Efficient Revocation of Dynamic Security Privileges in Hierarchically Structured Communities

Deholo Nali, Ali Miri, and Carlisle Adams
School of Information Technology and Engineering
University of Ottawa
Email: {deholo,cadams,samiri}@site.uottawa.ca

***Abstract*— This paper presents the first mediated hierarchical identity-based encryption and signature schemes. Both schemes are designed to support information access control in hierarchically structured communities of users whose access privileges change very dynamically.**

I. Introduction

In order to effectively deal with their complexity, large organizations (including hospitals, banks, universities, and governmental and military bodies) are often hierarchically structured. Moreover, their members are often dynamically granted *temporary* access to confidential information, on a *need-to-know* basis. Consequently, cryptographic schemes are needed both to handle the structural disposition of these organizations' members and to allow fast revocation of the members' privileges. Concretely, it should be possible for certain privileged members to access confidential documents which are already accessible by less privileged members, but it should not be possible for (recent and old) *revoked* users to access confidential documents to which they formerly had access.

To deal with these requirements, certificate-based public-key infrastructures are commonly used. In such infrastructures, digital certificates bind identities to their public keys. Thus the authenticated and up-to-date certificate of a person is needed in order to encrypt information for that person. Moreover, a whole privilege management infrastructure (*PMI*) is needed to handle the hierarchical disposition of organization's members. Unfortunately, the management (creation, storage, deployment, revocation, updating) of digital certificates can be extremely cumbersome, in some environments.

In this paper, we describe a mediated hierarchical identity-based encryption scheme and a related signature scheme, as two alternative and efficient ways to support information access control in hierarchically structured communities of users whose access privileges change very dynamically. The next two subsections discuss work related to our cryptographic schemes, and outline the contributions of this paper.

A. Related Work

Identity-based cryptography was originally suggested by Shamir [10], in 1984, as a method to avoid the exchange of public keys and the use of digital certificates in public key infrastructures. The underlying idea is to derive public keys directly from users' identifiers, such as email addresses, social insurance numbers or IP-addresses. On the other hand, each identity-based private key is generated as a combination of a user public key and a system-level secret key that is kept private by a central trusted authority. Thus, public keys can be derived from any string and private keys must be securely generated and delivered to users by a central authority (also known as the *Private Key Generator* or *PKG*). Since 2001, much research has been conducted to investigate applications and extensions of the first efficient identity-based encryption and signature schemes, which were presented by Boneh and Franklin [4] on this same year.

One such extension is the hierarchical identity-based encryption scheme (*HIDE*) of Gentry and Silverberg [7], which has a related hierarchical signature scheme [7]. These two schemes allow secure communications in hierarchically structured communities of users. The schemes' major benefit is to provide a method for a central trusted authority to delegate the computation and delivery of user decryption keys to lower level authorities. Consequently, the schemes are very scalable to large structured communities of users. However, some limitations of the schemes should be noted. First, they induce a linear expansion of ciphertexts' and signatures' length with respect to the depth of recipients and signers in the hierarchy. Second, the schemes are restricted to tree-shaped hierarchies (as opposed to general-graph type of hierarchies). Third, they cannot quickly disable the capability of a revoked user to decrypt ciphertexts which are encrypted after the user has been revoked, without binding cryptographic keys with very short time periods. Fourth, they are unable to prevent revoked users who have accumulated valid decryption keys from continuing to use them, after revocation, in order to decrypt previously accessible documents. Fifth, they suffer from the fact that the compromise of a *PKG* jeopardizes the confidentiality of all ciphertexts intended for its lower *PKGs*. In particular, the last two limitations can be disastrous when access to confidential data is to be controlled in a distributed system, in which the protection of *all PKGs* (especially *all* high-level *PKGs*) is difficult to ensure.

