EAI Endorsed Transactions
on Security and Safety

Research Article ICST.ORG

Mediated Encryption: Analysis and Design*

I. Elashry, Y. Mu and W. Susilo

University of Wollongong, Wollongong, Australia 2500

Abstract

Boneh, Ding and Tsudik presented identity-based mediated RSA encryption and signature systems in which the users are not allowed to decrypt/sign messages without the authorisation of a security mediator. We show that ID-MRSA is not secure and we present a secure modified version of it which is as efficient as the original system. We also propose a generic mediated encryption that translates any identity based encryption to a mediated version of this IBE. It envelops an IBE encrypted message using a user's identity into an IBE envelope using the identity of the SEM. We present two security models based on the role of the adversary whether it is a revoked user or a hacked SEM. We prove that GME is as secure as the SEM's IBE against a revoked user and as secure as the user's IBE against a hacked SEM. We also present two implementations of GME based on Boneh-Franklin FullIBE system which is a pairing-based system and Boneh, Gentry and Hamburg (BGH) system which is pairing-free system.

Keywords: Mediated Encryption, Key Revocation Problem, Identity-based Encryption

Received on 14 February 2014; accepted on 23 December 2014, published on 30 January 2015

doi:10.4108/sesa.2.3.e2

1. Introduction

For the last few years, the key revocation problem has received the attention of the cryptography community because the user's public key cannot be used if the corresponding private key is compromised. This problem occurs in public key cryptography because it depends on digital certificates. Digital certificates are signatures issued by a trusted certificate authority (CA) that securely ties together a number of quantities. Typically, these quantities contain at least the ID of a user (U) and its public key (PK). Frequently, the CA comprises a serial number (SN) for managing certificates. The CA also binds the certificates to an issue date D_1 and an expiration date D_2. By issuing the signature of $SigCA(U, PK, SN, D_1, D_2)$, the CA provides PK between the current date D_1 and the future date D_2.

A user's public key may have to be revoked before its expiration date D_2 if a user's secret key is accidentally leaked or an attacker successfully compromises it. A new key pair should be generated and the corresponding certificate should be issued.

If the CA can revoke a certificate, then third parties cannot depend on this certificate unless the CA shares certificate status information indicating whether this certificate is still valid. This certificate status information has to be recently generated and must be widely distributed. Sharing a great deal of fresh certificates periodically leads to the key revocation problem which consumes large amount of computation power and bandwidth. This is considered a hindrance to global application of public-key cryptography.

1.1. Some Previous Solutions to the Key Revocation Problem

The most widely-known and a very ineffective way to solve the key revocation problem is the certificate revocation list (CRL) [10, 23] which is a list that contains revoked certificates. The CA produces this list periodically with its signature. Because the CA will probably revoke many of its certificates -say 10 %- if they are produced for a validity time of one year [14, 20], the CRL will be too lengthy if the CA has many clients. Moreover, the complete CRL must be sent to any party that needs to carry out a certificate status check. There are improvements to this approach such as delta CRLs [5] which lists only those certificates revoked since the CA's last update. But the consumed transmission bandwidth and computation costs of the transmission of these lists are still very high. Another method of solving the key revocation problem is the

*This paper is an extended version of the paper entitled 'Generic Mediated Encryption' in Securecomm 2013.

online certificate status protocol (OCSP) [18]. If a client wants to check certificate status, he sends to the CA a certificate status query. The CA replies to this query by producing a fresh signature on the certificate's current status. This omits the need to send a list of all revoked certificates and reduces the transmission costs to a single signature per query but it significantly increases computation costs. It also negatively affects security. If the CA is centralised, the system will have a single point of failure and consequently will become highly vulnerable to denial-of-service (DoS) attacks [14, 20].

Kocher [17] suggested an improved version of OCSP called certificate revocation trees (CRTs). The CA can be considered as a global service provider and must be replicated using many servers in order to stand the entire load of certificate validation requests. The CA's signing key must be distributed securely over many servers. This process is expensive and insecure. A solution to this problem is that a highly secure root CA sends a signed CRL-like data structure to other less-secure servers and 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 CA. This structure is called a certificate revocation tree (CRT). If a user wants to check the validity of a certificate, he sends a request to the nearest less-secure CA 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. 2-3 trees proposed by Naor and Nissim [21], Aiello-Lodha-Ostrovsky [1] and skip-lists proposed by Goodrich [16] are two proposed solutions to this problem.

Micali [14, 19, 20] proposed a promising way to solve this problem. (See also [1, 12, 22].) Similar to previous PKI proposals, Micali's Novomodo system includes a CA, one or more directories (to distribute the certification information) and the users. Despite this similarity, it is more efficient than CRLs and OCSP without sacrificing security.

The advantage of Novomodo over a CRL-based system is that a directory's reply to a certificate status query is brief, only 160 bits per query (if T has cached $SigCA(U, PK, SN, D_1, D_2, X_n)$). On the other hand, the length of a CRL increases with the number of certificates that have been revoked (i.e. number of clients). Novomodo has several advantages over OCSP. First, Novomodo depends on hashing while OCSP depends on signing. Because hashing has lower computation costs than signing, the CA's computational costs in Novomodo is typically much lower. Second, the directories in Novomodo do not have to be trusted unlike the distributed components of an OCSP CA. Instead of issuing signatures depending on third parties, the directories publish only hashed pre-images sent by the CA (which cannot be produced by Novomodo directories). Third, the directories do not perform any online computation and make Novomodo less vulnerable to DoS attacks. Finally, although OCSP does not consume too much bandwidth, Novomodo's bandwidth consumption is typically even lower since public-key signatures are typically longer than 160 bits (length of X_{n-i} sent per query).

A disadvantage of all the above techniques is relaying on third-party queries [14]. It is preferable to eliminate third-party queries for several reasons. First, since anyone can ask for third-party queries, each certificate server must be able to get the certificate status of every client in the system. The situation is much simpler if third-party queries are eliminated. Each server is only required to have certification proofs for the clients that it works for. In addition, multi-cast can be used to push certificate proofs to clients to reduce the transmission costs. Second, third-party queries multiply the query computation costs of the CA and/or its servers. For example, if each client queries the certificate status of X clients per day, the system must process XN queries (where N is the number of clients). Third, from a business model perspective, non-client queries are not recommended because if T is not a client of the CA, he will not be motivated to deliver T fresh certificate status information. Finally, since the CA must reply to queries from non-clients, it becomes more vulnerable to DoS attacks and this is a security concern. In summary, removing third-party queries leads to a reduction in infrastructure costs, simplifies the business model and increases security. We can completely remove third-party queries by using an implicit certification such as identity-based encryption (IBE).

