Generic Mediated Encryption

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
ifeae231@uowmail.edu.au, {ymu,wsusilo}@uow.edu.au

Abstract. We propose a generic mediated encryption (GME) system that converts any identity based encryption (IBE) to a mediated IBE. This system is based on enveloping an IBE encrypted message using a user's identity into another IBE envelope, using the identity of a security mediator (SEM) responsible for checking users for revocation. 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 (the envelope) against a revoked user and as secure as the user's IBE (the letter) against a hacked SEM. We also present two instantiations of GME. The first instantiation is based on the Boneh-Franklin (BF) FullIBE system, which is a pairing-based encryption system. The second instantiation is based on the Boneh, Gentry and Hamburg (BGH) system, which is a non pairing-based encryption system.

Keywords: Key Revocation Problem, Identity-based Encryption, Double Encryption.

1 Introduction

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 the purpose of managing the 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, which is the user's public key, 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. For Instance, if a user's secret key is accidentally leaked or an attacker is successful in compromising it, the user's public key and private key should be revoked; a new key pair should be generated and the corresponding certificate should be issued.

T. Zia et al. (Eds.): SecureComm 2013, LNICST 127, pp. 154–168, 2013.

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. In addition, it must be widely distributed. Sharing a great deal of fresh certification information 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)[17,7]. A CRL is a list that contains certificates revoked before their due date. The CA produces this list periodically, with its signature. Since the CA will probably revoke many of its certificates -say 10 %- if they are produced for a validity time of one year[15,11], the CRL will be too lengthy if the CA has many clients. Despite this, 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[2] which list only those certificates revoked since the CA's last update, but the consumed transmission bandwidth costs, and the computation costs required to enable the transmission of these lists are still very high. Another method of solving the key revocation problem is the online certificate status protocol (OCSP)[13].

Micali [15,11,14] proposed a promising way to solve this problem. (See also [16,9,10].) 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, however, 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 depended on third parties, the directories only publish hash 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 because it only generates one signature per query, 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 for solving the key revocation problem is relaying on third-party queries[11]. It is preferable to eliminate third-party queries for several reasons. First, since anyone can ask for third-party

queries, each certificate server in the system 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 clines and consequently, the transmission costs are reduced. Second, third-party queries multiply the query computation costs of the CA and/or its servers. For example, if each client queries the certification status of X other clients per day, then 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 user's CA, what motivation does the CA have 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 where T, without acquiring any information other than the user's public key and the parameters of the user's CA, can encrypt its message to the user so that he can decrypt only if the key is currently certified. This allows us to enjoy the infrastructure benefits of eliminating third-party queries. This can be achieved by identity-based encryption (IBE).

The notion of identity-based cryptography was put forth by Shamir [19]. 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 the construction for identity-based encryption can not be achieved using a similar approach since sharing a common modulus between different users make RSA insecure. Examples of cryptanalysis RSA with the same modulus used for different encryption/decryption pairs are [20,1]. Sixteen years later, Sakai, Ohgishi and Kasahara [18] proposed the first identity-based cryptography and independently, Boneh and Franklin [4] proposed the first reliable and provable identity-based cryptography, which is based on Weil pairings over elliptic curves. Cocks [6] 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)[11]. 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 also being able to be used as a decryption key. This double encryption gives us implicit certification. If T wants to encrypt a message to the user, it double encrypts the message using PKI and IBE, and then the user 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 since 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 [3]. 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 (SEM) architecture, in which 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 based on 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 the message. This is also helpful to monitor the security of sent/received secure messages in the company. This SEM architecture was proven useful [3] 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 hence, the computation and transmission costs are kept to minimum, it has two major security concerns. First, There is a security flaw in [8,12]. 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, because SEM is a semi-trusted server, a hacked SEM can be a major threat to the system security.

1.2 Our Contribution

Assume that there is a company, XYZ, and the security manager of this company wants to upgrade the currently-used IBE system to one that supports key revocation. The security manager has two options. He can install a CBE system [11], 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 SEM structure as presented in [3,12]. The security manager also has to uninstall the current IBE system and install a new one that supports key revocation. The system will be more vulnerable to DoS attacks. The process of uninstalling the currently used IBE and install 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 in this paper is "Is there a way to make any IBE support key revocation without having to uninstall it?".

