An extended abstract of this paper appears in ICICS 2007, S. Qing, H. Imai, and G. Wang (Eds.), volume 4861 of LNCS, pp. 69-82, Sringer-Verlag, 2007.

# Short Group Signature without Random Oracles

Xiaohui Liang liangxh127@sjtu.edu.cn
Zhenfu Cao* zfcao@cs.sjtu.edu.cn
Jun Shao chn.junshao@gmail.com
Huang Lin faustlin@sjtu.edu.cn

Department of Computer Science and Engineering
Shanghai Jiao Tong University

December 6, 2007

**Abstract**

We construct a short group signature which is proven secure without random oracles. By making certain reasonable assumptions and applying the technique of non-interactive proof system, we prove that our scheme is full anonymity and full traceability. Compared with other related works, such as BW06 [9], BW07 [10], ours is more practical due to the short size of both public key and group signature.

**Keywords:** Group signature, standard model, short signature, non-interactive proof system

## 1 Introduction

Group signature is a useful cryptographical tool, which is widely discussed in the literature and also has many potential applications, such as network meeting, online business, and software trading. The similar requirement of these applications is to allow a member to sign a message on behalf of the group, and still remain anonymous within the group. Group signature schemes meet this requirement by providing anonymity and traceability at the same time, that is, a group signature can be related with its signer's identity only by a party who possesses an open authority. In such environment, there exists a group manager to distribute certificates, open authority and other group settings. If one group member generates a group signature, anyone can only verify the signature by using group public parameters. When some dissention happens, an opener finds out the real signer's identity. In this way, group members could protect their privacy.

In 1991, Chaum and van Heyst [13] firstly proposed group signature. Then, many papers on this subject proposed various of approaches to give a secure and practical group signature scheme. There exist a lot of practical schemes secure in the random oracle model [2, 7, 19, 20, 21]. However, Canetti, Goldreich and Halevi [11, 12, 14] have shown that security in the random oracle model does not imply the security in the real world in that a signature scheme can be secure in the random oracle model and yet be broken without violating any particular intractability assumption, and without breaking the underlying hash functions.

Therefore, to design a secure group signature scheme in the standard model becomes an open and interesting research problem. Bellare et. al. introduced security definitions for group signatures and proposed a scheme based on trapdoor permutation in [6]. Furthermore, Bellare et. [8] strengthened the security model to include dynamic enrollment of members. After that, Groth [15] also gave a group signature scheme based on bilinear groups which is proven CCA secure in the standard model under the decisional-linear assumption. Their scheme was constructed in the BSZ-model [8], but still the size of group signature is enormous.

*Corresponding Author.

Ateniese, Camenisch, Hohenberger and de Medeiros [1] designed a practical group signature with high efficiency which is also secure in the standard model. The drawback of their scheme was that if the user's private key is exposed, it can be used to trace the identity of the user's past signatures. Unfortunately, this is not according with BSZ-models, and needs to be prevented.

Boyen and Waters [9] suggested group signature schemes that are secure in a restricted version of the BMW-model [6], where the anonymity of the members relies on the adversary can not make any query on the tracing of group signature. The size of both public parameter and group signature are both logarithm of identity and message. Afterwards, they [10] proposed a group signature scheme the signature of which is of constant size (only 6 group elements). However, the size of public parameter is still logarithm of identity. Groth also presented a group signature scheme [16] based on non-interactive witness indistinguishable proof of knowledge and other existing tools, which enhances the security notion of BW [9, 10]. We will compare our scheme with theirs in Section 7, specifically.

*Our Contribution*

We propose a new group signature scheme secure in the standard model. We use short signature [3] and non-interactive proof system [17] as the foundation to construct ours. Then we prove our scheme is secure in a restricted BMW-model. Furthermore, the sizes of both public parameter and group signature are reduced to two constants, and are shorter than that of both schemes in [10, 16]. To the best of our knowledge, our group signature is the shortest one secure in the standard model. Besides, the overall computational cost of our scheme is low. Therefore, our scheme is more practical compared with the others.

*Roadmap*

The rest of this paper is arranged as follows. In next section, we provide the preliminaries of our scheme including bilinear groups of composite order and complexity assumptions. In Section 3, we describe the formal models of group signature scheme. Then we propose the two-level signature and group signature schemes in Section 4 & 5, respectively. We give the details of security proofs in Section 6. Finally, we draw comparisons between ours and other related works in Section 7 and summarize our paper in Section 8.

# 2 Preliminaries

## 2.1 Bilinear Groups of Composite Order

Recently, a lot of cryptographical schemes are based on bilinear groups of composite order. We briefly review some notions about it from other related works [5, 18, 17, 9, 10].

Consider two finite cyclic groups $G$ and $G_T$ having the same order $n$, where $n = pq$, $p, q$ are large primes and $p \neq q$. It is clear that the respective group operation is efficiently computable. Assume that there exists an efficiently computable mapping $e : G \times G \rightarrow G_T$, called a bilinear map or pairing, with the following properties.

- Bilinear: For any $g, h \in G$, and $a, b \in Z_n$, we have $e(g^a, h^b) = e(g, h)^{ab}$, where the product in the exponent is defined modulo $n$.
- Non-degenerate: $\exists\ g \in G$ such that $e(g, g)$ has order $n$ in $G_T$. In other words, $e(g, g)$ is a generator of $G_T$, whereas $g$ generates $G$.
- Computable: There is an efficient algorithm to compute $e(g, h)$ for all $g, h \in G$.

## 2.2 Complexity Assumptions

Before describing our new group signature, we firstly introduce the complexity assumptions from other related works [5, 18, 17] and then propose new ones.

**Subgroup Decision Problem.** The subgroup decision problem in $G$ of composite order $n = pq$ is defined as follows: given a tuple $(n, G, G_T, e)$ and an element $h$ selected at random either from $G$ or from $G_q$ as input, output 1 if $h \in G_q$; else output 0.

**Definition 1** *We say that the subgroup decision assumption holds for generator $\mathcal{G}_{BGN}$ if any non-uniform polynomial time adversary $\mathcal{A}$ we have*

$$\begin{aligned} & \Pr[(p, q, G, G_T, e, g) \leftarrow \mathcal{G}_{BGN}(1^k); n = pq; r \leftarrow \mathbb{Z}_n^*; h = g^r : A(n, G, G_T, e, g, h) = 1] \\ = \; & \Pr[(p, q, G, G_T, e, g) \leftarrow \mathcal{G}_{BGN}(1^k); n = pq; r \leftarrow \mathbb{Z}_n^*; h = g^{pr} : A(n, G, G_T, e, g, h) = 1] \end{aligned}$$

**$l$-Strong Diffie-Hellman Problem.** [3] The $l$-SDH problem in $G$ is defined as follows: given a $(l+1)$-tuple $(g, g^x, g^{(x^2)}, ..., g^{(x^l)})$ as input, output a pair $(c, g^{\frac{1}{x+c}})$ where $c \in Z_p^*$. An algorithm $\mathcal{A}$ has advantage $\epsilon$ in solving $l$-SDH in $G$ if

$$\Pr[\mathcal{A}(g, g^x, g^{(x^2)}, ..., g^{(x^l)}) = (c, g^{\frac{1}{x+c}})] \geq \epsilon$$

**Definition 2** *We say that the $(l, t, \epsilon)$-SDH assumption holds in $G$ if no $t$-time algorithm has advantage at least $\epsilon$ in solving the $l$-SDH problem in $G$.*