The notion of identity-based cryptography was put forth by Shamir [25]. In the same paper, Shamir also proposed a concrete construction of an identity-based signature system. 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 not identity-based encryption since sharing a common modulus between different users makes

RSA insecure. Examples of RSA cryptanalysis with the same modulus used for different encryption/decryption pairs are [3, 26]. Sixteen years later, Sakai, Ohgishi and Kasahara [24] proposed the first identity-based cryptography and independently Boneh and Franklin [7] proposed the first reliable and provable identity-based cryptography based on Weil pairings over elliptic curves. Cocks [9] presented a system that is based on factorisation of a composite integer. These cryptosystems opened a new era in cryptography.

Gentry presented the notion of certificate-based encryption (CBE) [14]. This system combines public-key encryption (PKE) and IBE while keeping most of the advantages of each. Using PKE, each client creates its own public-key/secret-key pair and asks for a certificate from the CA. The CA uses an IBE system to create the certificate. This certificate has all of the functionality of a conventional PKI certificate as well as a decryption key. This double encryption gives us implicit certification. If T wants to encrypt a message, it double encrypts it using PKI and IBE and then the decryptor uses both his secret key and an up-to-date certificate from his CA to decrypt the message. CBE has no escrow (since the CA does not know the user's secret key) and it does not a have secret key distribution problem because the CA's certificate needs not be kept secret. Although CBE consumes less computation and transmission costs than Novomodo, it is preferable to completely eliminate the use of certificates to preserve the infrastructure costs.

Boneh, Ding, Tsudik and Wong were the first to introduce the notion of mediated cryptosystems in [6]. 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 the first to propose a secure variant of identity-based RSA that shares a common modulus between different users. Their system is based on the so-called security mediator (SEM) architecture in which the SEM is a semi-trusted server. If an employee wants to decrypt/sign a message, he must co-operate with the SEM to do so. The idea behind their system is splitting the secret key of an employee between the employee himself and the SEM. Hence, without the SEM cooperation, the employee cannot sign or encrypt messages. This is also helpful to monitor the security of sent/received secure messages in the company. Later on, Ding and Tsudik presented a security proof for these systems. 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.

The SEM architecture was proven useful [6] to simplify signature validation and enable key revocation in legacy systems. Although this system does not require a CA to create a certificate or send certificate status information and the computation and transmission costs are kept to minimum, it has two major security concerns. First, There is a security flaw in [11, 15]. Second, since SEM is centralised, it represents a single point of failure for the system and hence the system is vulnerable to DOS attacks. Moreover, a hacked SEM can be a major threat to the system security because the SEM is a semi-trusted server.

1.2. Our Contribution

First, we investigate the mediated encryption [6, 11] by reviewing the security of the ID-MRSA. 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. Second, we take the work of [6] one step further and present a generic mediation system that is capable of making any IBE system support key revocation. This idea is based on a letter-envelope technique. If U_A wants to encrypt a message to U_B, he first encrypts it normally using U_B's identity (letter) then he encrypts the letter again using SEM identity (envelope) and sends the resulted ciphertext to U_B. To decrypts the ciphertext, U_B sends the message to the SEM. If U_B is revoked, the SEM will not open the envelope for him. If U_B is not revoked, the SEM will open the envelope and send the letter to U_B who decrypts the message using his private key. The structure of our system combines the advantages of both Gentry [14] and Boneh *et al.* [6]. It completely eliminates the use of certificates. In addition, the SEM in our system is not a single point of failure. If the SEM is compromised, the system can continue working using the user's IBE system. In addition, all messages sent to the SEM before or after an attack are safe and secure. Through the paper, U represents the user, S represents the SEM, P represents the system parameters, Gen represents the

setup algorithm, KG is the Key Generation Algorithm, Enc is the encryption algorithm, Dec is the decryption algorithm and r is the private key.

The rest of the paper is organised as follows: Sec. 2 discusses the ID-MRSA encryption/signature systems and their implementations. Sec. 3 discusses the security flaw of the ID-MRSA. Sec. 4 proposes two solutions to overcome the ID-MRSA security flaw. The effect of using these solutions on the ID-MRSA are discussed in Sec. 5. Sec. 6 presents the generic mediated encryption (GME) and its security proof. Sec. 7 presents two implementations of GME, the first one is based on the BF IBE system [7] which is a pairing-based system and the second one is based on the BGH system [8] which is not a pairing-based system. The last section presents the conclusions of the paper.

2. The ID-MRSA

We review the structure of ID-MRSA as follows. In the setup phase, PKG produces two safe primes p, q then computes $n = pq$. He preserves p, q as secret system parameters while makes the modulus n public. Next, PKG produces the private key for U_A by hashing his identity to a value $KG()$ then the PKG pads $KG()$ with one to get an odd public key for U_A. After that, he makes the corresponding full RSA private key for U_A and splits it between U_A and the SEM. U_B encrypts message m to U_A normally using the public key of U_A. After getting the encrypted message C from U_B, U_A directs it to the SEM to partially decrypt it. If U_A is revoked, the SEM declines to decrypt the message and returns 'error'. Otherwise, the SEM partially decrypts the message to get PD_S and sends it to U_A. After receiving the partially decrypted message PD_S from the SEM, U_A computes his own partially decrypted version of the message PD_U 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 below. The signature system has the same key generation as the encryption system. When U_A signs a message to U_B, he sends it to the SEM to partially sign the message for him if he is not revoked. U_A combines the partially signed message of the SEM with his partially signed version of the message to get his own signature. U_B can verify the signature of U_A normally as RSA.