In this paper, we present a technique that is capable of making any IBE system support key revocation. This idea is based on a letter-envelope technique. If T wants to encrypt a message to U, he first encrypts it, normally using U's identity (letter), then he encrypts the letter again using SEM identity (envelope). After that, the message is sent back to SEM. If U is revoked, SEM will not open the envelope for him. If U is not revoked, the SEM will open the envelope and send the letter to U who decrypts the message using his private key. This is like

installing a new lock beside the old one. The original key is with the user and the other key is with the SEM.

The structure of our system combines the advantages of both Gentry[11] and Boneh *et al.* [3]. It eliminates completely 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 IBE system. In addition, the SEM does not have to be trusted or semi-trusted. If the SEM is compromised, all the messages sent to the SEM, before or after an attack, are safe and secure.

Paper Organization. The rest of the paper is organized as follows: Section 2 presents the generic mediated encryption (GME) and its security proof. Section 3 presents two implementations of GME, the first one based on the BF IBE system[4], which is based on pairings, and the second one based on the BGH system[5], which is not based on pairing. The last section presents the conclusions of the paper.

2 Generic Mediated Encryption

In the following section, we explain the security model and security proof of GME. Table 1 presents the definitions of the symbols used.

Table 1. Symbols

Symbol	Definition
U	User
S	SEM
P	System Parameters
Gen	IBE Setup Algorithm
KG	IBE Key Generation Algoritm
Enc	Encryption Algorithm
Dec	Decryption Algorithm
r	The private Key

2.1 The Model

Definition 1. *A Generic Mediated Encryption system 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* 1^{k_1}*. It returns* MSK_S *(first PKG master secret) and public parameters* P_S*.*
- $Gen_U(1^{k_2})$*:PKG runs the probabilistic IBE key generation algorithm* Gen_U*, which takes as input a security parameter* 1^{k_2}*. It returns* MSK_U *(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 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 user with identity ID_U using P_U and MSK_U.*
- *$Enc(P_S, P_U, ID_U, ID_S, m)$: The probabilistic encryption algorithm Enc takes P_S, P_U, ID_U, ID_S, m. It returns a ciphertext C on message m.*
- *$Dec_S(P_S, r_S, C)$: 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. It returns m.*

2.2 Security

Our main concern is the GME security against two different types of attacks: 1) by a revoked user and 2) by a compromised 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 game.

Type 1: The challenger runs $Gen_S(1^{k_1}, t_1)$ and $Gen_U(1^{k_2}, t_2)$, 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), the challenger outputs r_U corresponding to the identity ID_U, otherwise it returns $\perp$.
- 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 $Dec_U(Dec_S(C))$, otherwise it returns $\perp$.
- On challenge query $(P_S, P_U, ID'_U, r'_U, M_0, M_1)$ the challenger checks that r_U is the private key related to ID_U. If so, it chooses random bit b and returns $Enc(m)$, otherwise it returns $\perp$.

In the end, the adversary outputs a guess $b' \in \{0, 1\}$. The adversary wins the game if $b' = b$ and ID'_U, r'_U was not a subject of a valid decryption query after the challenge. 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 $Gen_S(1^{k_1}, t_1)$ and $Gen_U(1^{k_2}, t_2)$, 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_S, P_S, ID_S), the challenger outputs r_S corresponding to the identity ID_S, otherwise it returns $\perp$.
- 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 $Dec_U(Dec_S(C))$, otherwise it returns $\perp$.
- On challenge query $(P_S, P_U, ID'_S, r'_S, M_0, M_1)$ the challenger checks that r_S is the private key related to ID_S. If so, it chooses random bit b and returns $Enc(m)$, otherwise it returns $\perp$.

In the end, the adversary outputs a guess $b' \in \{0, 1\}$. The adversary wins the game if $b' = b$ and $(ID'_S, r'_S$ was not a subject of a valid decryption query after the challenge. 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.