Now, we give some new assumptions and observe the relationship between them.
**l-One More Strong Diffie-Hellman Problem. ($l$-OMSDH)** The $l$-one more strong Diffie-Hellman problem in the prime-order bilinear group $G$ is defined as follows: on input two generators $g, g^x \in G$, and $l$ distinct tuples $(c_i, g^{\frac{1}{x+c_i}})$, where $c_i \in \mathbb{Z}_n, i \in \{1, 2, ..., l\}$, outputs another tuple $(c, g^{\frac{1}{x+c}})$ distinct of all the others. An algorithm $\mathcal{A}$ has advantage $\epsilon$ in solving $l$-OMSDH in $G$ if

$$\Pr[\mathcal{A}(g, g^x, c_1, g^{\frac{1}{x+c_1}}, c_2, g^{\frac{1}{x+c_2}}, ..., c_l, g^{\frac{1}{x+c_l}}) = (c, g^{\frac{1}{x+c}})] \geq \epsilon,$$

$$\text{where } c \neq c_i, \text{for } i = 1, 2, ..., l$$

**Definition 3** *We say that the $(l, t, \epsilon)$-*OMSDH *assumption holds in $G$ if no $t$-time algorithm has advantage at least $\epsilon$ in solving the $l$-*OMSDH *problem in $G$.*

**$l$-Modified One More Strong Diffie-Hellman Problem. ($l$-MOMSDH)** The $l$-modified one more strong Diffie-Hellman problem in the prime-order bilinear group $G$ is defined as follows: on input three generators $g, g^x, u \in G$, and $l$ distinct tuples $(c_i, g^{\frac{1}{x+c_i}})$, where $c_i \in \mathbb{Z}_n, i \in \{1, 2, ..., l\}$, outputs another tuple $(g^c, g^{\frac{1}{x+c}}, u^{\frac{1}{c+m}}, m)$ where $c \notin \{c_1, ..., c_i\}$ and $m \in_R \mathbb{Z}$. An algorithm $\mathcal{A}$ has advantage $\epsilon$ in solving $l$-SDH in $G$ if

$$\Pr[\mathcal{A}(g, g^x, u, c_1, g^{\frac{1}{x+c_1}}, c_2, g^{\frac{1}{x+c_2}}, ..., c_l, g^{\frac{1}{x+c_l}}) = (g^c, g^{\frac{1}{x+c}}, u^{\frac{1}{c+m}}, m)] \geq \epsilon,$$

$$\text{where } c \neq c_i, \text{for } i = 1, 2, ..., l$$

**Definition 4** *We say that the $(l, t, \epsilon)$-*MOMSDH *assumption holds in $G$ if no $t$-time algorithm has advantage at least $\epsilon$ in solving the $l$-*MOMSDH *problem in $G$.*

It is easy to see that for any $l \geq 1$, hardness of the $l$-SDH problem implies hardness of the $l$-OMSDH problem in the same group. Meanwhile, hardness of the $l$-MOMSDH problem implies hardness of the $l$-OMSDH problem in the same group. We claim all of these problems are hard to solve. To be more convincing, the discussion of these problems is in the Appendix B and the proof of them will appear in the full paper.

## 3 Formal Models of Group Signatures

In this section, we introduce some basic models and security issues which have been defined in the papers [9, 10]. A group signature scheme consists of the following algorithms: `Setup`, `Join`, `Sign`, `Verify` and `Trace`.

1. `Setup`: Taking as input the system security parameter $\lambda$, this algorithm outputs group's public parameter `PP` for verifying signatures, a master key `MK` for enrolling group members, and a tracing key `TK` for identifying signers.

2. `Join`: Taking as input the master key `MK` and an identity $id$, and outputs a unique identifier $s_{id}$ and a private signing key $K_{id}$ which is to be given to the user. That is: $K_{id} \leftarrow \texttt{Join}(\texttt{PP}, \texttt{MK}, id)$.

3. `Sign`: Taking as input a user's private key $K_{id}$ and a message $M$, and outputs a group signature $\sigma$. That is $\sigma \leftarrow \texttt{Sign}(\texttt{PP}, K_{id}, M)$.

4. `Verify`: Taking as input a message $M$, a signature $\sigma$, and the group's public parameter `PP`, and outputs valid or invalid. That is "Valid" or "Invalid" $\leftarrow \texttt{Verify}(\texttt{PP}, \sigma, M)$.

5. `Trace`: Taking as input a group signature $\sigma$, and a tracing key `TK`, and outputs an identity $s_{id}$ or $\perp$. That is $s_{id}$ or $\perp \leftarrow \texttt{Trace}(\texttt{PP}, \sigma, \texttt{TK})$

*Consistency.* We require that the following equations hold.

$$\texttt{Verify}(\texttt{PP}, \texttt{Sign}(\texttt{PP}, K_{id}, M), M) = \text{Valid}$$

$$\texttt{Trace}(\texttt{PP}, \texttt{Sign}(\texttt{PP}, K_{id}, M), \texttt{TK}) = s_{id}$$

*Security.*

Bellare, Micciancio, and Warinschi [6] presented the fundamental properties of group signatures, which are considered to be restrictions in the following designs. The most two important properties are:

**Full Anonymity** which requires that no PPT adversary is able to find the identity of a group signature. The game could be described as follows: the adversary $\mathcal{A}$ could firstly query some private keys and some valid signatures from the simulator $\mathcal{B}$, then $\mathcal{A}$ outputs $id_1, id_2, m$ and sends them to $\mathcal{B}$. $\mathcal{B}$ randomly choose $b \in \{0, 1\}$ and generate $\sigma_b$ corresponding with $(id_b, m)$. If $\mathcal{A}$ has negligible advantage to guess the correct $b$, our group signature scheme is full anonymity (CPA). We notice that if we give the trace oracle to the adversary, the full anonymity is enhanced, which is similar with the CCA-secure notion. In this paper, we follow [10] and use non-interactive proof system to design a simple group signature in the CPA-full anonymity notion.

**Full Traceability** which requires that no forged signatures, even if there exists a coalition of users. The game could be described as follows: the adversary $\mathcal{A}$ is given group public parameters `PP` and the tracing key `TK`. Then $\mathcal{A}$ could query some private keys and some valid signatures from the simulator $\mathcal{B}$. The validity of signature and identity tracing could be checked by $\mathcal{A}$. At some point, $\mathcal{A}$ outputs a forged group signature $\sigma^*$ with its tracing identity $id^*$ and message $m^*$. The restrictions are that the private key of $id^*$ and $(id^*, m^*)$ should not be queried before. If $\mathcal{A}$ has only negligible advantage to forge a valid signature, our group signature scheme is full traceability.

We refer the reader to [6] for more details of these and related notion.

# 4 Hierarchical Signatures

We build a hierarchical signature scheme based on the short signature proposed by BB04 [3]. To implement a group signature scheme, we construct a short two-level hierarchical signature with existential unforgeability against chosen message attacks based on $l$-MOMSDH assumption. The first level can be seen as a certificate that signed by the group manager, while the second level is a short signature on message $m$.

## 4.1 Two-level Signature Scheme

Let $\lambda$ be the security parameter. Suppose the user's identity $id$ and the message $M$ are chosen from $\{0,1\}^\lambda$. We build a group $G$ with order $n = pq$ and record $g, u$ as two generators of $G_p$, where $G_p$ is a subgroup of $G$ with order $p$. There exists a bilinear map $e$ from $G \times G$ to $G_T$.