Another extension of Boneh and Franklin's schemes is the mediated identity-based encryption scheme of Libert and Quisquater [8], which also has a related signature scheme. Mediated cryptography [5], [3] is predicated on the idea that each user's private key can be split into two random shares, one of which is given to the user and the other to

an online entity called a *security mediator* (*SEM*). Thus, any decryption or signature must be performed as a cooperation between a user and his/her associated *SEM*. Moreover, *SEMs* are typically associated with a small number of local users and can be instantly instructed to revoke any of their associated users' decryption or signature privileges. Consequently, the *SEM* architecture allows fine-grained instant[1] revocation of user security capabilities. Moreover, this architecture allows a system's *PKG* to delegate its decryption- and signature-related duties to the *SEMs*. Furthermore, the schemes of Libert and Quisquater allow *SEMs* to be semi-trusted entities, in the sense that their compromise only affects the users associated with them. Recently, Baek and Zheng [1] have improved the encryption scheme of Libert and Quisquater [8] by ensuring that the compromise of a user's private key share does not compromise the confidentiality of plaintexts encrypted for that user, provided the associated *SEM*'s private key share is not compromised. However, one limitation of the above mediated schemes is their inefficiency at handling hierarchically structured user communities.

B. Contributions

Our main contribution is to extend and combine the above-mentioned hierarchical schemes and the mediated identity-based scheme of Baek and Zheng [1], by designing the first *mediated hierarchical identity-based* encryption and signature schemes. Due to the hierarchical nature of our schemes and to the instant revocation capability offered by the *SEM* architecture, we obtain a method to cryptographically support information access control in hierarchically structured communities of users whose access privileges change very dynamically.

A challenging aspect of our work consists in designing a mechanism which allows *SEMs* to generate the private key shares of other (children) *SEMs*. This allows to build *SEM* hierarchies and solves the inefficiency of previous identity-based mediated schemes at handling user hierarchies. Indeed, these previous schemes required system *PKGs* to generate the private key shares of *all SEMs*, which imposed both a high computational burden on the *PKGs* and a high communication cost between *PKGs* and *SEMs*.

Note that, to combine Baek and Zheng's mediated scheme with Gentry and Silverberg's hierarchical schemes, one needs to deal with two different ciphertext formats. Indeed, the first scheme uses a format that allows to publicly check the validity of ciphertexts, while the second scheme uses a different ciphertext format based on Fujisaki-Okamoto padding [6]. In order to design our schemes, we decided to use Gentry and Silverberg's methodology as a tool to expand Baek and Zheng's type of ciphertexts, so that the requirements of the hierarchical setting could be met. Then, for security proofs [9], we adapted arguments found in [7] and [2], and leveraged the security results of [1].

Note also that, in the *SEM* paradigm, each *SEM* is associated with many users. Consequently, one cannot simply duplicate a user hierarchy to produce a *SEM* hierarchy, since such a duplication would require the use of too many *SEMs*. Hence, one has to deal with an inherent asymmetry between the user hierarchy and the *SEM* hierarchy, in order to design efficient mediated hierarchical schemes.

Finally, remark that our schemes have the following two limitations (inherited from the hierarchical schemes of Gentry and Silverberg): first, the linear length expansion of ciphertexts and signatures with respect to the depth of recipients and signers in the hierarchy; second, the restriction to tree-shaped hierarchies.

C. Outline

The remainder of this paper is organized as follows: section II presents our encryption and signature schemes. Section III discusses operational aspects of the schemes and outlines their security guarantees. Finally, section IV concludes the paper. Fundamental mathematical definitions are presented in the Appendix.

II. Encryption and Signature Schemes

This section describes a novel mediated hierarchical identity-based encryption scheme denoted by *mHIDE*. The scheme assumes the existence of a two disjoint tree-shaped hierarchies of *SEMs* and users, respectively. To each *SEM* is associated a set of users. Moreover, the root node of the two hierarchies is a common entity, called the *root PKG*, denoted by $rPKG$. The set of nodes located at the t^{th} level of both hierarchies is denoted by $Level_t$. Furthermore, every entity located at $Level_t$ can be identified by a tuple $\overline{ID}_t = (ID_1, \cdots, ID_t)$ corresponding to the path $rPKG$-ID_1-$\cdots$-ID_t from the root *PKG* to the entity.