2.3 Security Proof

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

Theorem 1. *If an adversary 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 A, who plays the role of a compromised 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 adversary A, who plays the role of a revoked user, and challenger B against GME (Type 1) is identical to the game between the same adversary 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 $(P_S, P_U, ID'_U, r'_U, M_0, M_1)$ the challenger checks that r_U is the private key related to ID_U. If so, it chooses random bit b and returns $C = Enc(m)$, otherwise it returns $\perp$. Since the revoked user has r_U, then he can partially decrypt the message to get C_S=$Enc_S(m)$, where Enc_S is the the SEMs IBE encryption algorithm.

In the end, the adversary outputs a guess $b' \in \{0,1\}$. The adversary wins the game if $b' = b$ and $(ID'_S, r'_S$ was not a subject of a valid decryption query after the challenge. 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 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 information to identify m.

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

3 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 [4] which is based on pairings. The other instantiation is based on BGH IBE system[5], which is not based on pairings . 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.

3.1 Review on Pairings

BF IBE [4] 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 $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ 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: As in [4], we say that 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 $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$.

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: As in [6], if $\mathcal{IG}$ is a BDH parameter generator, the advantage $Adv_{\mathcal{IG}}(B)$ that an algorithm B has 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.

3.2 GME_{BF}

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

- **Setup**: The algorithm works as follows:
- 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 $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$.
- Picks an arbitrary generator $P \in \mathbb{G}_1$.
- Picks a master secret $s \in \mathbb{Z}_q$ and sets $P_{pup} = sP$.
- Chooses cryptographic hash functions $H_1\{0,1\}^* \to \mathbb{G}_1$, $H_2 : \mathbb{G}_1 \to \{0,1\}^n$, $H_3 : \{0,1\}^n \times \{0,1\}^n \to \mathbb{Z}_q$ and a hash function $H_4 : \{0,1\}^n \to \{0,1\}^n$ 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 $s \in \mathbb{Z}_q$.

- **Extract**: For given strings $ID_U, ID_S \in \{0,1\}^*$, the algorithm do the following:
- Computes $Q_S = H_1(ID_S)$ and $Q_U = H_1(ID_U)$.
- Sets the private key $r_S = sQ_S$ and $r_U = sQ_U$.

- **Encrypt**: To encrypt $M \in \mathcal{M}$ for a user with public key ID_U, do the following:
- Compute $Q_S = H_1(ID_S)$ and $Q_U = H_1(ID_U)$.
- Chooses a random $\sigma \in \{0,1\}^n$.
- Sets $r = H_3(\sigma, M)$.
- Sets the ciphertext C as:

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

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

- **Decrypt**: To decrypt $C = \langle U, V, W \rangle \in \mathcal{C}$ for a user with public key ID_U, the user sends C to the SEM. The SEM does the following:
- if user is revoked, the SEM returns $\perp$.
- if user is not revoked, the SEM returns

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

- The SEM sends C_U to user.
- After receiving $C_U = \langle U, V_U, W \rangle$, the user calculates M as follows:
- Computes $V_U \oplus H_2(\hat{e}(d_U, U)) = \sigma$.
- Computes $W \oplus H_4(\sigma) = M$.
- Sets $r = H_3(\sigma, M)$. Test that $U = rp$. If not, reject the ciphertext, otherwise the user outputs M as a decryption of C.

This concludes GME_{BF}.
Remark : As in [4], a symmetric encryption E can be used instead of Xor to encrypt the message m.

3.3 Security Proof

Lemma 1. *Let A be a IND-CCA adversary that has advantage ϵ against GME_{BF}. This adversary A can be a revoked client or a hacked SEM. Then, there is an IND-CCA adversary B with the same probability ϵ against the BF FullIBE.*

Proof. If an adversary A simulates the role of a revoked user, then he plays Type 1 with the challenger. The ciphertext sent to the adversary is $C = \langle rp, M \oplus H_2(g_U^r) \oplus H_2(g_S^r) \rangle$. The adversary then partially decrypts it using his secret key r_U to get $C_S = \langle rp, M \oplus H_2(g_S^r) \rangle$, which is the message m encrypted by FullIBE using the SEM's ID. This also can be applied for a hacked SEM.