Key Generation:
Input: two safe primes p and q
Output: r_U, r_S
$n = pq$ (Generating the modulus)
for U **do**
  $s = k - |KG()| - 1$
  $e = 0^s || KG() || 1$ (Padding the hashed identity with one)
  $r = \frac{1}{e} \pmod{\varphi(n)}$ (Calculating the private key r)
  $r_U \xleftarrow{R} Z_n - [0]$ (Choosing randomly an element r_U from $Z_n - [0]$)
  $r_S = (r - r_U) \pmod{\varphi(n)}$
end

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

Decryption:
Input: C, r_U, r_S
Output: m
for S **do**
  if U *is Revoked* **then**
    return (ERROR)
    Exit
  end
  $PD_S = C^{r_S} \pmod{n}$ (Calculate the partially decrypted message of the SEM)
end
for U **do**
  $PD_U = C^{r_U} \pmod{n}$ (Calculate the partially decrypted message of U)
  $M = (PD_S \times PD_U) \pmod{n}$ (Decrypt the message)
end
m = OAEP Decoding of M

3. The ID-MRSA Security

The ID-MRSA is assumed to be secure in the random oracle model based on [15] and [11]. However, there is a special attack that an insider user can initiate. He can modify the encrypted message so that it can be decrypted using his private key by finding a mapping function $f(C_A) = C_B$.

Lemma 1. Assume that there are two users U_A and U_B, U_B is able to obtain a mapping function $f(C_A) = C_B$ and decrypt/forge the encrypted message/signed message of U_A iff $e_a | e_b$.

This lemma and its proof are presented in [11]. If $e_a | e_b$ i.e. $e_b = k \times e_a$, we can build a mapping function f such that $f(a) = a^k \pmod{n}$. To protect the system

Signing:
Input: m, r_U, r_S
Output: h, S
$h = H(m)$
for S **do**
  if U *is revoked* **then**
    return (ERROR)
    Exit
  end
  $PD_S = h^{r_S} \pmod{n}$
end
for U **do**
  $PD_U = c^{r_U} \pmod{n}$
  $S = (PD_S \times PD_U) \pmod{n}$
end
Verification:
Input: $h, S, n, k, KG()$
Output: $\overline{h}$
$s = k - |KG()| - 1$
$e = 0^s || KG() || 1$
$\overline{h} = S^e \pmod{n}$
if $h \neq \overline{h}$ **then**
  return (ERROR)
end

against this attack, the user's public key cannot be a factor of the product of the other users' public keys. To ensure that, Ding and Tsudik 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 presented by Gennaro et al. [13]. A hash function $H()$ is division intractable if it is unfeasible to find a set of values $(X_1, X_2, ..., X_n, Y)$ such that $H(Y)|\prod_i(H(X_i))$.

In this section, we prove that the ID-MRSA is still vulnerable to this attack. A division intractable hash function does not necessarily produce division intractable public keys because the output of the hash function $KG()$ is padded with a 'one'. The public key is $e = KG()||1$ [11] or $e = KG()||00000001$ [4]. This means that $e = 2KG() + 1$ or $e = 8KG() + 1$. This multiplication and addition completely change the property of the public key and it is likely, with overwhelming probability, to no more becoming 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 $e_2 = 2|KG(ID_2)| + 1 = 2 \times 19 + 1 = 39$ 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

Table 1. 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

the ID-MRSA even though the hash function is division intractable. Real life values that represent the same idea are shown in table 1. These numbers are in hexadecimal.

We now demonstrate how an insider one-wayness adversary takes advantage of this simple notice to initiate two different attacks against 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 system, we prove that if such a mapping function exists, an insider attacker can forge the signature of another user without knowing his private key.

3.1. Attacks on the ID-MRSA Encryption

The first attack holds when the effect of using an intractable hash function is canceled by padding the output with one and the resulting public keys are in the form of $(e_B = k \times e_A)$. Under these conditions, U_B can obtain the message of U_A using the following formula:

$$C_B = C_A^{e_B/e_A} \pmod{n}$$

and then decrypt this message using his private key. This attack is executed as follows:

- The attacker U_B chooses an identity ID_B such that $e_B = k \times e_A$ where k is an integer.
- At the challenge phase, U_B sends to the challenger any two messages m_0 and m_1 and the identity ID_A.
- The challenger tosses a fair coin $b \in \{0, 1\}$ and sends $C_A \leftarrow Enc(m_b)$ to U_B.
- U_B calculates $C_B = C_A^{e_B/e_A} \pmod{n}$.
- U_B sends C_B to the SEM for decryption.
- After decryption, U_B can successfully find $b' = b$.

The gravity of this attack is that it makes the ID-MRSA exposed against a one-wayness adversary; not only can U_B distinguish between two messages m_0 and m_1, he can decrypt it as a message of his own.

The second attack can be applied if the same message was sent to two users, U_A and U_B, U_C with public key satisfies $gcd(e_A, e_B)|e_C$ can launch an attack to decrypt this message as follows.

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

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

and then he can decrypt it using his private key.

3.2. The attack on the ID-MRSA signature

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

- U_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 U_A: $\overline{m}_A = m_B^k \pmod n$ where $k = e_B/e_A$.
- $\overline{m}_A$ can be verified using the public key of U_A.

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

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

4. The ID-MRSA-V2

After showing the security flaw of the ID-MRSA encryption/signature systems, we present two solutions that correctly make the ID-MRSA secure against these types of attacks. We denote the ID-MRSA with these solutions as the ID-MRSA-V2. Any solution to these attacks 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 produced public keys must be co-prime with $\varphi(n)$.

The first solution ensures that the maximum value of a public key is less than three times the smallest public key value, 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. The relation between the hash function of the maximum and minimum public keys values must be:

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

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

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

where *step* is the value used to generate unique primes. After that, 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:

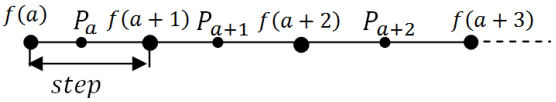

Figure 1. The distribution of primes

- The hash function must be collision resistant i.e. 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 chosen carefully such that $f(a) < P_a < f(a+1)$ 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 [2].
- If the mapping function satisfies the above conditions, it will overcome the first attack to the encryption system because primes satisfy the division intractable property. However, it cannot withstand the second attack because the greatest common divisor (gcd) between primes is one. The only solution for this attack is not to use the same OAEP padding when encrypting the same message to multiple users. For the signature systems, there is no mapping function exists between primes and consequently it will be safe from such attacks. After fixing these drawbacks, the ID-MRSA-V2 can be proven CCA2 secure in the random oracle model using the same methodology explained in [11] or [15].

