Identity-based Mediated RSA Revisited

Ibrahim Elashry, Yi Mu and Willy Susilo
Centre for Computer and Information Security Research
School of Computer Science and Software Engineering
University of Wollongong, Wollongong NSW2522, Australia
Email: ifeae231@uowmail.edu.au, ymu@uow.edu.au, wsusilo@uow.edu.au

***Abstract*—In SSYM 2001, Boneh, Ding, Tsudik and Wong presented encryption and signature schemes based on the identity-based mediated RSA (ID-MRSA), in which the users are not allowed to decrypt/sign messages without the permission of a security mediator (the SEM). This allows a simple key revocation. Subsequently, in CT-RSA 2003, Ding and Tsudik presented a security proof for these schemes. In particular, they stated that 'IB-mRSA/OAEP encryption offers equivalent the semantic security to RSA/OAEP against adaptive chosen ciphertext attacks in the random oracle model if the key generation function is division intractable'. To make the key generation function division intractable, Ding and Tsudik used a division intractable hash function to generate division intractable public keys. In this paper, we show that using a division intractable hash function does not necessarily mean that the key generation function is division intractable. We also modify the ID-MRSA so that the generated keys are always division intractable. We also show that these modifications do not passively affect the efficiency of the ID-MRSA.**

***Keywords*-Identity-based Cryptography, Mediated RSA**

I. Introduction

The notion of identity-based cryptography was suggested by Shamir [1]. He also proposed a concrete construction of an identity-based signature scheme. Identity-based cryptography offers the advantage of simplifying public key management, as it eliminates the need for public key certificates. In Shamir's seminal paper, he successfully achieved this goal by designing an identity-based signature based on RSA, but the construction for identity-based encryption could not be achieved using a similar approach since sharing a common modulus between different users makes RSA insecure [2], [3]. Sixteen years later, Sakai, Ohgishi and Kasahara [4] proposed the first identity-based cryptography and independently, Boneh and Franklin [5] proposed the first reliable and provably secure identity-based cryptography, which is based on Weil pairings over elliptic curves. Cocks [6] presented a scheme based on the factorisation of a composite integer. These cryptosystems opened a new era in cryptography.

Boneh, Ding, Tsudik and Wong were the first to introduce the notion of mediated cryptosystems in [7]. They designed a variant of RSA that allows an immediate revocation of, for instance, an employee's key by an employer for any reason. Their system is based on the so-called security mediator (the SEM) architecture, in which the SEM is a the semi-trusted server. If an employee wants to decrypt/sign a message, he must co-operate with the SEM to decrypt/sign this message. The idea behind their scheme is based on splitting the secret key of an employee between the employee himself and the SEM. Hence, without cooperation from the SEM, the employee cannot sign or encrypt the message. This is also helpful to monitor the security of sent/received secure messages in the company. The SEM architecture was proven useful [7] to simplify signature validation and enable key revocation in legacy systems.

Subsequently, Ding and Tsudik proposed an identity-based version of mediated cryptosystems [8]. It was the first identity-based variant of RSA encryption successfully proven secure under the random oracle model [9]. This scheme is based on the optimal asymmetric encryption padding (OAEP) [10]. This padding scheme is used to randomise the message prior to encryption. It uses two hash functions to mix the message with a random string. This makes a deterministic cryptosystem, such as RSA, into a probabilistic cryptosystem which does not give a *priori* information about the message.

Identity-based mediated RSA is completely compatible with public key RSA and it also supports optional public key certificates. The vulnerability of this cryptosystem is that it is based on sharing a common modulus between different users and consequently, requires specific conditions to assure its security: the SEM cannot be compromised during the lifetime of the system and the hash function used to hash the identities must be division intractable.

Our Contributions. In this paper, we review the security of the ID-MRSA proposed in [8]. We show that hashing users' identities using a division intractable hash function does not necessarily generate division intractable public keys. We show that an insider attacker can breach the ID-MRSA even if the hash function used is division intractable. We present two solutions that make the key generation function division intractable and hence, the ID-MRSA is secure.

Throughout the rest of the paper, m represents the message, c represents the ciphertext, k represents the security parameter, p and q represent the primes that generate the modulus n, e represents the encryption exponent, $\varphi(n) =$

$(p-1)(q-1)$, $\lambda(n) = Lcm(p-1, q-1)$, d is the private key, d_u represents the user's private key and d_{SEM} is the SEM private key. $KG()$ is a hash function that hashes the identity to the user's public key.

Organisation of the Paper. The rest of the paper is organised as follows. Section 2 reviews the related work done in this area. Section 3 discusses the ID-MRSA encryption/signature schemes and their implementation. Section 4 discusses the security flaw of the ID-MRSA. Section 4 proposes two solutions to overcome the ID-MRSA security flaw. The effect of using these solutions on the ID-MRSA are discussed in section 5. Finally, we conclude the paper in section 6.