3.4 Boneh-Gentry-Hanburg (BGH) Scheme

Boneh, Gentry and Hamburg presented an Anon-IND-ID-CPA scheme [5]. Unlike the Boneh-Franklin scheme, this scheme is secure based on the quadratic residuosity (QR) assumption. In the following, we present the QR assumption and Jacobi symbols, then we present GME based on the BGH scheme.

3.5 QR Assumption and Jacobi Symbols

For a positive integer N, define the following set:

$$J(N) = [a \in \mathbb{Z}_N : \tfrac{a}{N} = 1]$$

where $\frac{a}{N}$ is the Jacobi symbol of a w.r.t N. The quadratic residue set $QR(N)$ is defined as follows

$$QR(N) = [a \in \mathbb{Z}_N : gcd(a, N) \wedge x^2 \equiv a \bmod N \text{ has a solution}].$$

Definition 1. *Quadratic Residuosity Assumption: Let RSAgen(1^k) be a probabilistic polynomial time (PPT) algorithm. This algorithm generates two equal size primes p, q. The QR assumption holds for RSAgen if it cannot distinguish between the following two distributions for all PPT algorithms A.*

$$P_{QR}(1^k) : (N, V)(p, q) \leftarrow RSAgen(1^k), N = pq, V \in_R QR(N)$$
$$P_{NQR}(1^k) : (N, V)(p, q) \leftarrow RSAgen(1^k), N = pq, V \in_R J(N) \setminus QR(N)$$

i.e. adversary A cannot distinguish between elements in $J(N) \setminus QR(N)$ and elements in $QR(N)$.

Definition 2. *Interactive Quadratic Residuosity Assumption: Let H be a collision free hash function such that $H : [0, 1]^* \rightarrow J(N)$. 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 residue assumption holds for the pair $(RSAgen, H)$ if for all PPT algorithms A, the function $IQRAdv_{A,(RSAgen(1^k),H)} =$*

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

is negligible. $IQRAdv_{A,(RSAgen(1^k),H)}$ is the IQR advantage of A against $(RSAgen, H)$.

3.6 $\overline{Q}$ Algorithm

$\overline{Q}$ is a deterministic algorithm with inputs (N, u, R, I), where $N \in \mathbb{Z}^+$ and $R, u, I \in \mathbb{Z}_N$. This algorithm outputs four polynomial functions $f, \overline{f}, g, \tau \in Z[x]_N$. This Algorithm must satisfy the following conditions to be Enhanced IBE compatible:

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

An example of $\overline{Q}$ is explained in [5] as follows:

- Find a solution $(x, y) \in \mathbb{Z}_N^2$ to the equation $Rx^2 + Sy^2 = 1 \bmod N$.

- Find a solution $(\alpha, \beta) \in \mathbb{Z}_N^2$ to the equation $u\alpha^2 + I\beta^2 = 1 \bmod N$.
- Calculate the polynomials $f(r) \leftarrow xr + 1$, $\overline{f}(\overline{r}) \leftarrow 1 + Sy\beta + \alpha x\overline{r}$, $g(i) \leftarrow 2ys + 2$, $\tau(i) = 1 + \beta i$.

The proof that $\overline{Q}$ Algorithm is Enhanced IBE Compatible can be found in [5].

3.7 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 key for decryption (r_j) is generated using the following algorithm:

  foreach $j \in [1, l]$ **do**
  $R_{U,j} \leftarrow H(ID_U, j) \in j(N)$
  $R_{S,j} \leftarrow H(ID_S, j) \in j(N)$
  $w \leftarrow F_K(ID, j) \in [0, 1, 2, 3]$
  choose $a_U \in [0, 1]$ such that $u^{a_U} R_{U,j} \in QR(N)$
  choose $a_S \in [0, 1]$ such that $u^{a_S} R_{S,j} \in QR(N)$
  let$[z_{U,0}, z_{U,1}, z_{U,2}, z_{U,3}]$ be the four square roots of $u^{a_U} R_{U,j} \in \mathbb{Z}_N$
  let$[z_{S,0}, z_{S,1}, z_{S,2}, z_{S,3}]$ be the four square roots of $u^{a_S} R_{S,j} \in \mathbb{Z}_N$
  $r_{U,j} \leftarrow z_{U,w}$
  $r_{S,j} \leftarrow z_{S,w}$
  end