**Setup**$(1^\lambda)$: It firstly generates the master key $\texttt{MK} = z \in \mathbb{Z}_p$ and calculates the public parameter $\texttt{PP} = \{g, Z = g^z, u\} \in G_p^3$. Moreover, it generates the public collision-resistant hash function $H : \{0,1\}^\lambda \to \mathbb{Z}_p$.

**Extract**$(\mathbf{PP}, \mathbf{MK}, id)$: To create a private key for an user, it chooses a secret value $s_{id} \in \mathbb{Z}_p$ and return:

$$K_{id} = (K_1, K_2) = (s_{id}, g^{\frac{1}{z+s_{id}}}) \in \mathbb{Z}_p \times G_p$$

Note that the value $z + s_{id}$ must lie in $\mathbb{Z}_p^*$

**Sign**$(\mathbf{PP}, K_{id}, M)$: To sign a message $M \in \{0,1\}^\lambda$, the algorithm generates and outputs:

$$\sigma = (\sigma_1, \sigma_2, \sigma_3) = (g^{s_{id}}, g^{\frac{1}{z+s_{id}}}, u^{\frac{1}{s_{id}+H(M)}})$$

Note that the probability of $s_{id} + H(M) \equiv 0 \ (mod\ p)$ is negligible.

**Verify**$(\mathbf{PP}, M, \sigma)$: To verify whether the signature $\sigma$ is valid for a message $M$, the algorithm checks:

$$e(Z\sigma_1, \sigma_2) \stackrel{?}{=} e(g, g)$$

$$e(g^{H(M)}\sigma_1, \sigma_3) \stackrel{?}{=} e(g, u)$$

If the above two equations hold, the verifier outputs valid; else outputs invalid.

Notice that this signature scheme doesn't reveal the user's identity, the private key generator could record the mapping from $id$ to $s_{id}$. However, the signatures signed by one user can be easily linked with invariant values $\sigma_1, \sigma_2$. We modified two-level hierarchical signature scheme to group signature which achieves unlinkability and anonymity by using non-interactive proof system mentioned in G07 [16].

## 4.2 Existential Unforgeability

The two-level signature scheme proposed above is existential unforgeable against chosen message attacks. We review the short group signature in BB04, and prove the security issues based on the hardness of $q$-SDH and $l$-MOMSDH problems.

**Theorem 1** *Our two-level signature scheme is $(t, q_e, q_s, \epsilon)$-secure against existential forgery under a chosen message attack provided that $(t', q, \epsilon_{qSDH})$-SDH assumption and $(t'', l, \epsilon_{MOMSDH})$-MOMSDH assumption hold in $G_p$, where*

$$\epsilon \leq 2q_s\epsilon_{qSDH} + 2\epsilon_{MOMSDH} \text{ and } t \approx max(t', t''),\ q \geq q_s + 1 \text{ and } l \geq q_e + q_s$$

**Proof.** See Appendix A.

# 5 Proposed Group Signature

We now present the group signature scheme in details.

## 5.1 Schemes

The group signature scheme is described as the following algorithms. Figure 1. presents the scheme executed by three parties: group manager, user and verifier.

**Setup**$(1^\lambda)$: The input is a security parameter $1^\lambda$. Suppose the maximum group members $2^k$ and the signing message in $\{0,1\}^m$, where $k = O(\lambda), m = O(\lambda)$. It firstly chooses $n = pq$ where $p, q$ are random primes of bit size $\lceil log_2 p \rceil, \lceil log_2 q \rceil = \Theta(\lambda) > k$. We builds a cyclic bilinear group $G$ and its subgroup $G_p$ and $G_q$ of respective order $p$ and $q$. Denote $g, u$ a generator of $G$ and $h$ a generator of $G_q$. Next, The algorithm picks a random exponents $z \in \mathbb{Z}_n^*$, and defines $Z = g^z \in G$. Additionally, a public collision-resistant hash function $H$ is from $\{0,1\}^m$ to $Z_n$.

The public parameters consist,

$$\mathtt{PP} = (g, h, Z, u) \in G \times G_q \times G \times G$$

The master key $\mathtt{MK}$ and the tracing key $\mathtt{TK}$ are

$$\mathtt{MK} = z \in \mathbb{Z}_n^*, \mathtt{TK} = q \in \mathbb{Z}$$

**Join**$(\mathtt{PP}, \mathtt{MK}, id)$: The input is a user's identity $id$. The algorithm assigns a secret unique value $s_{id} \in \mathbb{Z}_n$ for tracing purpose. Then the secret key is constructed as:

$$K_{id} = (K_1, K_2) = (s_{id}, g^{\frac{1}{z+s_{id}}})$$

The user may verify that the key is well formed by checking

$$e(Zg^{K_1}, K_2) \stackrel{?}{=} e(g, g)$$

**Sign**$(\mathtt{PP}, id, K_{id}, M)$: To sign a message $M \in \{0,1\}^m$, a user parses $K_{id} = (K_1, K_2)$ and computes a two-level signature:

$$\rho = (\rho_1, \rho_2, \rho_3) = (g^{K_1}, K_2, u^{\frac{1}{K_1 + H(M)}})$$

Notice that, $\rho$ does not satisfy the anonymity and unlinkability to anyone, since $\rho_1, \rho_2$ are unchangeable for each signature. So, by adopting the same approaches from BW07 [10] and G07 [16], we let the signer choose $t_1, t_2, t_3 \in \mathbb{Z}_n$ and compute:

$$\sigma_1 = \rho_1 \cdot h^{t_1}, \sigma_2 = \rho_2 \cdot h^{t_2}, \sigma_3 = \rho_3 \cdot h^{t_3}$$

Additionally, it computes a proof:

$$\pi_1 = \rho_2^{t_1} (Z\rho_1)^{t_2} h^{t_1 t_2}, \ \pi_2 = \rho_3^{t_1} (g^{H(M)} \rho_1)^{t_3} h^{t_1 t_3}$$

The output signature is:

$$\sigma = (\sigma_1, \sigma_2, \sigma_3, \pi_1, \pi_2) \in G^5$$

**Verify**(PP, $M$, $\sigma$): To check the validity of signature $\sigma$, the verifier calculates:

$$T_1 = e(\sigma_1 Z, \sigma_2) \cdot e(g, g)^{-1}, \ T_2 = e(\sigma_1 g^{H(M)}, \sigma_3) e(g, u)^{-1}$$

Then verifies:

$$T_1 \stackrel{?}{=} e(\pi_1, h),\ T_2 \stackrel{?}{=} e(\pi_2, h)$$

If the above equations hold, the verifier outputs valid; else outputs invalid.