II. Related Work

Mediated RSA and its identity-based variant can be classified as key revocation schemes and two-party RSA schemes. In the following, we review some of the related work done on these two types of schemes.

A. Key revocation

The key revocation problem received the attention of the cryptography community because a public key cannot be used if the corresponding private key is compromised. An initial solution to this problem is certificate revocation lists (CRLs) [11], [12]. The CRLs hold the serial numbers of revoked certificates and should not be used. The disadvantages of CRLs can be summarised in three points:

- When a user wants to encrypt a message, he must verify the validity of the certificate that holds the public key. This requires access to updated CRLs. To achieve this goal, an online validation system is required. This negates one of the advantages of public key cryptography over private ones, which is the 'self-authentication' of the public key certificates.
- A third party, called the validation authority (VA), is required to validate each certificate. This third party must be fully trusted because a user will not be able to receive any messages if his certificate is mistakenly revoked. In addition, if someone attacks the VA, the whole encryption system will halt.
- Because CRLs must identify all revoked certificates, they may be too long and consume the network's bandwidth. A solution to this problem is $\Delta - CRLs$, which contain the revoked certificates since the last issued CRLs [13]. So instead of sending a complete list of CRLs, $\Delta - CRLs$ are used to update the existing CRLs.

To overcome the disadvantages of CRLs, an internet protocol, the online certificate status protocol (OCSP) was introduced [14] by Myers et al. Instead of sending a complete list of revoked certificates periodically, which results in substantial network bandwidth consumption. A client sends to the VA a certificate validation request. Then the VA responds with the status of this certificate (revoked, non-revoked (valid), unknown). One disadvantage of OCSP is that it does not support binding signature semantics because it is impossible to ask a VA if a certificate was valid in the past. Boneh [13] provided a technique to support binding signature semantics, but unfortunately, there is no infrastructure to support his idea.

Kocher [15] suggested an improved version of OCSP, certificate revocation trees (CRTs). The VA can be considered as a global service provider, so it must be replicated using many servers in order to withstand the entire load of certificate validation requests. This means that the VA's signing key must be distributed securely over many servers. This process is expensive and insecure. Kocher suggested a solution to this problem: a highly secure root VA sends a signed CRL-like data structure to other less-secure servers, then clients can query these servers for their certificate validation requests. The data structure is like a tree, where the leaves are the revoked certificates and the root is a signature of the highly secure root VA. This structure is called a certificate revocation tree (CRT). If a user wants to check the validity of a certificate, all he has to do is to send a request to the nearest less-secure VA server.

A disadvantage of the current CRT structure is that the whole CRT must be recalculated and sent to all servers if a new certificate is revoked. This problem can be solved if the CRT can be updated without the need to recalculate it all. 2-3 trees proposed by Naor and Nissim [16] and skip-lists proposed by Goodrich [17] are two proposed solutions to this problem.

Other examples of key revocation schemes include efficient revocation of security capability in certificateless public key cryptography[18], the secure mediated certificateless signature scheme [19], the efficient mediated certificateless public-key encryption scheme without pairings [20], pseudonym management using mediated identity-based cryptography [21], mediated ciphertext-policy attribute-based encryption[22], an identity-based mediated signature scheme from bilinear pairing[23] and security-mediated certificateless cryptography [24].

B. Two-party RSA

Two-party RSA schemes are based on sharing the private key of a user between him and a server. Examples of two-party RSA are the Yaksha system [25] and S-RSA [26]. The Yaksha system is a security infrastructure that enables key escrow and key exchange. It enables an authority to know the short key session of a user without knowing his long term private key. In the SEM architecture schemes, the RSA private key of a user is shared between that user and a Yaksha server so that the multiplication of their keys forms the complete private key. When a user requests a session key from a Yakasha server, the server generates a session key at random, encrypts it with the user's public key and partially decrypts it using his partial decryption key and sends it to the user. The user partially decrypts it to recover his session key. Compared to the SEM architecture, this scheme is more expensive and in addition, the Yaksha must be completely trusted, unlike the SEM, which is partially trusted.

Another scheme which is based on two-party RSA is S-RSA [26]. This scheme is proposed by MacKenzie and Reiter [26] and is used to guard password-protected private keys from offline dictionary attacks on a network device captured by an adversary. Like the Yaksha system, a private key of a user is shared between him and a server. The user's share of the private key is a function of his password, while the server's share of the private key is encrypted within a token and stored in the network device using the server's public key. The sum of the two partial keys forms the complete private key of the user. When a user wants to issue a signature, the device sends the token to the server, then the server extracts his partial private key from the token to help the user to issue his signature. Boneh and Franklin [28] provided an algorithm to share an RSA key generation function between two users. Nicolosi et al. [29] designed a proactive two-party signature for user authentication. MacKenzie and Reiter [30] developed a two-party DSA signature scheme which was provably secure.

