Journal of Theoretical and Applied Information Technology
10th April 2013. Vol. 50 No.1
© 2005 - 2013 JATIT & LLS. All rights reserved·

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

SHORTER VERIFIER-LOCAL REVOCATION GROUP SIGNATURE BASED ON DTDH ASSUMPTION

JINGLIANG ZHANG

Assoc. Prof., College of Mathematical Sciences, Ocean University of China, Qingdao 266100, China

E-mail: zjlmlz@yahoo.com.cn

ABSTRACT

Membership revocation is an important issue in group signature. Verifier-Local revocation (VLR for short) is an effective method. In VLR group signature, revocation messages are only sent to signature verifiers (as opposed to both signers and verifiers). Consequently there is no need to contact individual signers when some user is revoked. Since signers have no load, VLR group signature schemes are suitable for mobile environments. To decrease the user's storage load in mobile communication, shortening signature length is an essential requirement at the current research of VLR group signatures. Based on this idea, a new VLR group signature is proposed based on q-SDH assumption and DTDH assumption. Compared with the existing VLR group signatures based on DTDH assumption, the proposed scheme has the shortest signature size which reduces about 25%-54% than those of previous schemes.

Keywords: *Group Signature, Verifier-Local Revocation, DTDH Assumption*

1. INTRODUCTION

Group signatures, introduced by Chaum and van Heyst [1], provide anonymity for signers. A group member can sign on behalf of the group, no one can identify the signing member except the group manager (GM).

A group signature scheme generally includes the following steps: Setup, Join, Sign, Verify, and Open. Later, a new step, Revocation, is added into it [2]. GM can revoke a dishonest group member with revocation algorithm. The revoked member can't sign again on behalf of the group, but his former signatures are still valid.

There are two main revocation methods in group signature: one is based on witness, the other is based on revocation list (RL). In a membership revocation resolution based on witness [3], GM publishes a single accumulated value a, every group member proves in a zero-knowledge way that he/she knows corresponding witness w to a. It should be hard for users outside the group to forge such witnesses. Revocations in this category are more efficient than RL based resolutions, but they have a common drawback that previously signed signatures might not being able to pass verifying algorithm under the current verification keys. In the category of membership revocation schemes based on RL [4], GM issues a revocation list of identities (public membership keys). Any group member proves in a zero-knowledge way that his/her identity embedded in the signature is not equal to any one in the RL. The corresponding revocation messages are only sent to verifiers, while the signers are not involved. Since the signer's costs are lower, this approach is suitable for mobile environments where mobile hosts anonymously communicate with the servers. This type of group signature is called Verifier-Local Revocation (VLR for short) group signature.

VLR group signature was formalized in [5], which presented a short group signature with VLR based on [6]. Nakanishi et al.[7] pointed out that this scheme did not satisfy the security of backward unlinkability (or BU-anonymity), and proposed another VLR scheme with the feature of backward unlinkability, i.e., group signatures generated by the same group member is unlinkable except himself and GM, even after this member has been revoked (his/her revocation token is published). From then, many VLR group signature schemes with BU-anonymity (BU-VLR group signature for short) were proposed based on different security assumptions [8-15]. Among these schemes, there is a type of VLR group signatures which are based on q-SDH assumption and DTDH (Decisional Tripartite Diffie-Hellman) assumption. Based on these two security assumptions, in 2006, Zhou and Lin proposed the first BU-VLR group signature scheme [13]. To overcome the shortcoming that the computational costs were linear with the length of the revocation list, in [14] Zhang et al improved Zhou's scheme, but the backward unlinkability (BU-anonymity) was not satisfied. Then in [15], Wei et al proposed another scheme which had shorter signature size and lower computation costs.

In this paper, we propose a new VLR group signature scheme from bilinear maps with backward unlinkability and traceablity based on q-SDH assumption and DTDH assumption. Compared with the existing VLR group signature schemes based on these two security assumptions, the proposed scheme has lower computation costs and the shortest signature length.