**Trace**$(\mathtt{PP}, \mathtt{TK}, \sigma)$: Let $\sigma$ be a valid signature, the opener parses it and finds the element $\sigma_1$. Then, to trace the identity of signer, it calculates $\sigma_1^q$ and tests:

$$(\sigma_1)^q = (g^{s_i d} \cdot h^{t_1})^q \stackrel{?}{=} (g^{s_{id}})^q$$

Since all the $(g^{s_{id}})^q$ can be pre-calculated firstly and recorded in a list by opener, the time to find the identity $id$ is linearly dependent on the number of initial users.

| $Group\ Manage$ | $User(id)$ | $Verifier$ |
|---|---|---|
| generate secret value $K_1 = s_{id}$ $K_2 = g^{\frac{1}{z+s_{id}}}$ $\xrightarrow{K_1, K_2}$ | Verifies $e(Zg^{K_1}, K_2) \stackrel{?}{=} e(g,g)$ random chooses $t_1, t_2, t_3 \in Z_q^*$ $\sigma_1 = g^{K_1} \cdot h^{t_1}$ $\sigma_2 = K_2 \cdot h^{t_2}$ $\sigma_3 = u^{\frac{1}{K_1+H(M)}} \cdot h^{t_3}$ $\pi_1 = K_2^{t_1}(Zg^{K_1})^{t_2} h^{t_1 t_2}$ $\pi_2 = u^{\frac{t_1}{K_1+H(M)}} g^{t_3(K_1+H(M))} h^{t_1 t_3}$ $\xrightarrow{\sigma_1, \sigma_2, \sigma_3, \pi_1, \pi_2}$ | Verifies $T_1 = e(\sigma_1 Z, \sigma_2) \cdot e(g,g)^{-1}$ $T_2 = e(\sigma_1 g^{H(M)}, \sigma_3) \cdot e(g,u)^{-1}$ $T_1 \stackrel{?}{=} e(\pi_1, h)$ $T_2 \stackrel{?}{=} e(\pi_2, h)$ if all pass, the signature is valid |

Figure 1: Short Group Signature Scheme

# 6 Security Analysis

We now analyze the security of our group signature scheme.

## 6.1 Full Anonymity

Since our scheme adopts the same approach from BW06 [9] and BW07 [10], we only prove the security of our group signature scheme in the anonymity game against chosen plaintext attacks. The proof sketch borrows from G07 [16]. That is, if $h$ is chosen from $G$, we achieve perfect hiding property. Meanwhile, if $h$ is chosen from $G_q$, we achieve perfect biding property. However, the adversary $\mathcal{A}$ can not distinguish these two different environment, since subgroup decision problem is unsolvable in polynomial time. Therefore, we give the following theorem.

**Theorem 2** *Suppose no $t$-time adversary can solve the subgroup decision problem with advantage at least $\epsilon$. Then for every $t'$-time adversary $\mathcal{A}$ to break the full anonymity, we have that $Adv_{\mathcal{A}} < 2\epsilon_{sub}$, where $t \approx t'$.*

To prove the above theorem, the two lemmas are necessary.

**Lemma 1** *For all $t'$-time adversaries $\mathcal{A}$, the probability to distinguish the true environment and the simulated environment is negligible. That is $Adv_{\mathcal{A}} - Adv_{\mathcal{A},\mathcal{S}} < 2\epsilon_{sub}$*

*Proof.* Suppose there is a simulator $\mathcal{B}$ trying to solve subgroup problem. Upon receiving a tuple $(e, G, G_T, n, h)$, he wants to find out whether $h \in G_q$ or not. Firstly, he setups the group signature scheme by choosing the public parameters exactly as in the group signature scheme. Then $\mathcal{B}$ publishes them to the adversary $\mathcal{A}$. Whether $h$ is chosen from $G_q$ or not, $\mathcal{B}$ can always answer all queries, since it knows the master key. If $h \in_R G_q$, then the simulated environment is identical to the actual one.

At some point, the adversary $\mathcal{A}$ chooses a message $M$ and two identities $id$ and $id'$. The constraints are the secret keys of $id$ and $id'$, and $(M, id), (M, id')$ should not be queried before. Then, $B$ outputs the challenge signature with $(M, id^*)$, where $id^* \in \{id, id'\}$. After that, $A$ outputs its guess. If it is correct, $B$ outputs 1; else outputs 0. Denote by $Adv_{\mathcal{B}}$ the advantage of the simulator $\mathcal{B}$ in the subgroup decision game. Assume that

$$\Pr[h \in G] = \Pr[h \in G_q] = \frac{1}{2}$$

we obtain that,

$$\begin{aligned} Adv_{\mathcal{A}} - Adv_{\mathcal{A},\mathcal{S}} &= Pr[b = 1 | h \in G_q] - Pr[b = 1 | h \in G] \\ &= 2Pr[b = 1, h \in G_q] - 2Pr[b = 1, h \in G] \\ &= 2Adv_{\mathcal{B}} \\ &< 2\epsilon_{sub} \end{aligned}$$

Thus, under our subgroup decision assumption in Section 2.2, the probability to distinguish the actual environment and the simulated one is negligible. ■

**Lemma 2** *For any adversary $\mathcal{A}$, we have $Adv_{\mathcal{A},\mathcal{S}} = 0$*

*Proof.* The proof sketch is similar to that of BW07 [10] and G07 [16]. We prove that when $h$ is chosen uniformly from $G$ at random, instead of $G_q$, the adversary $\mathcal{A}$ can not sense the identity from the challenge signature. Although the tracing value $s_{id}$ may have been used to answer previous signing queries on $(id, M)$ and $(id', M)$, the challenge signature is statistically independent of the real identity.

To proceed, we write the challenge ciphertext is $\sigma = (\sigma_1, \sigma_2, \sigma_3, \pi_1, \pi_2)$.

Since the signature $\sigma_1, \sigma_2, \sigma_3$ is blinded with random number $t_1, t_2, t_3 \in G$, respectively, they reveal nothing about the identity. Then, we give two signatures: $\sigma$ with $(id, M)$ and $\sigma'$ with $(id', M)$ and analyze two tuples $\pi = (\pi_1, \pi_2)$, $\pi' = (\pi'_1, \pi'_2)$.

If $\sigma_1 = \sigma'_1, \sigma_2 = \sigma'_2$, and $\sigma_3 = \sigma'_3$, we show that $\pi$ and $\pi'$ do not reveal the identity either.

$$\begin{aligned} g^{s_{id}} h^{t_1} &= g^{s_{id'}} h^{t'_1} \\ g^{\frac{1}{z+s_{id}}} h^{t_2} &= g^{\frac{1}{z+s_{id'}}} h^{t'_2} \\ u^{\frac{1}{s_{id}+H(M)}} h^{t_3} &= u^{\frac{1}{s_{id'}+H(M)}} h^{t'_3} \end{aligned}$$

Suppose $h = g^{\eta}, h = u^{\xi}, \varepsilon = \frac{z+s_{id}}{z+s_{id'}}, \tau = \frac{s_{id}+H(M)}{s_{id'}+H(M)}$, we obtain that

$$\begin{aligned} t'_1 &= t_1 + \frac{s_{id} - s_{id'}}{\eta} \\ t'_2 &= t_2 + \frac{1}{\eta}(\frac{1}{z+s_{id}} - \frac{1}{z+s_{id'}}) = t_2 + \frac{1-\varepsilon}{\eta(z+s_{id})} \\ t'_3 &= t_3 + \frac{1}{\xi}(\frac{1}{s_{id}+H(M)} - \frac{1}{s_{id'}+H(M)}) = t_3 + \frac{1-\tau}{\xi(s_{id}+H(M))} \end{aligned}$$

