings Of The Twenty-Second Annual ACM Symposium On Theory Of Computing*, pages 427–437. ACM, 1990.

[25] Roman Novak. SPA-Based Adaptive Chosen-Ciphertext Attack on RSA Implementation. In *Public Key Cryptography*, pages 252–262. Springer, 2002.

[26] Tatsuaki Okamoto and Shigenori Uchiyama. A New Public-Key Cryptosystem as Secure as Factoring. In *Advances In Cryptology - EUROCRYPT'98*, pages 308–318. Springer, 1998.

[27] Katsuyuki Okeya and Tsuyoshi Takagi. Security Analysis of CRT-Based Cryptosystems. *International Journal Of Information Security*, 5(3):177–185, 2006.

[28] Pascal Paillier. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In *Advances In Cryptology - EUROCRYPT'99*, pages 223–238. Springer, 1999.

[29] Pascal Paillier and Jorge L Villar. Trading one-wayness against chosen-ciphertext security in factoring-based encryption. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 252–266. Springer, 2006.

[30] Michael O Rabin. Digitalized Signatures and Public-Key Functions as Intractable as Factorization. *MIT Technical Report*, MIT/LCS/TR-212, 1979.

[31] Charles Rackoff and Daniel R Simon. Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. In *Advances In Cryptology - Crypto'91*, pages 433–444. Springer, 1992.

[32] R. L. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications Of The ACM*, 21(2):120–126, 1978.

[33] Werner Schindler. A Timing Attack Against RSA with the Chinese Remainder Theorem. In *Cryptographic Hardware And Embedded Systems - CHES 2000*, pages 109–124. Springer, 2000.

[34] Katja Schmidt-Samoa. A New Rabin-Type Trapdoor Permutation Equivalent To Factoring. *Electronic Notes In Theoretical Computer Science*, 157(3):79–94, 2006.

[35] Victor Shoup. Oaep reconsidered. *Journal of Cryptology*, 15(4):223–249, Sep 2002.

[36] Victor Shoup. ISO 18033-2: A Standard for Public-Key Encryption. *http://www.shoup.net/iso/*, 2006.

[37] Yodai Watanabe, Junji Shikata, and Hideki Imai. Equivalence Between Semantic Security and Indistinguishability Against Chosen Ciphertext Attacks. In *Public Key Cryptography - PKC 2003*, pages 71–84. Springer, 2002.

[38] Hc Williams. A Modification of the RSA Public-Key Encryption Procedure. *IEEE Transactions On Information Theory*, 26(6):726–729, 1980.