2. PRELIMINARIES

Definition 1 Bilinear maps:

(1) $G_1 =< g_1 >$, $G_2 =< g_2 >$, G' are multiplicative cyclic groups of prime order p . (2) ψ is an efficiently computed isomorphism from G_2 to G_1, with $\psi(g_2) = g_1$. (3) e is an efficiently computed bilinear map: $G_1 \times G_2 \rightarrow G'$, i.e., for all $u \in G_1$, $v \in G_2$ and $a,b \in Z$, $e(u^a, v^b) = e(u,v)^{ab}$, and $e(g_1, g_2) \neq 1$.

In this paper, we set $G_1 = G_2 = G$, $g_1 = g_2 = g$.

Definition 2 (q -SDH assumption [5,6]) For all PPT algorithms , the probability

Pr [$(g, g^{\gamma}, \ldots, g^{\gamma^q}) = (g^{1/\gamma+x}, x)$]

is negligible, where $x \in Z_p^*$, $\gamma \in Z_p^*$.

Definition 3 (DTDH assumption [13]) For all PPT algorithms, the probability

|Pr[$(g^a, g^b, g^c, g^{abc}) = 0$]-Pr[$(g^a, g^b, g^c, g^d) = 0$]|

is negligible, where $a,b,c,d \in_R Z_p^*$.

We also need the knowledge on signature proof of knowledge (SPK), which can be found in a lot of literatures such as [2] –[8], here, we omit it.

3. MODEL AND DEFINITIONS OF BU-VLR GROUP SIGNATURE

We review the model of BU-VLR group signature in [4,5,8] bellow.

Definition 4 (BU-VLR Group Signature) A BU-VLR group signature scheme consists of the following algorithms.

– **KeyGen**(n,T): A probabilistic algorithm, on input the number of members n and the number of time intervals T , generates a group public key gpk , an n-element vector of members' signing keys $\mathrm{gsk} = (\mathrm{gsk}_1, ..., \mathrm{gsk}_n)$ and revocation token $\mathrm{grt} = (\mathrm{grt}_{11}, ..., \mathrm{grt}_{nT})$, where gsk_i is kept secret by member $\mathrm{i} \in [1,\mathrm{n}]$ and grt_{ij} denotes the revocation token of member $i \in [1,n]$ at time interval $\mathrm{j} \in [1,\mathrm{T}]$.

– **Sign**(gpk, j, gsk_i, M): A probabilistic algorithm generates the signature σ on a message M at the current time interval j by member i using gsk_i and gpk .

– **Verify**(gpk, j, RL_j, σ, M): A deterministic algorithm includes signature check and revocation check, which can be performed by anyone to generate one bit b .If $b = 1$, it means σ is a valid signature on M at interval j by one member of the group whose revocation token is not in RL_j j. If $b = 0$, then σ is invalid.

– **Revoke**(RL_j, grt_{ij}): This algorithm adds grt_{ij} to RL_j if member i is to be revoked at the time interval $j \in [1,T]$.

Sometimes, a group signature need be opened to find the actually singer. An open algorithm can be constructed by using revocation check.

Definition 5 (Correctness) For all (gpk, gsk, grt) $= KeyGen(n,T)$, all $j \in [1,T]$, all $i \in [1,n]$, and all $M \in \{0,1\}^*$, this requires that,

Verify(gpk, j, RL_j, Sign(gpk, j, gsk_i, M), M) =1

$\Leftrightarrow grt_{ij} \notin RL_j$.

Definition 6 (BU-anonymity) BU-anonymity requires that for all PPTA, the advantage of A on the following BU-anonymity game is negligible.

– **Setup**: The challenger runs the key generation algorithm to obtain (gpk, gsk, grt) , and provides the adversary A with gpk .

– **Queries**: The challenger announces the beginning of every interval $j \in [1,T]$ toA, which is incremented with time. A can request the challenger about the following queries at the current interval j.

- **Signing**: A requests a signature of any member i on arbitrary message M at interval j. The corresponding signature is responded by the challenger.

- **Corruption**: A requests the secret key of any member i.

- **Revocation**: A requests the revocation token of any member i at interval j. The challenger responds with grt_{ij}.

– **Challenge**: A outputs some (M, i_0, i_1, j_0) with restriction that i_0 and i_1 have not been corrupted, and their revocation tokens have not been queried before the current interval j_0 (including j_0). The challenger randomly selects $\phi \in \{0,1\}$, and responds with signature of member i_ϕ on M at interval j_0.