  The decryption key for User is $d_{U,ID} \leftarrow (P, r_{U,1}, ..., r_{U,L})$ and the decryption key for the SEM is $d_{S,ID} \leftarrow (P, r_{S,1}, ..., r_{S,L})$.
- Enc(P, ID_U, ID_S, m): Generate a random value $i \leftarrow \mathbb{Z}_N$ and calculate $I \leftarrow i^2$ and then encrypt $m \in [-1, 1]^L$ using P as follows:

  $\tau(i) \leftarrow \overline{Q}(N, u, 1, I)$
  $k \leftarrow (\frac{\tau(i)}{N})$
  foreach $j \in [1, L]$ **do**
  $R_{U,j} \leftarrow H(ID_U, j) \in j(N)$
  $R_{S,j} \leftarrow H(ID_S, j) \in j(N)$
  $[x_{U,j}, y_{U,j}] \leftarrow \overline{Q}(N, u, R_{U,j}, I)$
  $[x_{S,j}, y_{S,j}] \leftarrow \overline{Q}(N, u, R_{S,j}, I)$
  $g_{U,j}(i) \leftarrow 2y_{U,j}i + 2$
  $g_{S,j}(i) \leftarrow 2y_{S,j}i + 2$
  $c_j \leftarrow m_j.(\frac{g_{U,j}(i)}{N}).(\frac{g_{S,j}(i)}{N})$
  $c \leftarrow c_1.....c_L$
  end

  The ciphertext is (I, k, c).

– Decrypt(C, d_{ID}): To decrypt a ciphertext $C = (I, K, c)$ for User with public key ID_U, User sends C to the SEM. The SEM then does the following:
– if User is revoked, the SEM returns $\perp$.
– if User is not revoked, the SEM Calculates c_U as follows:

foreach $j \in [1, L]$ **do**
  $R_{S,j} \leftarrow H(ID_S, j) \in j(N)$
  if $r^2_{S,j} = R_{S,j}$ **then**
    $[x_{S,j}, y_{S,j}] \leftarrow \overline{Q}(N, u, R_{S,j}, I)$
    $f_j \leftarrow x_{S,j} r_{S,j} + 1$
    $c_{U,j} \leftarrow c_j.(\frac{f_j}{N})$
  end
  if $\overline{r}^2_{S,j} = uR_{S,j}$ **then**
    $[x_{S,j}, y_{S,j}, \alpha, \beta] \leftarrow \overline{Q}(N, u, R_{S,j}, I)$
    $\overline{f}_j \leftarrow 1 + I^{2j-1} y_{S,j}\beta + \alpha x_j \overline{r}_{S,j}$
    $c_{U,j} \leftarrow c_j.(\frac{\overline{f}_j}{N})$
  end
end

and returns $C_U = (I, K, c_U)$ to User. Then User decrypts C_U as follows:

foreach $j \in [1, L]$ **do**
  $R_{U,j} \leftarrow H(ID_U, j) \in j(N)$
  if $r^2_{U,j} = R_{U,j}$ **then**
    $[x_{U,j}, y_{U,j}] \leftarrow \overline{Q}(N, u, R_{U,j}, I)$
    $f_j \leftarrow x_{U,j} r_{U,j} + 1$
    $m_j \leftarrow c_j.(\frac{f_j}{N})$
  end
  if $\overline{r}^2_{U,j} = uR_{U,j}$ **then**
    $[x_{U,j}, y_{U,j}, \alpha, \beta] \leftarrow \overline{Q}(N, u, R_{U,j}, I)$
    $\overline{f}_j \leftarrow 1 + I^{2j-1} y_{U,j}\beta + \alpha x_j \overline{r}_{U,j}$
    $m_j \leftarrow c_j.k.(\frac{\overline{f}_j}{N})$
  end
end

This concludes BGH_{BGH}.