III. The ID-MRSA

the ID-MRSA is described as follows. In the setup phase, PKG generates two safe primes p, q, then calculates $n = pq$. He keeps p, q as secret system parameters, while publish the modulus n to the users. After that, PKG generates the private key for user A by hashing his identity to a value $KG()$, then the PKG pads $KG()$ with one to generate an odd public key for user A. After that, he generates the corresponding full RSA private key for user A, then splits the user A private key between user A and the SEM. If user B wants to encrypt message m to user A, he encrypts it normally using user A's public key. After receiving the message from user B, user A sends it to the SEM to partially decrypt it. If user A is revoked, the SEM refuses to decrypt the message and returns 'error'. If user A is not revoked, the SEM partially decrypts the message and sends it to user A. After getting the partially decrypted message from the SEM, user A generates his own partially decrypted version of the message and then combines it with the SEM's partially decrypted message to get his fully decrypted message. The algorithms of key generation, encryption and decryption are shown as follows.

IV. The ID-MRSA Security

Based on [31] and [8], the ID-MRSA is secure in the random oracle model. However, there is a special attack that an insider user can attempt. He can manipulate the encrypted message so that it can be decrypted using his private key.

```
Key Generation:
Input: two safe primes $p$ and $q$
Output: $d_u$, $d_{SEM}$
$n = pq$ (Generating the modulus)
for user do
    $s = k - |KG()| - 1$
    $e = 0^s||KG()||1$ (Padding the hashed identity with one)
    $d = \frac{1}{e}$ mod$(\varphi(n))$ (Calculating the private key $d$)
    $d_u \overset{r}{\leftarrow} Z_n - [0]$ (Choosing randomly an element $d_{user,u}$ from $Z_n - [0]$)
    $d_{SEM} = (d - d_u)$ mod$(\varphi(n))$
end
```

```
Encryption:
Input: $n, k, KG()$
Output: $C$
$s = k - |KG()| - 1$
$e = 0^s||KG()||1$
$C =$ Encrypt the message using RSA/OAEP
```

This can be done by finding a mapping function $f(C_A) = C_B$.

Lemma 1: Assuming that there are two users, user A and user B, user B is able to find a mapping function $f(C_A) = C_B$ and hence, decrypt/forge the encrypted message/sign a message of user A iff $e_a|e_b$.

The proof of this lemma can be found in [8]. If $e_a|e_b$ i.e. $e_b = k \times e_a$, we can construct a mapping function f such that $f(a) = a^k$ mod (n). To solve this problem, the user's public key cannot be a factor of the product of the other users' public keys. To ensure that, Ding and Tsudik

```
Decryption:
Input: $C$, $d_u$, $d_{SEM}$
Output: $m$
for SEM do
    if user Revoked then
        return (ERROR)
    end
end
$PD_{SEM} = c^{d_{SEM}}$ mod$(n)$ (Calculate the partially decrypted message of the SEM)
$SendPD_{SEM}$to the user
for user do
    $PD_u = c^{d_u}$ mod$(n)$ (Calculate the partially decrypted message of user)
    $M = (PD_{SEM} \times PD_u)$ mod$(n)$ (Decrypt the message)
end
$m =$ OAEP Decoding of M
```

The digital signature scheme is shown below:

```
**Signing:**
**Input**: $m, d_u, d_{SEM}$
**Output**: $h, S$
$h = H(m)$
**for** *SEM* **do**
    **if** *user Revoked* **then**
        return (ERROR)
    **end**
**end**
$PD_{SEM} = h^{d_{SEM}} \bmod(n)$
Send $PD_{SEM}$ to the user
**for** *user* **do**
    $PD_u = c^{d_u} \bmod(n)$
    $S = (PD_{SEM} \times PD_u) \bmod(n)$
**end**
$S =$ OAEP Decoding of S
```

```
**Verification:**
**Input**: $h, S, n, k, KG()$
**Output**: $\overline{h}$
$s = k - |KG()| - 1$
$e = 0^s||KG()||1$
$\overline{h} = S^e \bmod(n)$
**if** $h \neq \overline{h}$ **then**
    return (ERROR)
**end**
```