– **Restricted queries**: A is allowed to make queries of signing, corruption and revocation, except the corruption queries of i_0, i_1 and their revocation queries at interval j_0. Note that A can query he revocations of i_0 and i_1 at interval j' ($j' \geq j_0$) for the BU property.

– **Output**: A outputs a bit ϕ' as its guess of ϕ.

If $\phi' = \phi$, A wins the game. The advantage of A is defined as $|\mathrm{pr}[\phi' = \phi] - 1/2|$.

Definition 7 (Traceability) Traceability requires that for all PPTA, the advantage of A on the following game is negligible.

– **Setup**: The challenger runs the key generation algorithm to obtain (gpk, gsk, grt), and sets U empty. The adversary A is provided with gpk and grt.

– **Queries**: A can request the challenger about the following queries at each interval $j \in [1,T]$.

- **Signing**: A requests a signature of any member i on arbitrary message M at interval j. The corresponding signature is responded by the challenger.

- **Corruption**: A requests the secret key of any member i. The challenger responds the corresponding key and adds i to U.

– **Output**: A outputs $(M^*, j^*, RL_{j^*}, \sigma^*)$. A wins if (1) Verify $(gpk, M^*, j^*, RL_{j^*}, \sigma^*)$ = 1, and (2) σ^* is traced to a member outside of $U \setminus RL_{j^*}$ or failure, and (3) A has not obtained σ^* in signing queries on message M^*.

4. PROPOSED VLR GROUP SIGNATURE SCHEME

Suppose n is the number of group members, T is the number of time intervals.

KEYGEN (n, T):

(1) GM selects a generator g of G and a collision-resistant hash function $H : \{0,1\}^* \to Z_p^*$.

(2) GM selects $\gamma \in_R Z_p^*$ and computes $\omega = g^\gamma$.

(3) GM selects $x_i \in_R Z_p^*$, computes $A_i = g^{1/(\gamma + x_i)}$, $B_i = g^{1/x_i}$ for all group members $i \in [1, n]$.

(4) GM selects $r_j \in_R Z_p^*$, calculates $h_j = g^{r_j}$ for all $j \in [1,T]$ and $B_{ij} = (\omega g^{x_i})^{r_j} = g^{(\gamma + x_i) r_j}$ for all i and j.

The group public key is $gpk = (g, \omega, h_1, \ldots, h_T)$, the private key of member i is $gsk[i] = (A_i, x_i)$, the revocation token of i at time interval j is $grt[i][j] = (B_i, B_{ij})$.

SIGN (gpk, j, $gsk[i]$, M): Hereafter, we assume that M includes the time interval j in order to bind the signature to the interval. Group member i does the followings:

1 Select random $\alpha \in Z_p^*$, compute $T_1 = A_i^\alpha$, $T_2 = h_j^{x_i \alpha}$.

2 Set $\eta = x_i \alpha$, generate a signature proof of knowledge V:

$$V = SPK\{(\alpha, x_i, A_i) : T_1 = A_i^\alpha, T_2 = h_j^{x_i \alpha}, e(A_i, \omega g^{x_i})$$

$$= e(g,g)\}(M) = SPK\{(\alpha, x_i, \eta) : T_2 = h_j^\eta, e(T_1, \omega)$$

$$= e(g,g)^\alpha \big/ e(T_1, g)^{x_i}\}(M)$$

The group signature on M signed by group member i at time interval j is $\sigma = (T_1, T_2, V)$, where V can be calculated as follows:

Choose $r_\alpha, r_{x_i}, r_\eta \in_R Z_p^*$, and compute $R_1 = h_j^{r_\eta}$, $R_2 = e(g,g)^{r_\alpha} / e(T_1, g)^{r_{x_i}}$, $c = H(gpk, M, T_1, T_2, R_1, R_2)$, and $s_\alpha = r_\alpha + c\alpha$, $s_{x_i} = r_{x_i} + cx_i$, $s_\eta = r_\eta + c\eta$, then $V = (c, s_\alpha, s_{x_i}, s_\eta)$.