A. Encryption Scheme

- ***Instance Generator***. This procedure, denoted by $\mathcal{IG}$, is a randomized algorithm which takes a security parameter $k > 0$, runs in time polynomial in k, and outputs not only the description of two groups $\mathcal{G}_1$ and $\mathcal{G}_2$ of prime order q, but also the description of an admissible [4] (i.e. Bilinear, non-degenerate and computable) pairing $\hat{e} : \mathcal{G}_1 \times \mathcal{G}_1 \rightarrow \mathcal{G}_2$ with respect to which $\mathcal{G}_1$ and $\mathcal{G}_2$ are Gap-Diffie-Hellman group [4]. See appendix for precise definitions.
- ***Root Setup***. Given a security parameter $k > 0$, the root *PKG*:
  1) runs $\mathcal{IG}$ with input k in order to generate groups $\mathcal{G}_1$ and $\mathcal{G}_2$ of prime order q and an admissible pairing $\hat{e} : \mathcal{G}_1 \times \mathcal{G}_1 \rightarrow \mathcal{G}_2$.
  2) chooses an arbitrary generator $P_0 \in \mathcal{G}_1$.
  3) picks, randomly and uniformly[2], $s_{(0,user)}, s_{(0,sem)} \in \mathbb{Z}_q^*$ and

[1]Note that, by *instant revocation*, we mean *revocation without delay* after an authoritative entity becomes aware that privileges must be revoked. By way of comparison, note that technologies such as *Certificate Revocation Lists* and their variants (e.g. *delta*-CRLs) do not provide such an *instantaneous* revocation of user security capabilities.

[2]In the sequel, we shall use the notation $x \in_R X$ to indicate that the element x is chosen uniformly at random from the set X.

computes $s_0 = s_{(0,user)} + s_{(0,sem)}$, $Q_{(0,user)} = s_{(0,user)}P_0$, $Q_{(0,sem)} = s_{(0,sem)}P_0$ and $Q_0 = Q_{(0,user)} + Q_{(0,sem)}$.

4) computes $n = poly(k)$, where $poly$ is a polynomial over the positive integers.
5) chooses cryptographic hash functions:
$\mathcal{H}_1 : \{0,1\}^* \rightarrow \mathcal{G}_1$, $\mathcal{H}_2 : \mathcal{G}_2 \rightarrow \{0,1\}^n$
$\mathcal{H}_3 : (\mathcal{G}_1^*)^\dagger \times \{0,1\}^n \rightarrow \mathcal{G}_1^*$
where $\mathcal{G}_1^*$ denotes the set of non-identity elements of $\mathcal{G}_1$ and $(\mathcal{G}_1^*)^\dagger$ denotes the set of arbitrary long tuples whose entries are in $\mathcal{G}_1^*$.

The message space is $\mathcal{M} = \{0,1\}^n$ and the ciphertext space is $\mathcal{C} = \mathcal{G}_1^t \times \{0,1\}^n$, where t is the recipient's level in the tree hierarchy. The system's public parameters are $params = (n, \hat{e}, P_0, Q_0, \mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3)$; they must be certified by a certification authority (*CA*). The *root PKG* keeps $s_0, s_{(0,user)}, s_{(0,sem)}$ secret.