used a division intractable hash function to map a user's identity to his public key ($KG()$). This notion of division intractable hash functions was proposed by Gennaro et al.[32]. A hash function $H()$ is said to be division intractable if it is unfeasible to find a set of values $(X_1, X_2, ..., X_n, Y)$, such that $H(Y)|\prod_i(H(X_i))$. Based on that, the authors of [8] stated that, '*IB-mRSA/OAEP encryption offers equivalent the semantic security to RSA/OAEP against adaptive chosen ciphertext attacks in the random oracle model, if the key generation function is division intractable*'. In this section, we prove that Ding and Tsudik's claim is wrong. Using a division intractable hash function does not necessarily generate division intractable public keys because the output of the hash function $KG()$ is padded with a 'one'. The public key is $e = KG()||1$ [8] or $e = KG()||00000001$ [9]. This means that $e = 2KG() + 1$ as in [8] or $e = 8KG()+1$ as in [9]. This multiplication and addition change completely the property of the public key and it is likely, with overwhelming probability, to lose its property of being division intractable. For example, if $|KG(ID_1)| = 6$ and $|KG(ID_2)| = 19$, these two values are division intractable, but if we calculate $e_1 = 2|KG(ID_1)| + 1 = 2 \times 6 + 1 = 13$ and $e_2 = 2|KG(ID_2)|+1 = 2\times19+1 = 39$, we can see that e_1 and e_2 are no longer division intractable ($e_2 = 3e_1$) and consequently, lemma 1 can be used to attack the ID-MRSA although the used hash function is division intractable. Real life values that represent the same idea are shown in table 1. These numbers are in hexadecimal.

Table I
EXAMPLE OF AN ATTACK ON THE ID-MRSA IN REAL WORLD

Variables	Value
$\|KG(ID_1)\|$	$A07B0C7AFE0A33D7A270D8A35B995B3546D77D6E$
$\|KG(ID_2)\|$	$808288FE7D6E2B83AD145D7AD059CE09A9BA8F717C$
e_1	$140F618F5FC1467AF44E1B146B732B66A8DAEFADD$
e_2	$1010511FCFADC57075A28BAF5A0B39C1353751EE2F9$
e_2/e_1	CD

In the following subsections, we show how this simple notice can be used by an insider one-wayness adversary to attack the ID-MRSA. The first attack is a direct application of lemma 1. The second attack is a common modulus attack against the ID-MRSA. For the signature scheme, we prove that, if such a mapping function exists, an insider attacker can forge the signature of another user without knowing his private key.

A. Attacks on the ID-MRSA Encryption scheme

The first attack applies when the effect of using an intractable hash function is canceled by padding the output with one and hence, the resulting public keys are in the form of ($e_B = k \times e_A$). If this happens, user B can decrypt the message of user A using the following formula:

$$C_B = C_A^{e_B/e_A} \bmod (n)$$

and then decrypt this message using his private key. This vulnerability can be used by an insider adversary user B to attack an encrypted message of user A. This attack is done as follows:

- user B chooses an identity ID_B such that $e_B = k \times e_A$, where k is an integer.
- At the challenge phase, user B sends to the challenger any two messages m_1 and m_2 and the identity ID_A.
- The challenger will toss a fair coin $b \in [0, 1]$ and will send $C \leftarrow Enc(m_b)$ to user B.
- user B then calculates $C_B = C_A^{e_B/e_A} \bmod (n)$.
- user B sends C_B to the SEM for decryption.
- After decryption, user B can successfully find $b' = b$.

The gravity of this attack is making the ID-MRSA vulnerable against one-wayness adversary; not only can user B distinguish between two messages m_1 and m_2, he can decrypt it as a message of his own. To illustrate this attack, we pick some toy examples. w.l.g, this attack can be applied in real time values. In this scenario, we assume that there are three users using this cryptosystem: user A, user B and user C with

$KG(ID_A) = 7$, $KG(ID_B) = 22$ and $KG(ID_C) = 6$. We can see that these values are division intractable. But if we calculate their public key $e_A = 2KG(ID_A) + 1 = 2 \times 7 + 1 = 15$, $e_B = 2 \times KG(ID_B) + 1 = 2 \times 22 + 1 = 45$ and $e_c = 2KG(ID_C) + 1 = 2 \times 6 + 1 = 13$. We can see that $e_B = 3e_A$. In the following, we will show that user B can convert the encrypted message of user A to an encrypted version of his own and let the SEM decrypt it for him.

Assume that the two primes were $p = 23$ and $q = 47$, the modulus is $n = p \times q = 47 \times 23 = 1081$. If message $m = 12$ is encrypted to $C_A = m^{e_A} \bmod n = 12^{15} \bmod 1081 = 864$ and sent to user A. User B takes a copy of C_A and computes $C_B = C_A^{e_B/e_A} \bmod n = 864^3 \bmod 1081 = 380$. If the same message was sent to user B, then $C_B = m^{e_B} \bmod n = 12^{45} \bmod 1081 = 380$. This means that user B successfully converted the encrypted message of user A to an encrypted version of his own and he can now decrypt this message without the need to know the secret key of user A.