REVOKE (RL_j , $grt[i][j]$): If i is revoked at time interval j, then $RL_j \leftarrow RL_j \cup \{(B_i, B_{ij})\}$.

VERIFY (gpk , j , RL_j , σ , M): A verifier can check the validity of σ by:

1 Signature check. Check the validity of V as follows. Given $\sigma = (T_1, T_2, c, s_\alpha, s_{x_i}, s_\eta)$, calculate

$R_1' = h_j^{s_\eta} / T_2^c$, $R_2' = e(g,g)^{s_\alpha} / (e(T_1, g)^{s_{x_i}} e(T_1, \omega)^c)$,

Validate $c = H(gpk, M, T_1, T_2, R_1', R_2')$.

2 Revocation check. Check that the signer is not revoked at the interval j, by checking

$e(T_1, B_{ij}) \neq e(B_i, T_2)$ for all $B_{ij} \in RL_j$.

5. SECURITY

Theorem 1. The proposed VLR group signature scheme satisfies the BU-anonymity in the random oracle model under the DTDH assumption.

The following lemma implies the above theorem.

Lemma 1. Suppose adversary A breaks the BU-anonymity of the proposed scheme with the advantage ε and q_H hash queries and q_S signature queries. Then, we can construct B that breaks the DTDH assumption with the advantage $(1/nT - (q_H q_S)/p)\varepsilon$.

Intuition: in the proof, g^a, g^b, g^c, g^{abc} in the DTDH assumption are regarded as the followings: $a = x_i$, $g^b = h_j$, $c = \alpha$, and $g^{abc} = h_j^{x_i\alpha}$. The DTDH assumption means that $g^{abc} = h_j^{x_i\alpha}$ and a random g^d are indistinguishable, and thus $T_2 = h_j^{x_i\alpha}$ does not reveal any information on private key.

Proof: The input of B is $(g, g_1 = g^a, g_2 = g^b, g_3 = g^c, Z)$, where either $Z = g^{abc}$ or $Z = g^d$, and $a,b,c,d \in_R Z_p^*$. The task of B is to decide that Z it is given is g^{abc} or g^d by communicating with A, as follows.

□**Setup**: B simulates KEYGEN(n , T) as follows:

1B picks $i^* \in_R [1,n]$ and $j^* \in_R [1,T]$, furthermore, B selects $\gamma \in_R Z_p^*$, and computes $\omega = g^\gamma$.

2 B selects $r_j \in_R Z_p^*$, and computes

$$h_j = \{ \begin{matrix} g^{r_j}, j \neq j^* \\ g_2 = g^b, j = j^* \end{matrix} .$$

3 For all $i \in [1,n]$, B□ selects $x_i \in_R Z_p^*$ and computes $A_i = g^{1/(\gamma + x_i)}$, $B_i = g^{1/x_i}$ for all $i \in [1,n]$ except i^*. For i^*, define $x_{i^*} = a$ and $A_{i^*} = g^{1/(\gamma + a)}$, $B_{i*} = g^{1/a}$ which is unknown for B since B does not know a.

4 B computes $B_{ij} = h_j^{(\gamma + x_i)} = g^{(\gamma + x_i) r_j} = (\omega g^{x_i})^{r_j}$ for all $i \in [1,n]$ except i^* and all j. For i^*, B sets $B_{i^* j} = (\omega g_1)^{r_j} = g^{(\gamma + a) r_j} = h_j^{(\gamma + a)}$ except for j^*. For i^* and j^*, define $B_{i^* j^*} = g^{(\gamma + a) b}$, which is also unknown for B since B does not know a, b.

Hash queries: At any time, A can query the hash function used in SPK. B responds with random values with consistency.

Signing queries: A can query the signature of member i at any time interval j. If $i \neq i^*$, B knows (A_i, x_i), so B computes a signature with the algorithm SIGN to respond the query as usual. For $i = i^*$ and $j \neq j^*$, B selects $\alpha \in Z_p^*$, $T_1 \in_R G$, computes $T_2 = g_1^{r_j \alpha} = g^{a r_j \alpha} = h_j^{a\alpha} = h_j^{x_{i^*}\alpha}$. For $i = i^*$ and $j = j^*$, B selects $z \in_R Z_p^*$, $T_1 \in_R G$, then computes $T_2 = g_1^z$. From the view of A, the above choices also satisfy $T_2 = h_{j^*}^{x_{i^*}\alpha}$, where $\alpha = z/b$.