- ***Key Generation***.
  - *Root-Level Key Generation*.
    - *User Key Generation*:
      For each first level user (ID_1), *rPKG*:
      · computes $P_1 = \mathcal{H}_1(ID_1) \in \mathcal{G}_1$.
      · computes $S_{(1,user)} = s_{(0,user)}P_1$ and secretly gives $S_{(1,user)}$ to the child ID_1.
    - *SEM Key Generation*:
      For each first level *SEM* (say SEM_1) associated with a first level user (say ID_1), *rPKG*:
      · computes $P_1 = \mathcal{H}_1(ID_1) \in \mathcal{G}_1$.
      · computes $S_{(1,sem)} = s_{(0,sem)}P_1$ and secretly gives $S_{(1,sem)}$ to SEM_1.
  - *Lower-Level Key Generation*.
    Let $User_a \in Level_a$ be a user identified by the tuple $(ID_1, \cdots, ID_a)$ (where $a \geq 1$).
    - *User Key Generation*:
      For each of its child-user $\overline{ID}_{a+1} = (ID_1, \cdots, ID_a, ID_{a+1})$, $User_a$:
      · picks $s_{(a,user)} \in_R \mathbb{Z}_q^*$;
      · computes $P_{a+1} = \mathcal{H}_1(ID_1||\cdots||ID_a||ID_{a+1}) \in \mathcal{G}_1$;
      · computes $S_{(a+1,user)} = S_{(a,user)} + s_{(a,user)}P_{a+1}$ and secretly gives $S_{(a+1,user)}$ to its child $\overline{ID}_{a+1} = (ID_1, \cdots, ID_a, ID_{a+1})$;
      · computes $Q_{(a,user)} = s_{(a,user)}P_0$, and, for $1 \leq j \leq a$, publicly gives $Q_{(j,user)}$ to its child $\overline{ID}_{a+1} = (ID_1, \cdots, ID_a, ID_{a+1})$.
    - *SEM Key Generation*:
      Let SEM_a be the *SEM* associated with $User_a$, and let SEM_{a+1} be the child-*SEM* of SEM_a associated with $\overline{ID}_{a+1} = (ID_1, \cdots, ID_a, ID_{a+1})$. Then SEM_a:
      · picks $s_{(a,sem)} \in_R \mathbb{Z}_q^*$;
      · computes $P_{a+1} = \mathcal{H}_1(ID_1||\cdots||ID_a||ID_{a+1}) \in \mathcal{G}_1$;
      · computes $S_{(a+1,sem)} = S_{(a,sem)} + s_{(a,sem)}P_{a+1}$ and secretly gives $S_{(a+1,sem)}$ to SEM_{a+1};
      · computes $Q_{(a,sem)} = s_{(a,sem)}P_0$, and, for $1 \leq j \leq a$, publicly gives $Q_{(j,sem)}$ to its child SEM_{a+1}.
- ***Encryption***. Combining the encryption schemes of [7] and [1], the encryption of a message $m \in \mathcal{M}$ for $\overline{ID}_a = (ID_1, \cdots, ID_a)$ is performed as follows:
  1) compute $P_i = \mathcal{H}_1(ID_1||\cdots||ID_i)$ for $1 \leq i \leq a$;
  2) choose $r \in_R \{0,1\}^n$;
  3) compute $g = \hat{e}(Q_0, P_1)$ and $V = m \oplus \mathcal{H}_2(g^r)$;
  4) compute $U_0 = rP_0$, and, if $a \geq 2$, compute $U_i = rP_i$ for $2 \leq i \leq a$;
  5) – if $a \geq 2$,
     a) compute $W = r\mathcal{H}_3(U_0, U_2, U_3, \cdots, U_a, V)$, and
     b) set the ciphertext to be $c = (U_0, U_2, U_3, \cdots, U_a, V, W)$;
     – otherwise,
     a) compute $W = r\mathcal{H}_3(U_0, V)$, and
     b) set the ciphertext to be $c = (U_0, V, W)$.
- ***Decryption***. Upon reception of a ciphertext $c = (U_0, U_2, U_3, \cdots, U_b, V, W)$ (or $c = (U_0, V, W)$), the decryptor $\overline{ID}_a = (ID_1, \cdots, ID_a)$ proceeds as follows:
  1) Accept c if $a = 1$ and $c = (U_0, V, W)$, or if $a > 1$ and $(U_0, U_2, U_3, \cdots, U_b) \in \mathcal{G}_1^a$. Otherwise, reject c.
  2) computes $h_3 = \mathcal{H}_3(U_0, U_2, U_3, \cdots, U_a, V)$;
  3) if $\hat{e}(P, W) \neq \hat{e}(U, h_3)$, then $\overline{ID}_a$ returns $(\overline{ID}_a, \text{“}Invalid\ Ciphertext\text{”})$;
  4) otherwise, send $(c, \overline{ID}_a)$ to SEM_a (the *SEM* associated with $\overline{ID}_a$), so that the following be performed, in parallel:
     – SEM_a:
     a) checks whether any of $\overline{ID}_a$'s rights to decrypt c have been revoked; if so, return $(SEM_a, \text{“}\overline{ID}_a\ revoked\text{”})$ to $\overline{ID}_a$;
     b) computes $h_3 = \mathcal{H}_3(U_0, U_2, U_3, \cdots, U_a, V)$;
     c) ∗ if $\hat{e}(P, W) \neq \hat{e}(U, h_3)$, then SEM_a returns $(SEM_a, \text{“}Invalid\ Ciphertext\text{”})$;
        ∗ otherwise, SEM_a computes and returns it to $\overline{ID}_a$:
        · $g_{sem_a}^r = \hat{e}(U_0, S_{(a,sem)})$ if $a = 1$.
        · $g_{sem_a}^r = \hat{e}(U_0, S_{(a,sem)}) \cdot \left(\prod_{i=2}^{a} \hat{e}(Q_{(i-1,sem)}, U_i)\right)^{-1}$ if $a > 1$.
     – $\overline{ID}_a$:
     a) computes
        ∗ $g_{user_a}^r = \hat{e}(U_0, S_{(a,user)})$ if $a = 1$;
        ∗ $g_{user_a}^r = \hat{e}(U_0, S_{(a,user)}) \cdot \left(\prod_{i=2}^{a} \hat{e}(Q_{(i-1,user)}, U_i)\right)^{-1}$ if $a > 1$;
  5) – If $\overline{ID}_a$ receives either $(SEM_a, \text{“}\overline{ID}_a\ revoked\text{”})$ or $(SEM_a, \text{“}Invalid\ \ Ciphertext\text{”})$ from SEM_a, then $\overline{ID}_a$ terminates the decryption process, and returns either “$\overline{ID}_a\ revoked$” or “$Invalid\ Ciphertext$” accordingly;