5. Implementation

The ID-MRSA-V2 was programmed using MIRACL software C library and its performance was compared with the ID-MRSA and RSA. The PC 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:

- The ID-MRSA-V2 has the same performance of the original ID-MRSA.
- 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 and 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.
- The decryption times are longer than the encryption times in all systems. 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 systems are proportional to the modulus size.

6. Generic Mediated Encryption

In this section, we take the idea of ID-MRSA one step ahead. Assume that there is a company XYZ and the security manager of this company wants to upgrade the currently-used IBE to one that supports key revocation. The security manager has two options. He can install a CBE system [14] but he has to uninstall the currently-used IBE and install a PKE. PKE certificates will lead to more computation and transmission costs. The other option is using mediated cryptosystem such as ID-MRSA [6, 15]. The security manager also has to uninstall the current IBE system and install ID-MRSA. The process of uninstalling the currently-used IBE and installing a new encryption system is time-consuming and expensive. It is like having a safe with a one-key lock and you want to replace it with a two-key lock, you will have to completely remove the old lock and install the new one. The question we address here is "Is there a way to make any IBE support key revocation without having to uninstall it?". We take the idea of ID-MRSA and make it generic and applicable to any encryption system. In the following section, we explain the security model and security proof of GME.

6.1. The Model

Definition 1. A Generic Mediated Encryption is a 6- tuple of algorithms. These algorithms are (Gen_S, KG_S, Gen_U, KG_U, Enc, Dec_S, Dec_U) such that:

- $Gen_U(1^{k_1})$: The private key generator (PKG) runs the probabilistic IBE key generation algorithm Gen_S which takes as input a security parameter

Table 2. The time results

The Process	Modulus	Key Size	RSA	ID-MRSA	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

1^{k_1}. It returns MSK_S (the first PKG master secret) and public parameters P_S.

- $Gen_U(1^{k_2})$: The PKG runs the probabilistic IBE key generation algorithm Gen_U which takes as input a security parameter 1^{k_2}. It returns MSK_U (the second PKG master secret) and public parameters P_U.
- $KG_S(MSK_S, P_S, ID_S)$: This algorithm generates the secret key r_S for a SEM with identity ID_S using P_S and MSK_S.
- $KG_U(MSK_U, P_U, ID_U)$: This algorithm generates the secret key r_U for a user with identity ID_U using P_U and MSK_U.
- Enc$(P_S, P_U, ID_U, ID_S, m)$: The probabilistic algorithm Enc takes P_S, P_U, ID_U, ID_S, m. It returns a ciphertext C.
- $Dec_S(P_S, r_S, C)$: The deterministic decryption algorithm Dec_S takes (P_S, r_S, C) as input along with the user revocation status. If the user is revoked, Dec_S returns $\perp$. Otherwise it returns C_U.
- $Dec_U(P_U, r_U, C_U)$: The deterministic decryption algorithm Dec_U takes (P_U, r_U, C_U) as input and returns m.

6.2. Security

Our main concern is the GME security against two different types of attackers: 1) by a revoked user or 2) by a hacked SEM. GME must be secure against each of these individuals considering that each obtains 'half' of the information needed to decrypt. Correspondingly, we define IND-CCA security using two different games. The adversary selects the game to play. In the first game, Type 1, the adversary plays the role of a revoked user. After demonstrating knowledge of the private key related to his identity, the revoked user can make Dec_S queries. In the second game, Type 2, the adversary plays the role of a compromised SEM. After demonstrating knowledge of the private key related to his identity, a

compromised SEM can make Dec_U queries. We can say that our system is secure if no adversary can win either Type 1 or Type 2.

Type 1: The challenger runs $Gen_S(1^{k_1})$ and $Gen_U(1^{k_2})$ and gives P_S and P_U to the adversary. The adversary then interleaves key extraction quires and decryption queries with a single challenge query. These queries are answered as follows:

- On key extraction queries $(MSK_U, P_U, ID_U, P_S, ID_S)$, the challenger runs KG_U, KG_S and outputs r_U and r_S corresponding to the identities ID_U and ID_S.
- On decryption queries $(P_S, P_U, ID_U, ID_S, r_U, C)$, the challenger checks that r_U is the private key related to ID_U. If so, it generates r_S and outputs $Dec_U(Dec_S(C))$.
- On challenge query $(P_S, P_U, ID_S^*, ID_U^*, m_0, m_1)$, the challenger checks that r_U^* is the private key related to ID_U^*. Then, upon receiving two messages m_0 and m_1 from the adversary, the challenger chooses random bit $b \in \{0,1\}$ and returns $Enc(m_b)$ to the adversary. The adversary is allowed to make key extraction and decryption queries after submitting the challenge.

In the end, the adversary outputs a guess $b' \in \{0,1\}$. The adversary wins the game if $b' = b$ and ID_S^* and r_S^* were not subject of valid key extraction and decryption queries. The adversary's advantage is defined to be the absolute value of the difference between $1/2$ and its probability of winning.

Type 2: The challenger runs $Gen_S(1^{k_1})$ and $Gen_U(1^{k_2})$ and gives P_S and P_U to the adversary. The adversary then interleaves key extraction quires and decryption queries with a single challenge query. These queries are answered as follows:

- On key extraction queries $(MSK_U, P_U, ID_U, P_S, ID_S)$, the challenger runs KG_U, KG_S and outputs r_U and r_S corresponding to the identities ID_U and ID_S.
- On decryption queries $(P_S, P_U, ID_U, ID_S, r_S, C)$, the challenger checks that r_S is the private key related to ID_S. If so, it generates r_U and outputs $Dec_U(Dec_S(C))$.
- On challenge query $(P_S, P_U, ID_U^*, ID_S^*, m_0, m_1)$, the challenger checks that r_S^* is the private key related to ID_S^*. Then, upon receiving two messages m_0 and m_1 from the adversary, the challenger chooses random bit $b \in \{0,1\}$ and returns $Enc(m_b)$ to the adversary. The adversary is allowed to make key extraction and decryption queries after submitting the challenge.

In the end, the adversary outputs a guess $b' \in \{0,1\}$. The adversary wins the game if $b' = b$ and ID_U^* and r_U^* were not subject of valid key extraction and decryption queries.The adversary's advantage is defined to be the absolute value of the difference between $1/2$ and its probability of winning.

Definition 2. A generic mediated encryption system is secure against adaptive chosen ciphertext attack (IND-GME-CCA) if no PPT adversary has non-negligible advantage in either Type 1 or Type 2.