Now, we need to show that $\pi_1, \pi_2$ do not reveal any information about the user's identity. From the adversary's point of view, we see that $\pi_1, \pi_2, \pi_1', \pi_2'$ satisfy,

$$
\begin{aligned}
\pi_1' &= g^{\frac{t_1'}{z+s_{id'}}} g^{(z+s_{id'})t_2'} h^{t_1' t_2'} \\
log_g \pi_1' &= \frac{t_1 + \frac{s_{id}-s_{id'}}{\eta}}{z+s_{id'}} + (z+s_{id'})(t_2 + \frac{1-\varepsilon}{\eta(z+s_{id})}) + \eta(t_1 + \frac{s_{id}-s_{id'}}{\eta})(t_2 + \frac{1-\varepsilon}{\eta(z+s_{id})}) \\
&= \frac{t_1}{z+s_{id'}} + \frac{s_{id}-s_{id'}}{\eta(z+s_{id'})} + zt_2 + s_{id'}t_2 + \frac{(1-\varepsilon)(z+s_{id'})}{\eta(z+s_{id})} + \eta t_1 t_2 + s_{id}t_2 - s_{id'}t_2 + \\
&\quad \frac{t_1(1-\varepsilon)}{z+s_{id}} + \frac{(1-\varepsilon)(s_{id}-s_{id'})}{\eta(z+s_{id})} \\
&= \frac{t_1}{z+s_{id}} + (z+s_{id})t_2 + \eta t_1 t_2 \\
\pi_1' &= g^{\frac{t_1}{z+s_{id}} + (z+s_{id})t_2 + \eta t_1 t_2} \\
&= g^{\frac{t_1}{z+s_{id}}} g^{(z+s_{id})t_2} h^{t_1 t_2} \\
&= \pi_1
\end{aligned}
$$

---

$$
\begin{aligned}
\pi_2' &= u^{\frac{t_1'}{s_{id'}+H(M)}} g^{(s_{id'}+H(M))t_3'} h^{t_1' t_3'} \\
log_g \pi_2' &= (\frac{t_1 + \frac{s_{id}-s_{id'}}{\eta}}{s_{id'}+H(M)}) \cdot \frac{\eta}{\xi} + (s_{id'}+H(M))(t_3 + \frac{1-\tau}{\xi(s_{id}+H(M))}) \\
&\quad + \eta(t_1 + \frac{s_{id}-s_{id'}}{\eta})(t_3 + \frac{1-\tau}{\xi(s_{id}+H(M))}) \\
&= \frac{t_1}{s_{id'}+H(M)} \cdot \frac{\eta}{\xi} + \frac{s_{id}-s_{id'}}{\xi(s_{id'}+H(M))} + H(M)t_3 + s_{id'}t_3 + \frac{(1-\tau)(s_{id'}+H(M))}{\xi(s_{id}+H(M))} \\
&\quad + \eta t_1 t_3 + s_{id}t_3 - s_{id'}t_3 + \frac{t_1(1-\tau)}{s_{id}+H(M)} \cdot \frac{\eta}{\xi} + \frac{(1-\tau)(s_{id}-s_{id'})}{\xi(s_{id}+H(M))} \\
&= \frac{t_1}{s_{id}+H(M)} \cdot \frac{\eta}{\xi} + (s_{id}+H(M))t_3 + \eta t_1 t_3 \\
\pi_2' &= g^{\frac{t_1}{s_{id}+H(M)} \cdot \frac{\eta}{\xi} + (s_{id}+H(M))t_3 + \eta t_1 t_3} \\
&= u^{\frac{t_1}{s_{id}+H(M)}} g^{(s_{id}+H(M))t_3} h^{t_1 t_3} \\
&= \pi_2
\end{aligned}
$$

Therefore, $\pi_1, \pi_2$ is identical to $\pi_1', \pi_2'$. The challenge signature $\sigma$ does not reveal the identity $id$, though the simulator uses $s_{id}$ to generate it. Hence, we claim that the adversary $\mathcal{A}$ in the anonymity game under the simulated environment has negligible advantage to guess the correct identity. ■

### 6.2 Full Traceability

We prove that our group signature is existential unforgeability based on the security of two-level signature scheme proposed in Section 4.1.

**Theorem 3** *If there exists a $(t, \epsilon)$ adversary for the full traceability game against the group signature scheme, then there exists a $(t', \epsilon)$ chosen message existential unforgeability adversary against the two-level signature scheme, where $t \approx t'$.*

*Proof.* We note that our group signature scheme is an extension form of our two-level signature scheme by adding some random number on the signing and verifying equations. Intuitively, we prove that our group signature is secure against chosen message attack by using two-level signature's unforgeability.

Suppose there exists a simulator $\mathcal{B}$, who interacts with the adversary $\mathcal{A}$ and wants to break two-level signature scheme. Then, $\mathcal{B}$ executes the following algorithms and plays a game with $\mathcal{A}$.

In `Setup` algorithm, $\mathcal{B}$ runs two-level signature `Setup`, generates public parameters and publishes them. Furthermore, $\mathcal{B}$ deliveries $\mathtt{TK} = q$ to $\mathcal{A}$, and $\mathcal{A}$ is entitled the authority to tracing authority.

$\mathcal{A}$ queries a secret key on $id$ to $\mathcal{B}$. To answer this request, $\mathcal{B}$ queries the key extraction oracle of two-level signature scheme and obtains the user's secret key $K_{id}$. Then $\mathcal{B}$ sends $K_{id}$ to $\mathcal{A}$.

$\mathcal{A}$ queries a signature on $(id, M)$ to $\mathcal{B}$. $\mathcal{B}$ directly queries the signing oracle of two-level signature scheme and obtains $\sigma = (\sigma_1^\star, \sigma_2^\star, \sigma_3^\star)$ corresponding with $(id, M)$. Then, $\mathcal{B}$ randomly choose $t_1, t_2, t_3$, and generates the

group signature,

$$\sigma = (\sigma_1^\star \cdot h^{t_1}, \sigma_2^\star \cdot h^{t_2}, \sigma_3^\star \cdot h^{t_3}, (\sigma_2^\star)^{t_1}(Z\sigma_1^\star)^{t_2} h^{t_1 t_2}, (\sigma_3^\star)^{t_1}(g^{H(M)}\sigma_1^\star)^{t_3} h^{t_1 t_3}) \quad (1)$$

We could see that this is a valid group signature. After receiving the responding signature. $\mathcal{A}$ could check its validity by using PP and trace its identity by using $\mathtt{TK} = q$. These verification equations are correct.

At some point, $\mathcal{A}$ outputs its forgery signature $\sigma^* = (\sigma_1^*, \sigma_2^*, \sigma_3^*, \pi_1^*, \pi_2^*)$ with $(id^*, M^*)$. According to the game's constraints, $id^*$ should be excluded from key extraction queries and $(id^*, M^*)$ should not be queried from signing oracle before.

Then, $\mathcal{B}$ generates $\lambda$ which satisfies $\lambda \equiv 1 \pmod{p}$ and $\lambda \equiv 0 \pmod{q}$. Then, from $\pi_1^*, \pi_2^*$ and the verification equations, we obtain:

$$\begin{aligned} e(\sigma_1^* Z, \sigma_2^*) \cdot e(g,g)^{-1} &= e(\pi_1^*, h) \\ e(\sigma_1^* g^{H(M^*)}, \sigma_3^*) e(g,g)^{-1} &= e(\pi_2^*, h) \end{aligned}$$

And we use $\lambda$ to obtain:

$$\begin{aligned} e(\sigma_1^{*\lambda} Z, \sigma_2^{*\lambda}) &= e(g,g) \\ e(\sigma_1^{*\lambda} g^{H(M^*)}, \sigma_3^{*\lambda}) &= e(g,g) \end{aligned}$$

Since $(\sigma_1^{*\lambda}, \sigma_2^{*\lambda}, \sigma_3^{*\lambda})$ pass the verification equations of two-level signature scheme in Section 4.1, they are a forged two-level signature, which means $\mathcal{B}$ successfully breaks the unforgeability of two-level signature scheme. Thus, Theorem 3 has been proved.