There is a second attack in which if the same message was sent to two users, user A and user B, user C with public key satisfies $gcd(e_A, e_B) | e_C$ can decrypt this message by using the following attack:

- Assuming that $g = gcd(e_A, e_B) | e_C$, user C finds the values of a and b such that $a \times e_A + b \times e_B = g$ using extended euclidian algorithm.
- After obtaining a and b, user C calculates $C_g = C_A^a \times C_B^b \bmod (n) = m^{ae_A + be_B} \bmod (n) = m^g \bmod (n)$
- From C_g, user C can obtain his version of m as follows:

$$\begin{aligned} C_c &= C_g^{e_c/g} \bmod (n) \\ &= m^{ge_c/g} \bmod (n) \\ &= m^{e_c} \bmod (n) \end{aligned}$$

and then he can decrypt it using his private key. We use another toy example to illustrate this attack. Assume that three users, user A, user B and user C, have the following outputs of the hash function: $KG(ID_A) = 25$, $KG(ID_B) = 7$ and $KG(ID_C) = 13$. We can see that these values are accepted values for a division intractable hash function. Calculating their public keys: $e_A = 2KG(ID_A) + 1 = 2 \times 25 + 1 = 51$, $e_B = 2KG(ID_B) + 1 = 2 \times 7 + 1 = 15$ and $e_c = 2KG(ID_C) + 1 = 2 \times 13 + 1 = 27$. We can also see that these values are division intractable; $(e_B e_C) \nmid e_A$. We will assume that $m = 12$ and as in the previous scenario, $n = 1081$, then $C_A = m^{e_A} \bmod n = 12^{51} \bmod 1081 = 108$ and $C_B = m^{e_B} \bmod n = 12^{15} \bmod 1081 = 864$. For user C to attack this message, he first calculates $g = gcd(e_A, e_B) = gcd(51, 15) = 3$ and then he finds, using the extended euclidean algorithm, two values a and b such that $ae_A + be_B = g$. In this scenario, $a = -2,\ b = 7$. Then he gets $C_g = m^g \bmod n = m^{ae_A + be_B} \bmod n = C_g = C_A^a \times C_B^b \bmod (n) = 12^3 \bmod 1081 = 108^{-2} \times 864^7 \bmod 1081 = 647$. Then he finally obtains $C_C = C_g^{e_C/g} \bmod n = m^{e_C} \bmod n = 647^{27/3} \bmod 1081 = 12^{27} \bmod 1081 = 432$ which represents the original message m encrypted in his own key. This type of attack can be dealt with using OAEP. The probability that two messages are padded using the same random padding is negligible (about 2^{-160}). So if the same message were encrypted twice using different OAEP padding, then $C_g \neq C_A^a C_B^b \bmod n$ and the attack fails. So the advice is do not encrypt the same message with the same padding to different users.

B. Attack on the ID-MRSA signature scheme

In this subsection, we present an attack on the ID-MRSA signature scheme even with division intractable hash function. We assume that there are two users, user A and user B, and show that user B can forge the signature of user A without knowing the private key of user A using the following steps, as long as a mapping function between their public keys exists:

- user B signs the message m with the SEM using his private key.
- After obtaining his signed message (m_B), he calculates the forged signature of user A: $\overline{m}_A = m_B^k \bmod n$, where $k = e_B/e_A$.
- $\overline{m}_A$ can be verified using the public key of user A.

The proof of the correctness of this attack is described as follows:

$$\begin{aligned} e^b h_b &= 1 \bmod \varphi(n) \\ e^b &= ke^a \\ ke^a h_b &= 1 \bmod \varphi(n) \\ e^a(kh_b) &= 1 \bmod \varphi(n) \\ e^a \overline{h}_a &= 1 \bmod \varphi(n) \end{aligned}$$

We now give a toy example of this attack using the same parameters of the encryption scenario: $KG(ID_A) = 6$, $KG(ID_B) = 19$, $p = 23$, $q = 47$, $n = 1081$, $\varphi(n) = 1012$, $e_A = 2 \times KG(ID_A) + 1 = 2 \times 6 + 1 = 13$ and $e_B = 2 \times KG(ID_B) + 1 = 2 \times 19 + 1 = 39$. Using the extended euclidean algorithm, we can find that $h_a = 545$ and $h_b = 519$. User B will work with the SEM to sign his message, $m = 12$, the signed message will be $m_B = m^{h_b} \bmod (n) = 12^{519} \bmod 1081 = 6$. After that, he will generate a forged signature of user A by calculating $\overline{m}_A = m_B^{e_B/e_A} \bmod (n) = 6^3 \bmod 1081 = 216$, then he will send(12, 216) as a forged signature of user A. If user C wants to verify this message, he will calculate $m = \overline{m}_A^{e_A} \bmod (n) = 216^{13} \bmod 1081 = 12$, which is the same as the sent message.