Then, B computes the simulated SPK $V = SPK$ (T_1, T_2) by using the simulator of the perfect zero-knowledgeness, which includes the backpatch of the hash function. If the backpatch is failure, B outputs a random guess $\omega' \in_R \{0,1\}$ and aborts. Otherwise, B responds signature $\sigma = (T_1, T_2, V)$ toA. Note that each value in σ has the same distribution as the real due to $\alpha \in_R Z_p^*$.

Revocation queries: A can query the revocation token of i at time interval j. If $i \neq i^*$, $j \neq j^*$, B responds B_{ij} to A; If $i = i^*$ and $j = j^*$, B outputs a random guess $\omega' \in_R \{0,1\}$ and aborts.

Corruption queries: A can query the secret key of i. If $i \neq i^*$, B responds (A_i, x_i) to A□ If $i = i^*$, B outputs a random guess $\omega' \in_R \{0,1\}$ and aborts.

Challenge: A outputs a message M, the current time interval j and two members i_0, i_1 to be challenged. B picks $\phi \in_R \{0,1\}$. If $i_\phi \neq i^*$ or $j \neq j^*$, B outputs a random guess $\omega' \in_R \{0,1\}$ and aborts; If $i_\phi = i^*$ and $j = j^*$, B responds the following simulated group signature: B regards c as α, sets $T_1 \in_R G$, $T_2 = Z$, then computes the simulated SPK $V = SPK(T_1, T_2)$ by using the simulator of the perfect zero-knowledge-ness. Note that, if $Z = g^{abc}$, then, $T_2 = g^{abc} = {h_{j^*}}^{x_{i^*}c} = {h_{j^*}}^{x_{i^*}\alpha}$, thus (T_1, T_2, V) is a simulated signature with the same distribution as the real signature; If $Z = g^d$, then $T_2 = g^d$, thus A can decide φ only by guessing.

Output: A outputs its guess $\phi' \in \{0,1\}$. If $\phi = \phi'$, B outputs $\omega' = 1$ (implying $Z = g^{abc}$) , and otherwise outputs $\omega' = 0$ (implying $Z = g^d$).

Now, we evaluate the advantage of the guess of B. Let $\omega \in \{0,1\}$ denote whether the input Z is g^d ($\omega = 0$) or g^{abc} ($\omega = 1$). Let *abort* be the event that B aborts. Then, we have Pr[$\omega = \omega' \mid abort$]=$1/2$. On the other hand, assume that B does not abort. If $\omega = 0$, i.e., $Z = g^d$, then the challenged signature has no information on x_{i^*}, A decides the output only by guessing, thus Pr[ω'=0$\mid \overline{abort} \wedge \omega$ =0] = $1/2$. If $\omega = 1$, i.e., $Z = g^{abc}$, then B perfectly simulates the real and thus A guesses correctly with the advantage ε by the assumed condition. Therefore, we obtain Pr[ω' =1$\mid \overline{abort} \wedge \omega$ =1] =$1/2 + \varepsilon$.

Putting everything together, we obtain the advantage of B's guess, as follows.

$\mid$Pr[B $(g, g^a, g^b, g^c, g^{abc}) = 0$] -Pr[B $(g, g^a, g^b, g^c,$ $g^d)$ $= 0$]$\mid = \mid$ Pr[$\omega' = 0 \mid \omega = 1$]- Pr[$\omega' = 0 \mid \omega =$ 0]$\mid = \mid$(1- Pr[$\omega' = 1 \mid \omega = 1$]) - Pr[$\omega' = 0 \mid \omega = 0$] $\mid = \mid$1-Pr[$abort$]Pr[$\omega' = 1 \mid abort \wedge \omega = 1$]- Pr[$\overline{abort}$]Pr[$\omega' = 1 \mid \overline{abort} \wedge \omega = 1$]- Pr[$abort$]Pr [$\omega' = 0 \mid abort \wedge \omega = 0$]- Pr[$\overline{abort}$]Pr[$\omega' = 0 \mid$ $\overline{abort} \wedge \omega = 0$]$\mid = \mid$1- Pr[$abort$]($1/2$ + $1/2$)- Pr[$\overline{abort}$](($1/2 + \varepsilon$)+$1/2$)$\mid$=Pr[$\overline{abort}$] ε