3.8 Security Proof

Lemma 2. *Let A be an Anon-IND-CPA adversary that has advantage ϵ against GME_{BGH}. This adversary A can be a revoked client or hacked SEM. Then, there is an Anon-IND-CPA adversary B with the same probability ϵ against the BGH system.*

Proof. If an adversary A simulates the role of a revoked user, then he plays Type 1 with the challenger. The ciphertext sent to the adversary is $c_j \leftarrow m_j.(\frac{g_{U,j}(i)}{N}).(\frac{g_{S,j}(i)}{N})$. The adversary then partially decrypts it using his

secret key r_U to get $c_{S,j} \leftarrow m_j.(\frac{g_{S,j}(i)}{N})$, which is the message m encrypted by BGH using SEM's ID. This also can be applied for a hacked SEM.

Remark: Using the same encryption system for both the SEM and the users has a unique advantage. If roles of the SEM and a user are exchanged, the system will not be effected. For example, if the employee responsible for the SEM is promoted or fired and another employee becomes the one responsible for the SEM, all we have to do is assign the ID for the SEM to the the new employee's ID. On the other hand, the system will be vulnerable to escrow. This implementation is more suitable for closed environments, such as a company. If escrow is really a serious security concern, however, the public parameters can be generated using two PKGs, one for the users and the other for the SEMs.

4 Conclusion

In this paper, we present 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 [3] and [12] 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. Boneh, D., Boyen, X.: Efficient selective-ID secure identity-based encryption without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004), http://www.cs.stanford.edu/~xb/eurocrypt04b/
2. Boneh, D., Ding, X., Tsudik, G.: Fine-grained control of security capabilities. ACM Trans. Internet Technol. 4(1), 60–82 (2004)
3. Boneh, D., Ding, X., Tsudik, G., Wong, C.M.: A method for fast revocation of public key certificates and security capabilities. In: Proceedings of the 10th Conference on USENIX Security Symposium, SSYM 2001, vol. 10, p. 22. USENIX Association, Berkeley (2001)
4. Boneh, D., Franklin, M.: Identity-based encryption from the weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)
5. Boneh, D., Gentry, C., Hamburg, M.: Space-efficient identity based encryption without pairings. In: Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2007, pp. 647–657. IEEE Computer Society, Washington, DC (2007)
6. Cocks, C.: An identity based encryption scheme based on quadratic residues. In: Honary, B. (ed.) Cryptography and Coding 2001. LNCS, vol. 2260, pp. 360–363. Springer, Heidelberg (2001)
7. Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., Polk, W.: Rfc5280: Internet x.509 public key infrastructure certificate and certificate revocation list (crl) profile (May 2008)

8. Ding, X., Tsudik, G.: Simple identity-based cryptography with mediated RSA. In: Joye, M. (ed.) CT-RSA 2003. LNCS, vol. 2612, pp. 193–210. Springer, Heidelberg (2003)
9. Aiello, W., Lodha, S., Ostrovsky, R.: Fast digital identity revocation. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 137–152. Springer, Heidelberg (1998)
10. Gassko, I., Gemmell, P.S., MacKenzie, P.: Efficient and fresh certification. In: Imai, H., Zheng, Y. (eds.) PKC 2000. LNCS, vol. 1751, pp. 342–353. Springer, Heidelberg (2000)
11. Gentry, C.: Certificate-based encryption and the certificate revocation problem. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 272–293. Springer, Heidelberg (2003)
12. Gentry, C.: Practical identity-based encryption without random oracles. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 445–464. Springer, Heidelberg (2006)
13. Myers, M., Ankney, R., Malpani, A., Galperin, S., Adams, C.: Rfc 2560: Internet public key infrastructure online certificate status protocol - ocsp
14. Micali, S.: Efficient certificate revocation (1996)
15. Micali, S.: Novomodo: Scalable certificate validation and simplified pki management. In: 1st Annual PKI Research Workshop (2002)
16. Naor, M., Nissim, K.: Certificate revocation and certificate update. IEEE Journal on Selected Areas in Communications 18(4), 561–570 (2000)
17. Housley, R., Polk, W., Ford, W., Solo, D.: Rfc3280: Internet x.509 public key infrastructure certificate and certificate revocation list (crl) profile (April 2002)
18. Sakai, K.O.R., Kasahara, M.: Cryptosystems based on pairing. In: Symposium on Cryptography and Information Security (SCIS 2000), Japan (2000)
19. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)
20. Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005)