V. The ID-MRSA-V2

After reviewing the security flaw of the ID-MRSA encryption/signature schemes, we present two solutions that correctly make the ID-MRSA secure against these types of attacks. We denote the ID-MRSA with these solutions

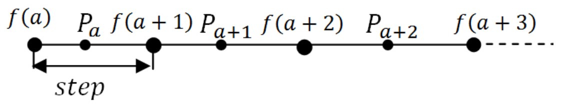

Figure 1. The distribution of primes

applied as the ID-MRSA-V2. These solutions must satisfy the following conditions:

- There is a deterministic one-to-one mapping function that maps the identities of the users to their public keys.
- This function must be division intractable.
- The generated public keys must be co-prime with $\varphi(n)$.

The first solution ensures that the maximum public key is less than three times the smallest public key, i.e. $e_M < 3e_m$. The subscript M denotes maximum while the subscript m denotes minimum. One can see that this completely eliminates the problem. To achieve this goal, the relation between the hash function of the maximum and minimum public keys must be:

$$\begin{array}{rcl} e_M & < & 3e_m \\ 2|KG_M|+1 & < & 3(2|KG_m|+1) \\ 2|KG_M|+1 & < & 6|KG_m|+3 \\ 2|KG_M| & < & 6|KG_m|+2 \\ |KG_M| & < & 3|KG_m|+1 \end{array}$$

If the inequality $|KG_M| < 3|KG_M| + 1$ holds, then all public keys are division intractable. The only disadvantage of this solution is that it limits the space of the hash function. Another solution to fix this security flaw is mapping the users' identities to public keys that are primes. Hence, the public keys will be division intractable. To generate primes from identities, we first use a collision resistance hash function, $a = H(ID)$ and then apply the following function:

$$f(a) = (a-1) \times \text{step} + 1.$$

where $step$ is a value used to generate unique primes. After that, we check if $f(a)$ is a prime. If it is, then $e = f(a)$. If not, find the next smallest prime larger than $f(a)$. The algorithm is shown as follows.

```
$a = H(ID)$
$f(a) = (a-1) \times \text{step} + 1$
**if** $f(a)$ *is not prime* **then**
    $f(a)$ =NxPrime($f(a)$)
**end**
return ($f(a)$)
```

where $NxPrime(x)$ is a function that finds the smallest prime larger than x.

This function must satisfy the following conditions to perform correctly:

- The hash function must be collision resistant: it is unfeasible to find two different values X, Y such that $a = H(Y) = H(X)$. This guarantees that each identity is mapped to a unique public key.
- The value of step is as follows. The step value must be chosen carefully such that $P_a < f(a+1)$ for any value a. This will guarantee that each identity will be mapped to a unique prime. Fig (1) shows this idea. The value of step can be determined by finding a value greater than the maximal prime gap, which is the gap larger than the gaps of smaller primes. For primes less than 2^{40}, a value of step greater than 1476 can be safely used [33].
- If the mapping function satisfies the above conditions, it will resist the first attack to the encryption scheme because we can guarantee that the generated public keys are primes and primes satisfy the division intractable property. On the other hand, however, it cannot withstand the second attack because since all the public keys are primes, their greatest common divisor (gcd) is one, and anyone can recover the message without knowing the secret key. The only solution for this attack is not to use the same padding in OAEP when encrypting the same message to multiple users. For the signature schemes, since the public keys are division intractable, there is no relation between their private keys and such an attack will fail.

  After fixing these drawbacks, the ID-MRSA can be proven CCA2 secure in the random oracle model using the same methodology explained in [8] or [31].

VI. Implementation

the ID-MRSA-V2 was implemented using MIRACL software C library and its performance was compared with the ID-MRSA and RSA. The PC that was used to run these tests has a processor Intel(R) Core(TM) i5-2410M CPU @ 2.30GHz (4 CPUs),and 4096MB RAM. Table 2 shows the test results. The results are in ms.

From these results, we can see that:

- $NxPrime$ does not affect the performance of the ID-MRSA, because the gaps between consecutive prime numbers are known to be quite small[33].
- The results of the key generation of RSA are larger than those of the ID-MRSA and the ID-MRSA-V2, because the key generation of the ID-MRSA and the ID-MRSA-V2 is for each user, so it does not involve the prime key generation that exists in RSA key generation.
- The encryption time increases slightly with the key length, so the key length is not problematic. This can be seen also in the encryption times of the ID-MRSA and IDMRSA-V2: although the key of the ID-MRSA-V2 is larger than that of the ID-MRSA by the value of step, the times are almost the same.