In the rest, we evaluate Pr[$\overline{abort}$]. There are two cases that B aborts: first, B does not correctly guess i^* and j^*; second, the backpatch is failure in the signing query. The probability that a specific signature causes the failure is at most q_H / p if the guesses of i^* and j^* are correct, thus, for q_S signing queries, the probability of B aborts due to the failure of the backbatch is at most $q_S q_H / p$. On the other hand, there are n members i and T time intervals j, so the probability that B correctly guesses i^* and j^* is at least $1/nT$. So, Pr[$\overline{abort}$] $\geq$ $1/nT$ - $q_S q_H / p$. Therefore, the advantage that B guesses ω, i.e., the advantage of B breaks DTDH assumption is at least ($1/nT$ - $q_S q_H / p$) ε . □

Theorem 2. The proposed VLR group signature scheme satisfies the traceability in the random oracle model under the q-SDH assumption.

The following lemma implies the above theorem.

Lemma 2. Suppose adversary A breaks the traceability of the proposed scheme with the advantage ε and q_H hash queries and q_S signature queries. Then, we can construct B that breaks the $(n+1)$ -SDH assumption with the advantage $(\varepsilon/n - 1/p)/16 q_H$.

Proof: Consider the following framework between B and A:

Setup: It is given g , $\omega = g^{\gamma}$ and n pairs (A_i, x_i) to B. For each $i \in [1, n]$, either $s_i = 1$ indicating that an SDH pair (A_i, x_i) is known, or $s_i = 0$ indicating that B knows x_i but doesn't know the corresponding A_i . B builds $gpk = (g, h_1, \dots, h_T, \omega)$, and $grt = ((B_1, B_{11}), \dots, (B_n, B_{nT}))$ 。

Hash queries: At any time, A can query the hash function used in SPK. B responds with random values with consistency.

Signing queries : A queries a signature on message M at member i and interval j . If $s_i = 1$, since knows the secret key (A_i, x_i) , B responds a signature using the SIGN algorithm. If $s_i = 0$, B picks $\alpha \in Z_p^*$, $T_1 \in_R G$, and computes $T_2 = h_j^{x_i \alpha}$. Then, B computes the simulated SPK $V = SPK(T_1,$ $T_2)$ by using the simulator of the perfect zero-knowledge-ness, which includes the backpatch of the hash function. B responds $\sigma = (T_1, T_2, V)$ to A。

Corruption queries: A requests the secret key at member i . If $s_i = 0$, then abort. If $s_i = 1$, B responds requested key (A_i, x_i) .

Output: Finally, A outputs a forged signature $\sigma^* = (T_1^*, T_2^*, V^*)$ including a secret key A^* . Using the implying open algorithm by all B_{ij} in the sign algorithm, B can identify the member. If the identification fails (i.e., the member is outside of all i), output σ^* . Otherwise, some i is identified. If $s_i = 0$, output σ^* . If $s_i = 1$, abort.

Then, there are two types of forger on the above framework. Type 1 forger forges a signature of the member who is different from all i . Type 2 forger forges a signature of the member i whose corruption is not requested.

In the following, as in [3], for a q -SDH instance $(g, g^{\gamma}, \dots, g^{\gamma^q})$, we can obtain g , $\omega = g^{\gamma}$ and q -1 SDH pairs (A_i, x_i) . On the other hand, any SDH pair besides these q -1 SDH pairs can be transformed to a solution of the q -SDH instance, which means that the q -SDH assumption is broken. Now, we treat two types of forger differently.

Type 1. Given (n+1)-SDH instance, as above, Bcan obtain n SDH pairs (A_i, x_i) with (g , ω).Then, perform the framework with type 1 forger A. Now, all $s_i = 1$, A finally outputs a signature with secret key $A^* \neq A_i$ for all i . By the assumption in the proposition, A succeeds with advantage ε .