■

By combining with Theorem 2 and Theorem 3, we prove our scheme to have full anonymity and full traceability in the standard model.

# 7 Comparison

In this section, we compare our group signature with others. Boyen and Waters [9] proposed a nice group signature based on the Waters's identity-based signature [22]. However, the hierachical identity-based signature in that scheme leads logarithmic size of both group public key and group signature. Then, Boyen and Waters [10] improved the signature to be constant size. Furthermore, we propose a new group signature to achieve constant size of both public key and signature. We could see the details in table 1. ($M \in \{0,1\}^m$, $id \in \{0,1\}^k$):

Table 1: Comparisons on size in Group Signatures

| | BW06 [9] | BW07 [10] | Our Scheme |
|---|---|---|---|
| **Public Key** | $(k+m+3)\|G\|$ $+\|G_q\| + \|G_T\|$ | $(m+4)\|G\|$ $+\|G_q\| + \|G_T\|$ | $3\|G\| + \|G_q\|$ |
| **Master Key** | $\|G\|$ | $\|G\| + \|\mathbb{Z}_n\|$ | $\|\mathbb{Z}_n\|$ |
| **User Key** | $3\|G\|$ | $3\|G\|$ | $\|G\| + \|\mathbb{Z}_n\|$ |
| **Signature** | $(2k+3)\|G\|$ | $6\|G\|$ | $5\|G\|$ |

More than that, we continue to compare the computational cost on every participant in these group signature schemes. In Table 2, we note that $\mathbf{T_{Exp}}, \mathbf{T_{Pair}}, \mathbf{T_{Mul}}$ to represent the time for one modular exponentiation, one bilinear pairing computation, and one group multiplication, respectively. Certainly, our approach reduces the computational cost and enhances the whole efficiency.

Table 2: Comparisons on computational cost in Group Signatures

| | BW06 [9] | BW07 [10] | Our Scheme |
|---|---|---|---|
| **Join** | $3\mathbf{T_{Exp}} + (k+2)\mathbf{T_{Mul}}$ | $3\mathbf{T_{Exp}}$ | $\mathbf{T_{Exp}}$ |
| **Sign** | $(2k{+}5)\mathbf{T_{Exp}}{+}(3k{+}m{+}6)\mathbf{T_{Mul}}$ | $12\mathbf{T_{Exp}} + (m+10)\mathbf{T_{Mul}}$ | $11\mathbf{T_{Exp}} + 8\mathbf{T_{Mul}}$ |
| **Verify** | $(2k + 3)\mathbf{T_{Pair}} + (2k + m + 4)\mathbf{T_{Mul}}$ | $6\mathbf{T_{Pair}} + 3\mathbf{T_{Exp}} + (m + 5)\mathbf{T_{Mul}}$ | $6\mathbf{T_{Pair}} + 3\mathbf{T_{Exp}} + 4\mathbf{T_{Mul}}$ |
| **Open** | $k\mathbf{T_{Exp}}$ | $\mathbf{T_{Exp}}$ | $\mathbf{T_{Exp}}$ |
| **Exhaustively Search** | No | Yes | Yes |

Recently, Groth [16] proposed a group signature scheme with full anonymity (CCA) in the standard model. His scheme adopts the existing tools, including certisignature scheme, strong one-time signature scheme, non-interactive proofs system for bilinear groups, selective-tag weakly CCA-secure encryption, but it increases the size and computational cost. The total size of a group signature is 50 group elements in $G$. In case full anonymity (CPA) is sufficient, the signature is reduced to 30 group elements. Thus, taking efficiency into consideration, our scheme is better.

## 8 Conclusion

In this paper, we proposed a practical group signature scheme, which has shorter sizes of both public key and signature than that of the other existing schemes. Since we adopted the approach of short signature proposed by BB04 [3] and non-interactive proof system [17], we proved the security of ours without random oracles, including full anonymity and full traceability. Furthermore, our scheme reduces the computational cost on both user and verifier sides. In the future work, we should improve ours on the full anonymity security in the CCA notion without random oracles and develop other practical group signature schemes based on weaker assumptions.

## References


[1] Giuseppe Ateniese, Jan Camenisch, Susan Hohenberger, and Breno de Medeiros. Practical group signatures without random oracles. Cryptology ePrint Archieve, Report 2005/385, 2005. http://eprint.iacr.org/. 1

[2] Giuseppe Ateniese, Jan Camenisch, Marc Joye, and Gene Tsudik. A practical and provably secure coalition-resistant group signature scheme. In Proceedings of *CRYPTO 2000*, volume 1880 of *LNCS*, pages 255-270. Springer-Verlag, 2000. 1

[3] Dan Boneh and Xavier Boyen. Short Signatures Without Random Oracles. In Proceedings of *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 56-73. Springer-Verlag, 2004. 1, 2.2, 4, 8, 8

[4] Matt Blaze, Gerrit Bleumer, and Martin Strauss. Divertible protocols and atomic proxy cryptography. In *EUROCRYPT 1998*, volume 1403 of *LNCS*, pages 127-144. Springer-Verlag, 1998.

[5] Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. Evaluating 2-dnf formulas on ciphertexts. In Proceedings of *TCC 2005*, volume 3378 of *LNCS*, pages 325-341. Springer-Verlag, 2005. 2.1, 2.2

[6] Mihir Bellare, Daniele Micciancio, and Bogdan Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In Proceedings of *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 614-629. Springer-Verlag, 2003. 1, 3

[7] Dan Boneh and Hovav Shacham. Group signatures with verifier-local revocation. In Proceedings of *ACM CCS 2004*, pages 168-177. ACM Press, 2004. 1

[8] Mihir Bellare, Haixia Shi and Chong Zhang. Foundations of group signatures: The case of dynamic groups. In Proceedings of *CT-RSA 2005*, volume 3376 of *LNCS*, pages 136-153. Springer-Verlag, 2005. 1

[9] Xavier Boyen and Brent Waters. Compact Group Signatures Without Random Oracles. In Proceedings of *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 427-444. Springer-Verlag, 2006. (document), 1, 2.1, 3, 6.1, 7, 1, 2

[10] Xavier Boyen and Brent Waters. Full-Domain Subgroup Hiding and Constant-Size Group Signatures. In Proceedings of *PKC 2007*, volume 4450 of *LNCS*, pages 1-15. Springer-Verlag, 2007. (document), 1, 2.1, 3, 3, 5.1, 6.1, 6.1, 7, 1, 2

[11] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. In Proceedings of *STOC 1998*, pages 209-218, 1998. 1

[12] Ran Canetti, Oded Goldreich, and Shai Halevi. On the random-oracle methodology as applied to length-restricted signature schemes. In Proceedings of *TCC 2004*, volume 2951 of *LNCS*, , pages 40-57. Springer-Verlag, 2004. 1

[13] David Chaum and Eugène van Heyst. Group signatures. In proceedings of *EUROCRYPT 1991*, volume 547 of *LNCS*, pages 257-265. Springer-Verlag, 1991. 1