– otherwise, $\overline{ID}_a$:
  a) receives $g^r_{sem_a}$ from SEM_a,
  b) computes $g^r = g^r_{sem_a} g^r_{user_a}$,
  c) computes $m = V \oplus \mathcal{H}_2(g^r)$, and
  d) outputs m as the decryption of c.

B. Signature Scheme

- ***Instance Generator***. As in the *mHIDE* Scheme.
- ***Root Setup***. As in the *mHIDE* Scheme, except that only $\mathcal{H}_1$ needs to be defined along with another cryptographic hash function $\mathcal{H}_4 : \{0,1\}^* \rightarrow \mathcal{G}_1$.
  The message space is $\mathcal{M} = \{0,1\}^n$. The signature space is $\mathcal{S} = \mathcal{G}_1^{a+1} \times \{0,1\}^*$, where a is the signer's level. The system's public parameters are $params = (n, \hat{e}, P_0, Q_0, \mathcal{H}_1, \mathcal{H}_4)$; they must be certified by a *CA*. The *root PKG* keeps $s_0, s_{(0,user)}, s_{(0,sem)}$ secret.
- ***Key Generation***. As in *mHIDE*.
  Each user $User_a$ and its associated *SEM* (denoted by SEM_a) jointly compute and publish $User_a$'s public Q-value: $Q_a = Q_{(a,user)} + Q_{(a,sem)}$. An authenticity proof $cert_{\overline{Q_a}}$ of the values $Q_1, \cdots, Q_a$ must also be issued and published by a trusted authority.
- ***Signing***. In order for a user $\overline{ID}_a = (ID_1, \cdots, ID_a)$ to sign a message $m \in \mathcal{M}$, the user proceeds as follows:
  1) compute $P_m^{\mathcal{H}_4} = \mathcal{H}_4(ID_1 || \cdots || ID_a || m)$,
  2) send $P_m^{\mathcal{H}_4}$ to SEM_a and perform the following, in parallel:
     – SEM_a:
       a) returns "$\overline{ID}_a$ $revoked$" to $\overline{ID}_a$ if $\overline{ID}_a$ is revoked,
       b) computes $Sig_{(m,sem)} = S_{(a,sem)} + s_{(a,sem)} P_m^{\mathcal{H}}$ and sends $Sig_{(m,sem)}$ back to $\overline{ID}_a$.
     – $\overline{ID}_a$:
       a) computes $Sig_{(m,user)} = S_{(a,user)} + s_{(a,user)} P_m^{\mathcal{H}}$.
       b) upon reception of $Sig_{(m,sem)}$ from SEM_a, $\overline{ID}_a$:
          i) computes $Sig_m = Sig_{(m,sem)} + Sig_{(m,user)}$,
          ii) fetches the certified Q-values Q_i for $1 \leq i \leq a$, along with their validity and authenticity proof string $cert_Q$.
          iii) returns $(Sig_m, Q_1, \cdots, Q_a, cert_{\overline{Q_a}})$ as a signature of m.
- ***Verification***. Given a signature $(Sig_m, Q_1, \cdots, Q_a, cert_{\overline{Q_a}})$ of a message m and a claimed signer $\overline{ID}_a = (ID_1, \cdots, ID_a)$, the verifier accepts the signature if and only if $cert_{\overline{Q_a}}$ is valid and