Type 2. Given n-SDH instance, as above, B can obtain n-1 SDH pairs (A_i, x_i) with (g , ω).B picks random index i^* and $x_{i^*} \in_R Z_p^*$ (A_{i^*} is unknown). Now, B has n SDH pairs, and $s_{i^*} = 0$,while $s_i = 1$ for other i . Then, perform the framework with type 2 forger A. In this case, it succeeds only if A never requests the corruption of i^* , but forges the signature including A_{i^*} .There are n pairs, the probability of a forged signature traced to i^* is $1/n$, and A breaks the traceability of the proposed scheme with the advantage ε by proposition, thus, A succeeds with advantage ε/n in this type forging.

Now we show how to obtain another SDH pairs beyond the given q -1 SDH pairs, using the framework with type 1 or type 2. As shown in [3], the successful probability is at least $(\varepsilon' - 1/p)^2 / 16q_H$, where ε' is the probability that the framework on each forger succeeds.

In all, we have the following conclusion. Using type 1 forger, we can solve the (n+1)-SDH instance with probability $(\varepsilon - 1/p)^2 / 16q_H$. Using type 2 forger, we can solve the n-SDH instance with probability $(\varepsilon/n - 1/p)^2 / 16q_H$ .We can suppose the type of forger with the probability $1/2$, so the less probability of the type 2 forger proves the lemma.□

6. PERFORMANCE AND COMPARISON

We compare the efficiency of the proposed scheme to the previous VLR schemes based on DTDH assumption. The comparisons are shown in table 1.

Size of signature: the proposed signature $\sigma = (T_1, T_2, c, s_{\alpha}, s_{x_i}, s_{\eta})$ includes 2 elements from G , 4 elements from Z_p . As in [13]-[15], p is170 bits, elements of G are 171bits, thus the size of proposed signature is 1022 bits.

Computations: In our signing algorithm, $R_2 = e(g, g)^{r_{\alpha}} / e(T_1, g)^{r_{x_i}}$. This can be computed as $R_2 = e(g,g)^{r_\alpha} / e(A_i, g)^{\alpha r_{x_i}}$. Thus, all bilinear map computations ($e(g,g)$ and $e(A_i, g)$) can be pre-computed. So, the signature generation requires 4 multi-exponentiations (denoted by ME). In the verification, $R_2' = e(g,g)^{s_\alpha} / (e(T_1, g)^{s_{x_i}} e(T_1, \omega)^c)$. This can be computed in the following way: $R_2' = e(g,g)^{s_\alpha} / e(T_1, g^{s_{x_i}} \omega^c)$, the bilinear map computation $e(g,g)$ can be pre-computed. So, the verification requires 2 multi-exponentiations (denoted by ME) and (1+2|RLj|) bilinear maps (denoted by BM).

The following table is a performance comparison of the existing VLR group signature schemes based on DTDH assumption and ours.

Table 1 Comparisons Among VLR Group Signature Schemes Based On DTDH Assumption (ME Denotes Multi-Exponentiations, BM Denotes Bilinear Map)

Scheme	Size of Signature (bits)	Costs of Signing	Costs of Verification
Zhou06 [13]	2215	11ME +2BM	7ME+ (3+|RLj||)BM
Zhang08 [14]	1533	6ME+1BM	3ME+1BM
Wei 08 [15]	1363	8ME+1BM	5ME+ (3+|RLj|)BM
Ours	1022	4ME	2ME+ (1+2|RLj|)BM

From table 1, we can see that the size of signature of our scheme is the shortest, it reduces about 54% than that of Zhou06 [13] scheme, and reduces about 33% and 25% than that of Zhang08 [14] and Wei08 [15] two schemes respectively. Also, our scheme has the lowest computation costs in the signing phase. Although it has more |RLj| bilinear maps in verifying phase than other schemes, this load is transferred to the verifiers. Because the signer does not involve in the verification, the signer's load is very light in our scheme. So it is suitable for mobile environments.

7. CONCLUSION

In this paper, we propose a new VLR group signature scheme with backward unlinkability based on q-SDH assumption and DTDH assumption. The proposed scheme has the lowest computation costs in the signing phase and achieves the shortest signature length among all existing VLR group signature schemes based on DTDH assumption, and can be applicable to mobile environments such as IEEE 802.1x [16][17].