[14] Jun Furukawa and Hideki Imai. An efficient group signature scheme from bilinear maps. In Proceedings of *ACISP 2005*, volume 3574 of *LNCS*, pages 455-467. Springer-Verlag, 2005. 1

[15] Jens Groth. Simulation-sound nizk proofs for a practical language and constant size group signatures. In Proceedings of *ASIACRYPT 2006*, volume 4284 of *LNCS*, pages 444-459. Springer-Verlag, 2006. 1

[16] Jens Groth. Fully Anonymous Group Signatures without Random Oracles. Cryptology ePrint Archieve, Report 2007/186, 2004. http://eprint.iacr.org/. 1, 4.1, 5.1, 6.1, 6.1, 7

[17] Jens Groth and Amit Sahai. Efficient Non-interactive Proof Systems for Bilinear Groups. Cryptology ePrint Archieve, Report 2007/155, 2005. http://eprint.iacr.org/. 1, 2.1, 2.2, 8

[18] Jens Groth, Rafail Ostrovsky, and Amit Sahai. New Techniques for Non-interactive Zero-Knowledge. In Proceedings of *CRYPTO 2006*, volume 4117 of *LNCS*, pages 97-111. Springer-Verlag, 2006. 2.1, 2.2

[19] Aggelos Kiayias and Moti Yung. Extracting group signatures from traitor tracing schemes. In Proceedings of *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 630-648. Springer-Verlag, 2003. 1

[20] Aggelos Kiayias and Moti Yung. Group signatures: Provable security, efficient constructions and anonymity from trapdoor-holders. Cryptology ePrint Archieve, Report 2004/076, 2004. http://eprint.iacr.org/. 1

[21] Aggelos Kiayias and Moti Yung. Group signatures with efficient concurrent join. In Proceedings of *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 198-214. Springer-Verlag, 2005. 1

[22] Brent Waters. Efficient Identity-Based Encryption Without Random Oracles. In Proceedings of *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 114-127. Springer-Verlag, 2005. 7

[23] Masayuki Abe and Serge Fehr. Perfect NIZK with Adaptive Soundness. In *TCC 2007*, volume 4392 of *LNCS*, pages 118-136. Springer-Verlag, 2007. 8

# Appendix A

*Proof of Theorem 1.*
Assume $\mathcal{A}$ is a forger that $(t, q_e, q_s, \epsilon)$-breaks the two-level signature scheme. We construct an algorithm $\mathcal{B}$ that breaks one of the assumptions mentioned above.

The inputs of $\mathcal{B}$ are two instances of the $(q_s+1)$-SDH problem and $(q_e+q_s)$-MOMSDH problem which include $(g, g^{\alpha}, ..., g^{\alpha^{q_s+1}})$ and $(g, u, g^x, s_j, g^{\frac{1}{x+s_j}})$ for $j = 1, 2, ..., q_e+q_s$.

Before any oracle queries, $\mathcal{B}$ randomly sets $s_b = 0$ or $s_b = 1$ by coin tossing. $\mathcal{A}$ outputs a list of distinct $q_s - 1$ messages $m_1, ..., m_{q_s-1}$.

- `Setup`:
  - If $s_b = 0$, $\mathcal{B}$ tries to solve $(q_s+1)$-SDH problem. $\mathcal{B}$ randomly chooses a secret key $z, k \in Z_p$. By adopting the same simulating approach from [3], $\mathcal{B}$ can generate $\bar{g}, \bar{g}^{\alpha}, \bar{g}^{\frac{1}{\alpha+z}}, \bar{g}^{\frac{1}{\alpha+m_1}}, ..., \bar{g}^{\frac{1}{\alpha+m_{q_s}}}$. The public parameter is $\mathtt{PP} = (\bar{g}, Z = \bar{g}^z, \bar{u} = \bar{g}^k)$.
  - If $s_b = 1$, $\mathcal{B}$ tries to solve $(q_e+q_s)$-MOMSDH problem. The public parameter is $\mathtt{PP} = (g, Z = g^x, u)$
- `Extract` oracle:
  - $s_b = 0$: $\mathcal{B}$ randomly selects a target $ID^{\star}$. To answer a key extraction query on $ID$,
    - if $ID = ID^{\star}$, $\mathcal{B}$ aborts;
    - if $ID \neq ID^{\star}$, $\mathcal{B}$ randomly selects $s_{ID} \in Z_p$ and records $(ID, s_{ID})$ into a list $L_0$. Then, it outputs $(s_{ID}, \bar{g}^{\frac{1}{z+s_{ID}}})$.
  - $s_b = 1$: To answer a key extraction query on $ID$, $\mathcal{B}$ randomly chooses and outputs a tuple $(s_j, g^{\frac{1}{x+s_j}})$ from the instance of $(q_e+q_s)$-MOMSDH problem. Then he records $(ID, s_j)$ into a list $L_1$.
- `Sign` oracle:
  - $s_b = 0$: To answer a sign query on $ID, m_i$, where $i \in \{1, 2, ..., q_s - 1\}$,
    - if $ID = ID^{\star}$, $\mathcal{B}$ outputs $(\bar{g}^{\alpha}, \bar{g}^{\frac{1}{z+\alpha}}, \bar{g}^{\frac{k}{\alpha+m_i}}, (\bar{g}^{\alpha})^k)$.
    - if $ID \neq ID^{\star}$, $\mathcal{B}$ looks up $(ID, s_{ID})$ from $L_0$, if not exists, $\mathcal{B}$ generates a distinct value $s_{ID} \in Z_p$ and adds $(ID, s_{ID})$ into list $L_0$. Then, it outputs the signature $(\bar{g}^{s_{ID}}, \bar{g}^{\frac{1}{z+s_{ID}}}, \bar{g}^{\frac{1}{s_{ID}+m_i}}, \bar{g}^{ks_{ID}})$.
  - $s_b = 1$: To answer a sign query on $(ID, m)$, $\mathcal{B}$ looks up $(ID, s_j)$ from $L_1$, if not exists, $\mathcal{B}$ chooses a distinct value $s_j$ from the instance and adds $(ID, s_j)$ into list $L_1$. Then, it outputs the signature $(g^{s_j}, g^{\frac{1}{x+s_j}}, u^{\frac{1}{s_j+m}})$.

At some point, let $\sigma^* = (\sigma_1^*, \sigma_2^*, \sigma_3^*)$ with $(ID^*, M^*)$ be a forgery produced by $\mathcal{A}$. $ID^*$ should not be queried to the key extraction oracle and $(ID^*, M^*)$ should not be queried to the sign oracle. Then, we distinguish between two cases which are $ID^*$ have been queried with other messages $m$ except $M^*$ to the sign oracle or not.

Now, we separately analyze the cases:

- **Case 1**: $\mathcal{A}(\mathcal{A}_1)$ has already queried some two-level signatures on target $ID^*$. If $s_b = 0$,

  The forgery should satisfies the verification equations:

$$e(Z\sigma_1^*, \sigma_2^*) = e(\bar{g}, \bar{g}) \text{ and } e(\bar{g}^{H(M)}\sigma_1^*, \sigma_3^*) = e(\bar{g}, \bar{u})$$

  We write $s^* = \log_{\bar{g}}\sigma_1^*$ and obtain $\sigma_2^* = \bar{g}^{\frac{1}{z+s^*}}, \sigma_3^* = \bar{g}^{\frac{1}{s^*+H(M)}}$.