$$\hat{e}(P_0, Sig_m) = \hat{e}(Q_0, P_1)\hat{e}(Q_a, P_m^{\mathcal{H}}) \prod_{i=2}^{a} \hat{e}(Q_{i-1}, P_i).$$

The above scheme allows any member of the user hierarchy to sign a document m. Such a signature is then verifiable using the system global parameters and the signer's public key. In the next section, we discuss operational aspects and the security guarantees of mediated hierarchical identity-based cryptosystems.

III. Discussion

A. Operational Aspects

In this section, we explain how a company X can both use a mediated hierarchical identity-based (*mHID*) cryptosystem, and integrate the branch B of an existing *mHID* system denoted by CS. Suppose that X is structured in two divisions X_1 and X_2. Then X can proceed as follows:

First, X sends B any required information to integrate CS. Assuming that B authorizes the integration of X, then: (a) B asks its associated *SEM* to create a child-*SEM* SEM_X for X. (b) If B's *SEM* agrees, X becomes a child-user of B associated with SEM_X. (c) Both X and SEM_X receive all required parameters from B and B's *SEM* respectively. Finally, X creates two children-*SEMs* SEM_{X_1} and SEM_{X_2}.

Let now u be a member of X_1. Then u asks X to be integrated to the company's *mHID* cryptosystem. If the above request is authorized, X makes u its child-user and asks SEM_X to associate u with its appropriate child i.e. SEM_{X_1}. Thus, if u is successfully associated with SEM_{X_1}, then u receives from X all required parameters.

Assume now that an entity v wants to encrypt a message m in such a way that u's privileges are required to recover m. Then u retrieves the public parameters of CS and encrypts m using the mediated hierarchical identity-based encryption scheme (*mHIDE*) described in section II-A.

Suppose now that u wants to sign a message m addressed to an entity v. Then u retrieves the public parameters of CS (through those of X) and signs m using the mediated hierarchical identity-based signature scheme (*mHIDS*) described in section II-B. v may then use the public parameters of CS to verify the signature.

B. Security

It can be formally shown (see [9] for a detailed security analysis) that our cryptographic schemes achieve the current highest levels of formal security guarantees, for public-key encryption and signature schemes. In a nutshell, the encryption scheme is *semantically secure* with respect to *adaptive chosen ciphertext attacks* (assuming the difficulty of the Computational Bilinear Diffie-Hellman problem), and the signature scheme is *strongly existentially unforgeable* (assuming the difficulty of the computational Diffie-Hellman problem). Detailed definitions of these problems are stated in the Appendix.

IV. Conclusion

In this paper, we presented the first mediated hierarchical identity-based encryption and signature schemes, which support information access control in hierarchically structured communities of users whose privileges change very dynamically. We discussed operational aspects of the schemes, and noted that the schemes offer strong formal security guarantees.