Remark: Type 1 and Type 2 are IND-GME-CCA secure if both IBE_S and IBE_U are IND-ID-CCA secure. If IBE_S and IBE_U are IND-ID-CPA secure, then Type 1 and Type 2 are modified by eliminating the decryption queries to get IND-GME-CPA security.

6.3. Security Proof

The security proof of GME is defined by the following two theorems.

Theorem 1. If an adversary $\mathcal{A}$ who plays the role of a revoked user has an advantage ϵ against GME, then this adversary has the same advantage against IBE_S.

Theorem 2. If an adversary $\mathcal{A}$ who plays the role of a hacked SEM has an advantage ϵ against GME, then this adversary has the same advantage against IBE_U.

Proof: Theorem 1 means that the game between an adversary $\mathcal{A}$ who plays the role of a revoked user with a challenger B against GME (Type 1) is identical to the game between the same adversary $\mathcal{A}$ and the challenger B against IBE_S. To prove that, we rewrite Type 1 as follows:

Type 1':

- The Setup phase is the same as Type 1.
- Key extraction queries are the same as Type 1.
- Decryption queries are the same as Type 1.
- On challenge query $(P_S, P_U, ID_S^*, ID_U^*, m_0, m_1)$, the challenger checks that r_U^* is the private key related to ID_U^*. Then, upon receiving two messages m_0 and m_1 from the adversary, the challenger chooses a random bit $b \in \{0,1\}$ and returns $Enc(m_b)$

to the adversary. Since the revoked user has r_U, he can partially decrypt the message to get C_S=$Enc_S(m)$ where Enc_S is the SEM's IBE encryption algorithm.

In the end, the adversary outputs a guess $b' \in \{0,1\}$. The adversary wins the game if $b' = b$ and ID_S^* and r_S^* were not subject of valid key extraction and decryption queries. The adversary's advantage is defined to be the absolute value of the difference between 1/2 and its probability of winning. This concludes Type 1'.

From Type 1', we can see that:

- Type 1' represents a game against IBE_S because in the challenge phase the adversary $\mathcal{A}$ has to attack $C_S = Enc_S(m)$ to get the message m.
- The only difference between a game against GME (in the case of a revoked user) and IBE_S is the excess information of P_U which does not give the adversary any excessive information to identify m.

This concludes the proof of Theorem 1. The proof of Theorem 2 is similar.

7. Implementation of GME

Generally speaking, a GME system is produced by the combination of two IBE systems. To prove that GME is generic, we present GME in two different instantiations. The first one is based on the BF FullIBE [7] which is a pairing-based system. The other instantiation is based on BGH IBE system [8] which is not a pairing-based system. We first briefly review bilinear pairings and the bilinear Diffi-Helman assumption which is the base of the BF FullIBE security then we present GME using BF FullIBE. After that, we briefly review some of the security topics related to the BGH IBE system then we represent GME using BGH IBE system. We note here that the proposed GMEs systems use the same setup and key generation algorithms for both the users and SEM. I

7.1. Review on pairings

BF IBE [7] is based on bilinear map called a 'pairing'. The pairing which is often used to construct BF IBE is a modified Weil or Tate pairing on a supersingular elliptic curve or Abelian variety. However, we review pairings and the related mathematics in a more general form here.

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two cyclic groups of a large prime order q. $\mathbb{G}_1$ is an additive group and $\mathbb{G}_2$ is a multiplicative group.

Admissible pairings: $\hat{e}$ is called an admissible pairing if $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is a map with the following properties:

- Bilinear: $\hat{e}(aQ, bR)$=$\hat{e}(Q, R)^{ab}$ for all $Q, R \in \mathbb{G}_1$ and all $a, b \in \mathbb{Z}$.
- Non-degenerate: $\hat{e}(Q, R) \neq 1$ for all $Q, R \in \mathbb{G}_1$.
- Computable: There is an efficient algorithm to compute $\hat{e}(Q, R)$ for any $Q, R \in \mathbb{G}_1$.
- Symmetric: $\hat{e}(Q, R) = \hat{e}(R, Q)$ for any $Q, R \in \mathbb{G}_1$.

Bilinear Diffie-Hellman (BDH) Parameter Generator: A randomized algorithm $\mathcal{IG}$ is a BDH parameter generator if $\mathcal{IG}$ takes a security parameter $k > 0$, runs in time polynomial in k and outputs the description of two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of the same prime order q and the description of an admissible pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ [7].

BDH Problem: Given a randomly chosen $P \in \mathbb{G}_1$ as well as aP, bP and cP (for unknown randomly chosen $a, b, c \in \mathbb{Z}_q$), compute $\hat{e}(P, P)^{abc}$.

For the BDH problem to be hard, $\mathbb{G}_1$ and $\mathbb{G}_2$ must be chosen so that there is no known algorithm for efficiently solving the Diffie-Hellman problem in either $\mathbb{G}_1$ or $\mathbb{G}_2$.

BDH Assumption: If $\mathcal{IG}$ is a BDH parameter generator, the advantage $Adv_{\mathcal{IG}}(B)$ of algorithm B in solving the BDH problem is defined to be the probability that the algorithm B outputs $\hat{e}(P, P)^{abc}$ when the inputs to the algorithm are $\mathbb{G}_1$, $\mathbb{G}_2$,$\hat{e}$,aP, bP and cP where ($\mathbb{G}_1$, $\mathbb{G}_2$,$\hat{e}$) is $\mathcal{IG}$'s output for large enough security parameter k, P is a random generator of $\mathbb{G}_1$ and a, b, c are random elements of $\mathbb{Z}_q$. The BDH assumption is that $Adv_{\mathcal{IG}}(B)$ is negligible for all efficient algorithms B [7].

7.2. GME_{BF}

Let k be the security parameter given to the setup algorithm and let $\mathcal{IG}$ be a BDH parameter generator.