  If $ID^* = ID^\star$, $(H(M^*), \sigma_3^*)$ is $\mathcal{B}$'s solution of $(q_s+1)$-SDH problem. We note that the probability of $\Pr[ID^* = ID^\star] = 1/q_s$

  Thus, we conclude:

$$\Pr[\text{solving } \mathrm{P}_{(q_s+1)-SDH} | \mathcal{A}_1 = succ] = \frac{1}{q_s}$$

$$\Pr[\mathrm{P}_{(q_s+1)-SDH}] \geq \Pr[\mathrm{P}_{(q_s+1)-SDH}, \mathcal{A}_1 = succ]$$

$$= \Pr[\mathcal{A}_1 = succ]\Pr[\mathrm{P}_{(q_s+1)-SDH} | \mathcal{A}_1 = succ] = \frac{1}{q_s}\Pr[\mathcal{A}_1 = succ]$$

$$Adv_{\mathcal{A}_1} \leq q_s \epsilon_{(q_s+1)-SDH}$$

- **Case 2**: $\mathcal{A}(\mathcal{A}_2)$ has not queried any signature related with target $ID^*$. If $s_b = 1$,

  The forgery should satisfies the verification equations:

$$e(Z\sigma_1^*, \sigma_2^*) = e(g, g) \text{ and } e(g^{H(M)}\sigma_1^*, \sigma_3^*) = e(g, u)$$

  We write $s^* = \log_g \sigma_1^*$ and obtain $\sigma_2^* = g^{\frac{1}{z+s^*}}, \sigma_3^* = u^{\frac{1}{s^*+H(M)}}$.

  By given the inputs of $(q_s + q_e)$-MOMSDH problem, $\mathcal{B}$ could easily respond on all of $\mathcal{A}$'s queries correctly. We consider the worst case is that $\mathcal{A}$ has queried for $q_e$ distinct IDs and $q_s$ signatures with other IDs. Thus, the forgery $(\sigma_1^*, \sigma_2^*, \sigma_3^*, H(M^*))$ is $\mathcal{B}$'s solution to the $(q_s + q_e)$-MOMSDH problem.

  Therefore, we conclude:

$$Adv_{\mathcal{A}_2} \leq \epsilon_{MOMSDH}$$

Now, the simulator $\mathcal{S}$ runs either case with the same probability $\frac{1}{2}$, denoted as $\mathcal{S}_1, \mathcal{S}_2$. The adversary $\mathcal{A}$ can not distinguish either simulator from his view since both simulations are perfect.Therefore, we obtain:

$$\begin{aligned}
\epsilon = Adv_{\mathcal{A}} &\leq \Pr[\mathcal{A}_1 = succ | S_1] + \Pr[\mathcal{A}_2 = succ | S_2] \\
&= \frac{\Pr[\mathcal{A}_1 = succ, S_1]}{\Pr[S_1]} + \frac{\Pr[\mathcal{A}_2 = succ, S_2]}{\Pr[S_1]} \\
&= 2\Pr[\mathcal{A}_1 = succ, S_1] + 2\Pr[\mathcal{A}_2 = succ, S_2] \\
&\leq 2q_s\epsilon_{OMSDH} + 2\epsilon_{MOMSDH}
\end{aligned}$$

■

Similar with the paper [3], we could easily modify our two-level signature scheme to the one existential unforgeable against adaptive chosen message attack. The sizes of private key and signature in the enhanced scheme are $2|G_p| + 2|Z_p|$ and $5|G_p| + |Z_p|$, respectively. They increases a little but still be two constants.

# Appendix B

To provide more confidence in the l-$MOMSDH$ assumption, we give an argument on it and will finish the complete proof in the full paper.

Review the previous paper [23], it presents a Knowledge Exponent Assumption in a bilinear group.

**Assumption 1** *(KEA). For every non-uniform poly-time algorithm $\mathcal{A}$ there exists a non-uniform poly-time algorithm $\mathcal{X}_{\mathcal{A}}$, the* extractor*, such that*

$$P[pub \leftarrow \mathcal{BGG}, x \leftarrow Z_q, (A, \hat{A}; a) \leftarrow (\mathcal{A}||\mathcal{X}_{\mathcal{A}})(pub, g^x) : \hat{A} = A^x \wedge A \neq g^a] \leq negl$$

*Recall that $(A, \hat{A}; a) \leftarrow (\mathcal{A}||\mathcal{X}_{\mathcal{A}})(pub, g^x)$ means that $\mathcal{A}$ and $\mathcal{X}_{\mathcal{A}}$ are executed on the same input $(pub, g^x)$ and the* same random tape*, and $\mathcal{A}$ outputs $(A, \hat{A})$ whereas $\mathcal{X}_{\mathcal{A}}$ outputs $a$.*

Similarly, we define knowledge exponent inversion assumption in a bilinear group with composite order.

**Assumption 2** *(KEIA). For every non-uniform poly-time algorithm $\mathcal{A}$ there exists a non-uniform poly-time algorithm $\mathcal{X}_{\mathcal{A}}$, the* extractor*, such that*

$$P[pub \leftarrow \mathcal{BGG}, x \leftarrow Z_q, (A, \hat{A}; a) \leftarrow (\mathcal{A}||\mathcal{X}_{\mathcal{A}})(pub, g^x) : e(A, \hat{A}) = e(g, g^x) \wedge A \neq g^a] \leq negl$$

*Recall that $(A, \hat{A}; a) \leftarrow (\mathcal{A}||\mathcal{X}_{\mathcal{A}})(pub, g^x)$ means that $\mathcal{A}$ and $\mathcal{X}_{\mathcal{A}}$ are executed on the same input $(pub, g^x)$ and the* same random tape*, and $\mathcal{A}$ outputs $(A, \hat{A})$ whereas $\mathcal{X}_{\mathcal{A}}$ outputs $a$.*

Now, under the knowledge exponent inversion assumption, we prove that $l$-MOMSDH assumption is equal with $l$-OMSDH assumption. First, we prove that given $l$-MOMSDH oracle, $l$-OMSDH problem is solved under KEIA.

Given the input from $l$-OMSDH problem:

$$(g, g^x, c_1, g^{\frac{1}{x+c_1}}, c_2, g^{\frac{1}{x+c_2}}, ..., c_l, g^{\frac{1}{x+c_l}})$$

Then we randomly chooses $u \in G$, and inputs $(g, g^x, u, c_1, g^{\frac{1}{x+c_1}}, c_2, g^{\frac{1}{x+c_2}}, ..., c_l, g^{\frac{1}{x+c_l}})$ into $l$-MOMSDH oracle. Now, we have the outputs from this oracle:

$$(g^c, g^{\frac{1}{x+c}}, u^{\frac{1}{c+m}}, m), \text{ where } c \neq \{c_1, ..., c_l\}$$

Note that

$$e(g^c g^m, u^{\frac{1}{c+m}}) = e(g, u)$$

Since the inputs has no information related with element $u$, by using KEIA we have an extractor which outputs $a = c + m$. Thus, $(a - m, g^{\frac{1}{x+c}})$ is a solution to the $l$-OMSDH problem.

On the other side, we could easily see that given $l$-OMSDH oracle, $l$-MOMSDH problem is solved. Here, we give a simple analysis to our new assumption and would finish it in the full paper.