The schemes require users to cooperate with an on-line entity called the security mediator (*SEM*), in order to complete decryption and signing procedures. This allows to instantaneously revoke user security privileges by instructing their associated *SEM* to stop cooperating with these users. This also allows to guarantee that signers have all required privilege in order to sign a document. The scheme are scalable to (large) hierarchically structured user communities, by providing a mechanism for *SEMs* to generate the private key shares of other (children) *SEMs*. Moreover, the encryption scheme prevents the compromise of a *SEM* to affect the confidentiality of ciphertexts addressed to users associated with anyone of the *SEM*'s descendants.

Designing our schemes required to describe a mechanism enabling *SEMs* to generate the keys of their children. This was achieved using Gentry and Siverberg's methodology to support identity-based hierarchical encryption [7]. Designing our schemes also required to deal with the ciphertext format difference between two previous schemes which we sought to extend and combine. This difference was addressed by using the ciphertext format of Baek and Zheng's mediated scheme [1], as a basis, and by modifying this format in the same way that Gentry and Silverberg [7] modified Boneh and Franklin's original ciphertext format [4], in order to meet the requirements of hierarchical settings. Furthermore, the design of our schemes required to deal with an inherent asymmetry between user hierarchies and *SEM* hierarchies (since each *SEM* can be associated with many users). This smaller challenge was overcome by labelling *SEMs* in such a way that each *SEM* may correspond to many users.

More work is needed to address the following two limitations which our schemes inherit from the hierarchical identity-based schemes of Gentry and Silverberg: 1) the linear expansion of ciphertexts and signatures with respect to the depth of recipients and signers in the hierarchy, and 2) the restriction to tree-shaped hierarchies (as opposed to general-graph type of hierarchies). An extension of this work is to design a *mHIDE* scheme for hierarchies of arbitrary graph shapes. Another extension consists in designing a threshold *HIDE* scheme which generalizes our mediated scheme.

ACKNOWLEDGMENT

This work was partially supported by NSERC.

APPENDIX

This section outlines fundamental mathematical definitions used in the paper.

- **Bilinear Pairing**: Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be two Abelian groups of prime order q, where $\mathcal{G}_1$ is additive and $\mathcal{G}_2$ is multiplicative. Let $P_0 \in \mathcal{G}_1^*$ be a generator of $\mathcal{G}_1$. A Bilinear Pairing $\hat{e}$ is a map $\hat{e} : \mathcal{G}_1 \times \mathcal{G}_1 \rightarrow \mathcal{G}_2$ such that $\hat{e}(aP_0, bP_0) = \hat{e}(P_0, P_0)^{ab}$ for all $a, b \in \mathbb{Z}_q^*$.
- **Admissible Pairing**: Let $\mathcal{G}_1, \mathcal{G}_2, q, P_0$ be defined as above, and $\hat{e} : \mathcal{G}_1 \times \mathcal{G}_1 \rightarrow \mathcal{G}_2$ be a map. The map $\hat{e}$ is said to be an *admissible pairing* if it is a *non-degenerate* (i.e. $\hat{e}$ does not send all pairs of points in $\mathcal{G}_1 \times \mathcal{G}_1$ to the identity in $\mathcal{G}_2$), computable (i.e. $\hat{e}$ efficiently computes the image of any pair of points in $\mathcal{G}_1 \times \mathcal{G}_1$) *Bilinear pairing*.
- **Computational BDH Problem**: Let $\mathcal{G}_1, \mathcal{G}_2, q, P_0$ be defined as above, and $\hat{e}$ be a Bilinear Pairing. Let $\mathcal{A}$ be an attacker modelled as a probabilistic Turing machine. The *computational Bilinear Diffie-Hellman* (*CBDH*) problem is that in which $\mathcal{A}$ is to compute $\hat{e}(P_0, P_0)^{abc}$ given $(\mathcal{G}_1, q, P, aP_0, bP_0, cP_0)$ and a security parameter k, where $a, b, c \in \mathbb{Z}_q^*$ are unknown. The success (or *advantage*) of $\mathcal{A}$ is then defined as the function $Succ_{\mathcal{G}_1,\mathcal{A}}^{BDH}(k) = Pr[\mathcal{A}$ outputs $\hat{e}(P_0, P_0)^{abc}]$.
- **Decisional BDH Problem**: Let $\mathcal{G}_1, \mathcal{G}_2, q, P_0, \hat{e}$ and $\mathcal{A}$ be defined as above. The *decisional Bilinear Diffie-Hellman* (*DBDH*) problem is that in which $\mathcal{A}$ is to decide whether $\hat{e}(P_0, P_0)^{ab} = \hat{e}(P_0, P_0)^c$, given $(\mathcal{G}_1, q, P, aP_0, bP_0, cP_0)$ and a security parameter k, where $a, b, c \in \mathbb{Z}_q^*$ are unknown. The success (or *advantage*) of $\mathcal{A}$ is then defined as the function $Succ_{\mathcal{G}_1,\mathcal{A}}^{DBDH}(k) = Pr[\mathcal{A}$ accurately determines whether or not $\hat{e}(P_0, P_0)^{ab} = \hat{e}(P_0, P_0)^c]$.
- **Gap DH Groups**: Let $\mathcal{G}_1, \mathcal{G}_2, q, P_0, \hat{e}$ and $\mathcal{A}$ be defined as above. $\mathcal{G}_1$ and $\mathcal{G}_2$ are said to be a *Gap-Diffie-Hellman groups* if, with respect to any Bilinear pairing $\hat{e} : \mathcal{G}_1 \times \mathcal{G}_1 \rightarrow \mathcal{G}_2$, the *CBDH* problem is hard while the *DBDH* problem is solvable in polynomial time.
- **Computational DH Problem**: Let $\mathcal{G}_1, q, P_0$ and $\mathcal{A}$ be defined as above. The *computational Diffie-Hellman* (*CDH*) problem is that in which $\mathcal{A}$ is to compute abP_0 given (aP_0, bP_0) and a security parameter k, where $a, b, c \in \mathbb{Z}_q^*$ are unknown.