Table II
THE TIME RESULTS

The Process	Modulus	Key Size	RSA	the ID-MRSA	the ID-MRSA-V2
Key Generation	1024 Bits	16 Bits	17.19	0.13	0.11
		128 Bits	22.04	0.13	0.13
		160 Bits	19.8	0.14	0.14
	2048 Bits	16 Bits	128.26	0.17	0.16
		128 Bits	130.26	0.14	0.14
		160 Bits	127.86	0.16	0.16
Encryption / Verify	1024 Bits	16 Bits	0.03	0.06	0.05
		128 Bits	0.03	0.03	0.05
		160 Bits	0.03	0.05	0.03
	2048 Bits	16 Bits	0.03	0.06	0.06
		128 Bits	0.01	0.06	0.05
		160 Bits	0.03	0.06	0.06
Decryption / Sign	1024 Bits	16 Bits	0.14	0.12	0.14
		128 Bits	0.13	0.13	0.14
		160 Bits	0.14	0.13	0.13
	2048 Bits	16 Bits	0.22	0.22	0.22
		128 Bits	0.23	0.23	0.23
		160 Bits	0.22	0.22	0.22

- The decryption times are longer than the encryption time in all schemes. This drawback is actually inherited from RSA, because the decryption keys are extremely large (of the length of n).
- The times consumed by all these schemes are proportional to the modulus size.

VII. CONCLUSION

In this paper, we found some security issues of the ID-MRSA. We showed that using a division intractable hash function does not necessarly guarantee that the generated public keys are also division intractable. Consequently, the cryptosystem may not be secure even if the hash function used is division intractable. We proposed two solutions to overcome this drawback. After applying these modifications, the ID-MRSA is secure in the random oracle model if the mapping function parameters have been chosen correctly.

VIII. ACKNOWLEDGEMENT

The authors gratefully acknowledge the help of Dr. Madeleine Strong Cincotta in the final language editing of this paper.

REFERENCES

[1] A. Shamir, "Identity-based cryptosystems and signature schemes." in *Advances in Cryptology CRYPTO 84*, 1985.

[2] N. Howgrave-Graham and J.-P. Seifert, "Extending wieners attack in the presence of many decrypting exponents," in *Secure Networking CQRE99*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1999, vol. 1740, pp. 153–166. [Online]. Available: http://dx.doi.org/10.1007/3-540-46701-7_14

[3] S. Sarkar and S. Maitra, "Cryptanalysis of rsa with more than one decryption exponent," *Information Processing Letters*, vol. 110, no. 89, pp. 336 – 340, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020019010000505

[4] K. O. R. Sakai and M. Kasahara., "Cryptosystems based on pairing." in *Symposium on Cryptography and Information Security (SCIS 2000), Japan*, 2000.

[5] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Advances in Cryptology CRYPTO 2001*.

[6] C. Cocks, "An identity based encryption scheme based on quadratic residues," in *Int. Conf., of Lecture Notes in Computer Science*, 2001.

[7] D. Boneh, X. Ding, G. Tsudik, and C. M. Wong, "A method for fast revocation of public key certificates and security capabilities," in *Proceedings of the 10th conference on USENIX Security Symposium - Volume 10*, ser. SSYM'01. Berkeley, CA, USA: USENIX Association, 2001, pp. 22–22. [Online]. Available: http://dl.acm.org/citation.cfm?id=1251327.1251349

[8] X. Ding and G. Tsudik, "Simple identity-based cryptography with mediated rsa," in *RSA, Proceedings CT-RSA 2003, LNCS 2612, Springer-Verlag 2003*, 2003.

[9] D. Boneh, X. Ding, and G. Tsudik, "Identity-based mediated rsa," in *Dow Jones & Company, Inc*, 2002.

[10] M. Bellare and P. Rogaway, "Optimal asymmetric encryption how to encrypt with rsa." in *Advances in Cryptology EUROCRYPT 94*, 1994.

[11] W. F. a. D. S. R. Housley, W. Polk, *RFC3280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, Network Working Group Std., April 2002. [Online]. Available: http://tools.ietf.org/pdf/rfc3280.pdf

[12] S. F. S. B. R. H. D. Cooper, S. Santesson and W. Polk, *RFC5280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, Network Working Group Std., May 2008.

[13] D. Boneh, X. Ding, and G. Tsudik, "Fine-grained control of security capabilities," *ACM Trans. Internet Technol.*, vol. 4, no. 1, pp. 60–82, Feb. 2004. [Online]. Available: http://doi.acm.org/10.1145/967030.967033

[14] A. M. S. G. a. C. A. M. Myers, R. Ankney, *RFC 2560: Internet public key infrastructure online certificate status protocol - OCSP*, Std.