ACKNOWLEDGEMENT

This work was supported by the National Natural Science Foundation of China (Grant No. 61103199) and China Scholarship Council.

REFRENCES:

[1] Chaum D., Van Heyst E., "Group Signatures", in *Advances in Cryptology- EUROCRYPT'91*, LNCS 547, Berlin: Springer-Verlag, 1991, pp. 257-265.

[2] Bresson E., Stern J., "Efficient Revocation in Group Signatures", in *Public Key Cryptography-PKC 2001*, LNCS 992, Berlin: Springer-Verlag, 2001, pp. 190-206.

[3] Lan Nguyen, "Accumulators from Bilinear Pairings and Applications", in: *Alfred Menezes (Ed.), CT-RSA'05*, LNCS 3376, Berlin, 2005, pp. 275-292.

[4] G. Ateniese, D. Song, and G. Tsudik, "Quasi-Efficient Revocation in Group Signatures", In: *Rebecca N. Wright (Ed.), Financial Cryptography'02*, LNCS 2357, Springer-Verlag, Berlin, 2002, pp. 183-197.

[5] Boneh D., Shacham H., "Group Signatures with Verifier-Local Revocation", In *Proceedings of the 11th ACM conference on Computer and communications security-CCS'04*, New York: ACM Press, 2004, pp.168-177.

[6] Dan Boneh, Xavier Boyen, and Hovav Shacham, "Short Group Signatures", In: *M. Franklin (Eds.), CRYPTO'04*, LNCS 3152, Springer-Verlag, Berlin, 2004, pp. 45-55.

[7] Nakanishi T., Funabiki N., "Verifier-Local Revocation Group Signature Schemes with Backward Unlinkability from Bilinear Maps", in *Advances in Cryptology-ASIACRYPT 2005*, Berlin: Springer-Verlag, 2005, 533-548.

[8] Nakanishi T., Funabiki N., "A Short Verifier-Local Revocation Group Signature Scheme with Backward Unlinkability", *IEICE Transactions on Fundamentals of Electronics*, Vol. E90-A, No.9, 2007, PP. 1793-1802.

[9] Zhang J., Ma L., Sun R. et al., "More Efficient VLR Group Signature Satisfying Exculpability", *IEICE Transactions on Fundamentals of Electronics*, Vol. E91-A, No.7, 2008, pp. 1831-1835.

[10] Wei L., Wu C. Zhou S., "Efficient Verifier-Local Revocation Group Signature Schemes

with Backward Unlinkability", *Chinese Journal of Electronics*, Vol.18, No.2, 2009, pp. 379-384.

[11] Wei L., Liu J., "Shorter Verifier-Local Revocation Group Signature with Backward Unlinkability", in *Pairing 2010*, LNCS 6487, Berlin: Springer-Verlag, 2010, pp. 136-146.

[12] Li J., Sun G., Zhang Y., "Practical Group Signature with Verifier-Local Revocation", *Journal on Communications*, Vol. 32, No.10, 2011, pp. 67-77.

[13] Zhou S., Lin D., "A Short Group Signature with Verifier-Local Revocation and Backward Unlinkability", *Cryptology ePrint Archive: Report 2006/100*, 2006.

[14] Zhang J., Li Y., Wang Y., "Shorter Group Signature Scheme with Verifier-Local Revocation", *Journal of Xi'an Jiaotong University*, Vol. 42, No.10, 2008, pp. 1250-1253.

[15] Wei L., Wu C. Zhou S., "A New Verifier-Local Revocation Group Signature with Backward Unlinkability", *Journal of Computer Research and Development*, Vol.45, No.8, 2008, pp. 1315-1321.

[16] Dinakaran M., Balasubramanie P., "Network Mobility (NEMO) Security: Threats and Solutions", *Journal of Theoretical and Applied Information Technology*, Vol. 35, No. 1, January 2012, pp. 77-82.

[17] K. Govinda, E. Sathiyamoorthy, "Secure Key Sharing for Group Communication under Community Cloud", *Journal of Theoretical and Applied Information Technology*, Vol. 48, No. 2, December 2013, pp. 674-679.