Setup: The public key generator (PKG) runs $\mathcal{IG}$ on input k to generate groups $\mathbb{G}_1$, $\mathbb{G}_2$ of some prime order q and an admissible pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$. It picks an arbitrary generator $P \in \mathbb{G}_1$ and a master secret $s \in \mathbb{Z}_q$ and sets $P_{pup} = sP$ and chooses cryptographic hash functions $H_1\{0,1\}^* \rightarrow \mathbb{G}_1$, $H_2 : \mathbb{G}_1 \rightarrow \{0,1\}^n$, $H_3 : \{0,1\}^n \times \{0,1\}^n \rightarrow \mathbb{Z}_q$ and a hash function $H_4 : \{0,1\}^n \rightarrow \{0,1\}^n$ for some n. The system parameters are P = ($\mathbb{G}_1$,$\mathbb{G}_2$, $\hat{e}$, P, Q, H_1, H_2, H_3, H_4). The message

space is $\mathcal{M} = \{0,1\}^n$. The master secret is $s \in \mathbb{Z}_q$.

KG: For given strings $ID_U, ID_S \in \{0,1\}^*$, the PKG computes $Q_S = H_1(ID_S)$ and $Q_U = H_1(ID_U)$ and sets the private key $r_S = sQ_S$ and $r_U = sQ_U$.

Enc: To encrypt a message m for a user with public key ID_U, compute $Q_S = H_1(ID_S)$ and $Q_U = H_1(ID_U)$. After that, choose a random $\sigma \in \{0,1\}^n$ and set $r = H_3(\sigma, m)$. The ciphertext C is:

$$C = \langle rp, \sigma \oplus H_2(g_U^r) \oplus H_2(g_S^r), m \oplus H_4(\sigma)) \rangle$$

where $g_U = \hat{e}(Q_U, P_{pub})$ and $g_S = \hat{e}(Q_S, P_{pub})$.

Dec: To decrypt $C = \langle U, V, W \rangle$ for a user with public key ID_U, the user sends C to the SEM. If the user is revoked, the SEM returns $\perp$. If the user is not revoked, the SEM calculates

$$C_U = \langle U, V \oplus H_2(\hat{e}(d_S, U)), W \rangle$$

and returns C_U to the user. After receiving $C_U = \langle U, V_U, W \rangle$, the user computes $V_U \oplus H_2(\hat{e}(d_U, U)) = \sigma$ and $W \oplus H_4(\sigma) = m$ and sets $r = H_3(\sigma, m)$. He outputs m as a decryption of C if $U = rp$. This concludes GME_{BF}.
Remark: A symmetric encryption E can be used instead of Xor to encrypt the message m [7].

7.3. Security Proof

Lemma 2. Let $\mathcal{A}$ be a IND-CCA adversary that has advantage ϵ against GME_{BF}. This adversary $\mathcal{A}$ can be a revoked user or a hacked SEM. Then, there is an IND-CCA adversary B with the same probability ϵ against the BF FullIBE.

Proof. As shown in section 6.3.

7.4. Boneh-Gentry-Hanburg (BGH) system

Boneh, Gentry and Hamburg presented an anonymous IND-ID-CPA secure system (BGH) [8]. Unlike Boneh-Franklin system, this system is secure based on the interactive quadratic residuosity (IQR) assumption. In the following, we present the IQR assumption and the core algorithm of the BGH system, then we present GME based on that system.

7.5. The IQR assumption

For a positive integer n, define the following set:

$$J(n) = [a \in \mathbb{Z}_n : \left(\frac{a}{n}\right) = 1],$$

where $\left(\frac{a}{n}\right)$ is the Jacobi symbol of a w.r.t n [8]. The Quadratic Residue set $QR(n)$ is defined as follows

$$QR(n) = [a \in \mathbb{Z}_n : gcd(a,n) = 1 \wedge x^2 \equiv a \pmod{n} \text{ has a solution}].$$

Definition 3. Interactive Quadratic Residuosity Assumption: Let H be a collision free hash function such that $H : [0,1]^* \to J(n)$. Let $\mathcal{O}$ be a square root oracle that picks $u_n \leftarrow J(n) \setminus QR(n)$ and maps input pair (n,x) to one of $H_n(x)^{\frac{1}{2}}$ or $u_n H_n(x)^{\frac{1}{2}}$ in $\mathbb{Z}_n$ based on which value is quadratic residue. The interactive quadratic residuosity assumption holds for the pair $(RSAgen, H)$ if for all PPT algorithms $\mathcal{A}$, the function $IQRAdv_{\mathcal{A},(RSAgen(\lambda),H)} =$

$$|\Pr[(n,V) \leftarrow P_{QR}(\lambda) : \mathcal{A}^{\mathcal{O}}(n,V) = 1]| - |\Pr[(n,V) \leftarrow P_{NQR}(\lambda) : \mathcal{A}^{\mathcal{O}}(n,V) = 1]|$$

is negligible. $IQRAdv_{\mathcal{A},(RSAgen(\lambda),H)}$ is the IQR advantage of $\mathcal{A}$ against $(RSAgen, H)$ [8].

7.6. $\overline{Q}$Algorithm

$\overline{Q}$ is a deterministic algorithm with inputs (n,u,R,T) where $n \in \mathbb{Z}^+$ and $R,u,T \in \mathbb{Z}_n$. This algorithm outputs four polynomial functions $f, \overline{f}, g, \tau \in \mathbb{Z}_n$. This algorithm must satisfy the following conditions :

- If R and T are quadratic residues, then $f(r)g(t)$ is also quadratic residue for all values of $r \leftarrow R^{\frac{1}{2}}$ and $t \leftarrow T^{\frac{1}{2}}$.
- If uR and T are quadratic residues, then $\overline{f}(\overline{r})g(t)\tau(t)$ is also quadratic residue for all values of $\overline{r} \leftarrow uR^{\frac{1}{2}}$ and $t \leftarrow T^{\frac{1}{2}}$.
- If R is quadratic residue, then $f(r)f(-r)T$ is quadratic residue for every $r \leftarrow R^{\frac{1}{2}}$.
- If uR is quadratic residue, then $\overline{f}(\overline{r})\overline{f}(-\overline{r})T$ is quadratic residue for every $\overline{r} \leftarrow uR^{\frac{1}{2}}$.
- If T is quadratic residues, then $\tau(t)\tau(-t)u$ is also quadratic residue for all values of $t \leftarrow T^{\frac{1}{2}}$.
- τ is independent of R, that is $\overline{Q}(n,u,R_1,T)$ and $\overline{Q}(n,u,R_2,T)$ produces the same value of τ for any value of n,u,R_1,R_2,T.