[15] P. Kocher, "On certificate revocation and validation," in *Financial Cryptography*, ser. Lecture Notes in Computer Science, R. Hirchfeld, Ed. Springer Berlin / Heidelberg, 1998, vol. 1465, pp. 172–177, 10.1007/BFb0055481. [Online]. Available: http://dx.doi.org/10.1007/BFb0055481

[16] M. Naor and K. Nissim, "Certificate revocation and certificate update," in *Proceedings of the 7th conference on USENIX Security Symposium - Volume 7*, ser. SSYM'98. Berkeley, CA, USA: USENIX Association, 1998, pp. 17–17. [Online]. Available: http://dl.acm.org/citation.cfm?id=1267549.1267566

[17] M. T. Goodrich, R. Tamassia, and A. Schwerin, "Implementation of an authenticated dictionary with skip lists and commutative hashing," *DARPA Information Survivability Conference and Exposition,*, vol. 2, p. 1068, 2001.

[18] H. S. J. et al, "Effcient revocation of security capability in certificateless public key cryptography," in *Proceedings of KES 2005 (LNAI 3682), Sep 14-16, 2005*, 2005.

[19] a. W. X.-m. Y. Chen, M. Wen-ping, "Secure mediated certificateless signature scheme," *THE JOURNAL OF CHINA UNIVERSITIES OF POSTS AND TELECOMMUNICATIONS*, vol. Volume 14, Issue 2,, pp. 75–78, June 2007.

[20] F. W. C. Yang and X. Wang, "Efficient mediated certificateless public-key encryption scheme without pairings," in *21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07)*, 2007.

[21] C. D. T. Candebat and D. T. Gray, "Pseudonym management using mediated identity-based cryptography," in *Proceedings of the 2005 workshop on Digital identity management*, 2005.

[22] S. N. P. H. L. Ibraimi, M. Petkovic and W. Jonker, "Mediated ciphertext-policy attribute-based encryption and its application," in *Information Security Applications, Lecture Notes in Computer Science, Springer Verlag*, 2009.

[23] L. G. Xi. Cheng and X. Wang, "An identity-based mediated signature scheme from bilinear pairing," *International Journal of Network Security,*, vol. Vol.2, No.1, pp. 29–33, 2006.

[24] C. B. Sherman S.M. Chow and J. M. G. Nieto, "Security-mediated certificateless cryptography," *Public Key Cryptology- PKC 2006 (LNCS)*, vol. 3958, pp. 508–524, 2006.

[25] R. Ganesan, "The yaksha security system," *Commun. ACM*, vol. 39, no. 3, pp. 55–60, Mar. 1996. [Online]. Available: http://doi.acm.org/10.1145/227234.227242

[26] P. MacKenzie and M. K. Reiter, "Delegation of cryptographic servers for capture-resilient devices," in *Proceedings of the 8th ACM conference on Computer and Communications Security*, ser. CCS '01. New York, NY, USA: ACM, 2001, pp. 10–19. [Online]. Available: http://doi.acm.org/10.1145/501983.501986

[27] MacKenzie, Philip and Reiter, Michael K., "Networked cryptographic devices resilient to capture," in *Proceedings of the 2001 IEEE Symposium on Security and Privacy*, ser. SP '01. Washington, DC, USA: IEEE Computer Society, 2001, p. 12. [Online]. Available: http://dl.acm.org/citation.cfm?id=882495.884440

[28] D. Boneh and M. Franklin, "Efficient generation of shared rsa keys," *J. ACM*, vol. 48, no. 4, pp. 702–722, Jul. 2001. [Online]. Available: http://doi.acm.org/10.1145/502090.502094

[29] A. Nicolosi, M. Krohn, Y. Dodis, and D. Mazieres, "Proactive two-party signatures for user authentication," in *Proc. 10th Annual Network and Distributed System Security Symposium (NDSS03), The Internet Society*, 2003, pp. 233–248.

[30] P. MacKenzie and M. K. Reiter, "Two-party generation of dsa signatures," 2004.

[31] B. Libert and J.-J. Quisquater, "Efficient revocation and threshold pairing based cryptosystems," in *Proceedings of the twenty-second annual symposium on Principles of distributed computing*, ser. PODC '03. New York, NY, USA: ACM, 2003, pp. 163–171. [Online]. Available: http://doi.acm.org/10.1145/872035.872059

[32] S. H. R. Gennaro and T. Rabin, "Secure hash-and-sign signatures without the random oracle," in *Advances in Cryptology EUROCRYPT 99,*, 1999.

[33] J. K. Andersen. Maximal prime gap. [Online]. Available: http://users.cybercity.dk/~dsl522332/math/primegaps/maximal.htm