REFERENCES

[1] Joonsang Baek and Yuliang Zheng, *Identity-Based Threshold Decryption*, Proceedings of the 7th International Workshop on Theory and Practice in Public Key Cryptography (PKC'04), Lecture Notes in Computer Science, vol. 2947, Springer-Verlag, 2004, pp. 262–276.

[2] Benot Libert and Jean-Jacques Quisquater, *The Exact Security of an Identity Based Signature and its Applications*, Cryptology ePrint Archive, Report 2004/102, 2004.

[3] Dan Boneh, Xuhua Ding, and Gene Tsudik, *Fine-grained control of security capabilities*, ACM Trans. Inter. Tech. **4** (2004), no. 1, 60–82.

[4] Dan Boneh and Matthew K. Franklin, *Identity-Based Encryption from the Weil Pairing*, Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology, Springer-Verlag, 2001, pp. 213–229.

[5] Gene Tsudik Dan Boneh, Xuhua Ding and C. Wong, *A Method for Fast revocation of Public Key Certificates and Security Capabilities*, Proceedings of the 10th USENIX Security Symposium, USENIX, 2001, pp. 297–308.

[6] Eiichiro Fujisaki and Tatsuaki Okamoto, *Secure Integration of Asymmetric and Symmetric Encryption Schemes*, Lecture Notes in Computer Science **1666** (1999), 537–554.

[7] Craig Gentry and Alice Silverberg, *Hierarchical ID-Based Cryptography*, Proceedings of the 8th International Conference on the Theory and Application of Cryptology and Information Security, Springer-Verlag, 2002, pp. 548–566.

[8] Benoît Libert and Jean-Jacques Quisquater, *Efficient Revocation and Threshold Pairing Based Cryptosystems*, Proceedings of the twenty-second annual symposium on Principles of distributed computing, ACM Press, 2003, pp. 163–171.

[9] Deholo Nali, Ali Miri, and Carlisle Adams, *Mediated Hierarchical Identity-Based Cryptography*, In preparation (2004).

[10] Adi Shamir, *Identity-Based Cryptosystems and Signature Schemes*, Proceedings of CRYPTO 84 on Advances in cryptology, Springer-Verlag New York, Inc., 1985, pp. 47–53.