An example of $\overline{Q}$ is explained [8] as follows:

- Find a solution $(x,y) \in \mathbb{Z}_n^2$ to the equation $Rx^2 + Ty^2 \equiv 1 \pmod{n}$.
- Find a solution $(\alpha,\beta) \in \mathbb{Z}_n^2$ to the equation $u\alpha^2 + T\beta^2 \equiv 1 \pmod{n}$.
- Calculate the polynomials $f(r) \leftarrow xr + 1$, $\overline{f}(\overline{r}) \leftarrow 1 + Ty\beta + \alpha x\overline{r}$, $g(t) \leftarrow 2yt + 2$, $\tau(t) = 1 + \beta t$.

7.7. GME_{BGH}

- Setup(1^k): Using RSAgen(1^k), generate (p,q). Calculate the modulus $n \leftarrow pq$. Choose $u \in J(n) \setminus QR(n)$ and choose a hash function $H : ID \times [1,l] \rightarrow J(n)$. The public parameters P are $[n,u,H]$. The master secret MSK parameters are p,q and a secret key K for a pseudorandom function $F_K : ID \times [1,l] \rightarrow [0,1,2,3]$.

- KG(MSK, ID_U, ID_S, l): Using the master secret MSK, ID and the message length l, the private keys $(r_{U,i}, r_{S,i})$ are generated using the key generation algorithm.

Key Generation
foreach $i \in [1,l]$ **do**
  $R_{U,i} \leftarrow H(ID_U, i) \in J(n)$
  $R_{S,i} \leftarrow H(ID_S, i) \in J(n)$
  $w_U \leftarrow F_K(ID_U, i) \in [0,1,2,3]$
  $w_S \leftarrow F_K(ID_S, i) \in [0,1,2,3]$
  choose $a_U \in [0,1]$ such that $u^{a_U} R_{U,i} \in QR(n)$
  choose $a_S \in [0,1]$ such that $u^{a_S} R_{S,i} \in QR(n)$
  let$[z_{U,0}, z_{U,1}, z_{U,2}, z_{U,3}]$ be the four square roots of $u^{a_U} R_{U,i} \in \mathbb{Z}_n$
  let$[z_{S,0}, z_{S,1}, z_{S,2}, z_{S,3}]$ be the four square roots of $u^{a_S} R_{S,i} \in \mathbb{Z}_n$
  $r_{U,i} \leftarrow z_{U,w_U}$
  $r_{S,i} \leftarrow z_{S,w_S}$
end

The decryption key for the user U is $d_U \leftarrow (P, r_{U,1}, ..., r_{U,l})$ and the decryption key for the SEM is $d_S \leftarrow (P, r_{S,1}, ..., r_{S,l})$.

- Enc(P, ID_U, ID_S, m): Generate a random value $t \leftarrow \mathbb{Z}_n$ and calculate $T \leftarrow t^2$ and then encrypt $m \in [-1,1]^l$ using P as shown in the encryption algorithm. The ciphertext is $C = (T, k, c)$ where $c = [c_1, c_2, ..., c_l]$.

- Decrypt(C, d_{ID}): To decrypt a ciphertext $C = (T, k, c)$ for a user with public key ID_U, he sends C to the SEM. The SEM then does the following:

- if U is revoked, the SEM returns $\perp$.

- if U is not revoked, the SEM Calculates $c_{U,i}$ as shown in SEM decryption algorithm.

  and returns $C_U = [C_{U,1}, C_{U,2}, ..., C_{U,l}]$ to the user. Then he decrypts C_U as shown in the user decryption algorithm.

Encryption
$\tau(t) \leftarrow \overline{Q}(n,u,1,T)$
$k \leftarrow \left(\frac{\tau(t)}{n}\right)$
foreach $i \in [1,l]$ **do**
  $R_{U,i} \leftarrow H(ID_U, i) \in J(n)$
  $R_{S,i} \leftarrow H(ID_S, i) \in J(n)$
  $[x_{U,i}, y_{U,i}] \leftarrow \overline{Q}(n,u,R_{U,i},T)$
  $[x_{S,i}, y_{S,i}] \leftarrow \overline{Q}(n,u,R_{S,i},T)$
  $g_{U,i}(t) \leftarrow 2y_{U,i}t + 2$
  $g_{S,i}(t) \leftarrow 2y_{S,i}t + 2$
  $c_i \leftarrow m_i \cdot \left(\frac{g_{U,i}(t)}{n}\right) \cdot \left(\frac{g_{S,i}(t)}{n}\right)$
end

SEM Decryption
foreach $i \in [1,l]$ **do**
  $R_{S,i} \leftarrow H(ID_S, i) \in J(n)$
  if $r_{S,i}^2 = R_{S,i}$ **then**
    $[x_{S,i}, y_{S,i}] \leftarrow \overline{Q}(n,u,R_{S,i},T)$
    $f_{S,i}(r_{S,i}) \leftarrow x_{S,i} r_{S,i} + 1$
    $c_{U,i} \leftarrow c_i \cdot \left(\frac{f_{S,i}(r_{S,i})}{n}\right)$
  end
  if $\overline{r}_{S,i}^2 = uR_{S,i}$ **then**
    $[x_{S,i}, y_{S,i}, \alpha, \beta] \leftarrow \overline{Q}(n,u,R_{S,i},T)$
    $\overline{f}_{S,i}(\overline{r}_{S,i}) \leftarrow 1 + T^{2i-1} y_{S,i}\beta + \alpha x_{S,i} \overline{r}_{S,i}$
    $c_{U,i} \leftarrow c_i \cdot \left(\frac{\overline{f}_{S,i}(\overline{r}_{S,i})}{n}\right) \cdot k$
  end
end

User Decryption
foreach $i \in [1,l]$ **do**
  $R_{U,i} \leftarrow H(ID_U, i) \in J(n)$
  if $r_{U,i}^2 = R_{U,i}$ **then**
    $[x_{U,i}, y_{U,i}] \leftarrow \overline{Q}(n,u,R_{U,i},T)$
    $f_{U,i}(r_{U,i}) \leftarrow x_{U,i} r_{U,i} + 1$
    $m_i \leftarrow c_i \cdot \left(\frac{f_{U,i}(r_{U,i})}{n}\right)$
  end
  if $\overline{r}_{U,i}^2 = uR_{U,i}$ **then**
    $[x_{U,i}, y_{U,i}, \alpha, \beta] \leftarrow \overline{Q}(n,u,R_{U,i},T)$
    $\overline{f}_{U,i}(\overline{r}_{U,i}) \leftarrow 1 + T^{2i-1} y_{U,i}\beta + \alpha x_{U,i} \overline{r}_{U,i}$
    $m_i \leftarrow c_i \cdot \left(\frac{\overline{f}_{U,i}(\overline{r}_{U,i})}{n}\right) \cdot k$
  end
end

This concludes GME_{BGH}.

7.8. Security Proof

Lemma 3. Let $\mathcal{A}$ be an Anon-IND-CPA adversary that has advantage ϵ against GME_{BGH}. This adversary $\mathcal{A}$ can be a revoked user or hacked SEM. Then, there is an Anon-IND-CPA adversary B with the same probability ϵ against the BGH system.

Proof. As shown in section 6.3.

8. Conclusion

In this paper, we investigate the mediated structure of the ID-MRSA which is a solution to the key revocation problem. We showed that using a division intractable hash function does not necessarily guarantee that the generated public keys are also division intractable. Consequently, the system 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. After that, we extended the idea of the ID-MRSA to be generic by presenting a generic mediated encryption (GME) system that converts any IBE system to a mediated system. Although it is based on double encryption, our system is efficient. The ciphertext size is the same as a single IBE. It combines the advantage of CBE and SEM structures. Our system is more efficient than CBE because it does not depend on certificates and it is more secure than [6] and [15] because the SEM in GME is not a single point of failure and can be untrusted. We prove that GME is as secure as the IBE system used in the case of a revoked user or a hacked SEM.

References

[1] W. Aiello, S. Lodha, and R. Ostrovsky. Fast digital identity revocation. In H. Krawczyk, editor, *Advances in Cryptology, CRYPTO '98*, volume 1462 of *Lecture Notes in Computer Science*, pages 137–152. Springer Berlin Heidelberg, 1998.

[2] J. K. Andersen. Maximal Prime Gap.

[3] D. Boneh and X. Boyen. Efficient Selective-ID Secure Identity Based Encryption Without Random Oracles. In *Advances in Cryptology—EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 223–238. Berlin: Springer-Verlag, 2004. Available at http://www.cs.stanford.edu/~xb/eurocrypt04b/.

[4] D. Boneh, X. Ding, and G. Tsudik. Identity-based mediated RSA. In *In Proceedings of the 3rd International Workshop on Information Security Applications(WISA'02) (Jeju Island, Korea)*, 2002.

[5] D. Boneh, X. Ding, and G. Tsudik. Fine-grained control of security capabilities. *ACM Trans. Internet Technol.*, 4(1):60–82, Feb. 2004.

[6] 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*, SSYM'01, pages 22–22, Berkeley, CA, USA, 2001. USENIX Association.

[7] D. Boneh and M. Franklin. Identity-Based Encryption from the Weil Pairing. In J. Kilian, editor, *Advances in Cryptology, CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer Berlin Heidelberg, 2001.

[8] D. Boneh, C. Gentry, and M. Hamburg. Space-Efficient Identity Based Encryption Without Pairings. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '07, pages 647–657, Washington, DC, USA, 2007. IEEE Computer Society.

[9] C. Cocks. An Identity Based Encryption Scheme Based on Quadratic Residues. In B. Honary, editor, *Cryptography and Coding*, volume 2260 of *Lecture Notes in Computer Science*, pages 360–363. Springer Berlin Heidelberg, 2001.

[10] 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, May 2008.

[11] X. Ding and G. Tsudik. Simple Identity-Based Cryptography with Mediated RSA. In M. Joye, editor, *Topics in Cryptology, CT-RSA 2003*, volume 2612 of *Lecture Notes in Computer Science*, pages 193–210. Springer Berlin Heidelberg, 2003.

[12] I. Gassko, P. Gemmell, and P. MacKenzie. Efficient and Fresh Certification. In H. Imai and Y. Zheng, editors, *Public Key Cryptography*, volume 1751 of *Lecture Notes in Computer Science*, pages 342–353. Springer Berlin Heidelberg, 2000.

[13] R. Gennaro, S. Halevi, and T. Rabin. Secure Hash-and-Sign Signatures Without the Random Oracle. In J. Stern, editor, *Advances in Cryptology, EUROCRYPT99*, volume 1592 of *Lecture Notes in Computer Science*, pages 123–139. Springer Berlin Heidelberg, 1999.

[14] C. Gentry. Certificate-Based Encryption and the Certificate Revocation Problem. In E. Biham, editor, *Advances in Cryptology, EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 272–293. Springer Berlin Heidelberg, 2003.

[15] C. Gentry. Practical Identity-Based Encryption Without Random Oracles. In *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, volume 4004 of *Lecture Notes in Computer Science*, pages 445–464. Springer, 2006.

[16] 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,*, 2:1068, 2001.

[17] P. Kocher. On certificate revocation and validation. In R. Hirchfeld, editor, *Financial Cryptography*, volume 1465 of *Lecture Notes in Computer Science*, pages 172–177. Springer Berlin / Heidelberg, 1998. 10.1007/BFb0055481.

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

[19] S. Micali. Efficient Certificate Revocation. Technical report, Cambridge, MA, USA, 1996.

[20] S. Micali. Novomodo: Scalable Certificate Validation and Simplified PKI Management. In *1st Annual PKI Research Workshop*, pages 15–25, 2002.

[21] M. Naor and K. Nissim. Certificate revocation and certificate update. In *Proceedings of the 7th conference on USENIX Security Symposium - Volume 7*, SSYM'98, pages 17–17, Berkeley, CA, USA, 1998. USENIX Association.

[22] M. Naor and K. Nissim. Certificate revocation and certificate update. *Selected Areas in Communications, IEEE Journal on*, 18(4):561–570, 2000.

[23] W. F. a. D. S. R. Housley, W. Polk. RFC3280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, April 2002.

[24] K. O. R. Sakai and M. Kasahara. Cryptosystems based on Pairing. In *Symposium on Cryptography and Information Security (SCIS 2000), Japan*, 2000.

[25] A. Shamir. Identity-Based Cryptosystems and Signature Schemes. In G. Blakley and D. Chaum, editors, *Advances in Cryptology*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53. Springer Berlin Heidelberg, 1985.

[26] B. Waters. Efficient Identity-Based Encryption Without Random Oracles. In R. Cramer, editor, *Advances in Cryptology, EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 114–127. Springer Berlin Heidelberg, 2005.