Breaking and Building of Group Inside Signature

S. Sree Vivek*, S. Sharmila Deva Selvi, S. Gopi Nath, and C. Pandu Rangan*

Indian Institute of Technology Madras,
Theoretical Computer Science Laboratory,
Department of Computer Science and Engineering,
Chennai, India
{svivek,sharmila,gopinath,prangan}@cse.iitm.ac.in

Abstract. Group Inside Signature (GIS) is a signature scheme that allows the signer to designate his signature to be verified by a group of people. Members other than the designated group cannot verify the signature generated by the signer. In Broadcast Group Oriented Signature (BGOS), a user from one group can designate his signature to be verified by members of another group. An Adaptable Designated Group Signature (ADGS), is one in which an user can designate his signature to be verified by a selected set of members who are from different groups. The two GIS schemes [5], [6] and the BGOS scheme [7], we consider are certificateless schemes and the ADGS scheme [8] which we consider here is an identity based scheme. In this paper, we present the cryptanalysis of all the four schemes that appeared in [5], [6], [7] and [8]. We also present a new identity based ADGS (N-ADGS) scheme and prove its security in the random oracle model. The existing model described in [8] for ADGS did not consider unlinkability which is one of the key properties required for ADGS. We provide the security model for unlinkability and also prove our scheme is unlinkable.

Keywords: Cryptanalysis, Group Inside Signature, Broadcast Group Oriented Signature, Adaptable Designated Group Signature, Provable Security, Random Oracle model.

1 Introduction

In general, digital signatures are publicly verifiable. Jackbson et.al (1996) [4] proposed the concept of Designated Verifier Signatures (DVS) and strong DVS (SDVS). In DVS, only a designated person can verify the signature, which is signed by a signer. DVS achieves this property by providing an ability called *Simulatability* to the designated verifier, which allows him to simulate the actual signers signature. In SDVS, any third party cannot verify the validity of the signature unless the private key of the designated verifier or the actual signer is exposed.

* Work supported by Project No. CSE/05-06/076/DITX/CPAN on Protocols for Secure Communication and Computation sponsored by Department of Information Technology, Government of India.

Y. Chen et al. (Eds.): SecureComm 2009, LNICST 19, pp. 330–339, 2009.

Extending a single party verification scheme to a designated group verification scheme is a challenging problem. In practice, there may be different group models. First, in networks like Local Area Networks, all group members reside in a single network and no member of the group may hang outside network. Certificateless GIS schemes [5] and [6] provide solutions for designating a signature to be verified inside such a group. Secondly, in distributed networks, the users of different companies or institutions naturally come under different work groups. If a member of one group wants to send a signed document to members of another group, BGOS [7] can be used. Moreover the signer wants to prevent the members outside the designated group from verifying the signature. The scheme in [7] focuses on this problem. Finally, in distributed networks, a signer may want several members to verify his signature, no matter whether those members are in same or different groups. The signer wants to prevent the members outside the defined group from verifying the signature. This model can be visualized as a more generalized version of the previous two models. ADGS scheme in [8] focuses on this problem. In fact even if a designated verifier v_i belongs to a group say G, while v_i can verify the signature of the sender, other members of G cannot verify the signature.

Suppose that a organization initiates a call for tender, asking for quotations to some companies for a set of instruments and tasks to be accomplished. Here, the requirement is that, the competing companies should not be able to verify the quotations quoted by their counter parts. So each company will encrypt and sign the quotation and send it to the organization. But nothing prevents the organization from revealing the quoted values once decrypted, since the organizations goal is to obtain quotations with low price. In this situation the organization could show the signed offers to some other companies and influence them to make better quotations. Here, we can use the ADGS scheme, because the company which proposes the quotation can designate the signature to the organization who has called for the tender and other companies can not verify the validity unless the verifier uses the private key of the organization.

Simulatability vs Unlinkability. The notion "Simulatability" in the context of DVS ensures that the designated verifier has the ability to simulate the transcript as if it is generated by the actual signer i.e., we can say that the designated verifier is also capable of generating the signature of the signer. Where as the notion of "Unlinkability" in the context of ADGS ensures that only the designated group members can verify the signature designated to them, members other than the designated group can not verify the signature. Thus, Unlinkability is different from Simulatability and should not be confused with each other.

Our Contribution. In this paper, we show that GIS in [5] and BGOS in [7] are not secure against both Type-I and Type-II adversaries, and the GIS in [6] is not secure against Type-I adversary. We also show that the basic ADGS scheme [8] is universally forgeable. We also propose a new Adaptable Designated Group Signature scheme (New-ADGS) and prove its security formally in the random oracle model. Due to page limitation, we omit the reviews of the broken schemes and the security proofs of the newly proposed ADGS scheme and is given in the full version of this paper [10].

2 Preliminaries

2.1 Bilinear Pairing

Let $\mathbb{G}_1$ be an additive cyclic group generated by P, with prime order q, and $\mathbb{G}_2$ be a multiplicative cyclic group of the same order q. A bilinear pairing is a map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ with the following properties.

- ***Bilinearity.*** For all $P, Q, R \in_R \mathbb{G}_1$ and $a, b \in_R \mathbb{Z}_q^*$, $\hat{e}(P + Q, R) = \hat{e}(P, R)$ $\hat{e}(Q, R)$, $\hat{e}(P, Q + R) = \hat{e}(P, Q)\hat{e}(P, R)$ and $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$
- ***Non-Degeneracy.*** There exist $P, Q \in \mathbb{G}_1$ such that $\hat{e}(P, Q) \neq I_{\mathbb{G}_2}$, where $I_{\mathbb{G}_2}$ is the identity element of $\mathbb{G}_2$.
- ***Computability.*** There exists an efficient algorithm to compute $\hat{e}(P, Q)$ for all $P, Q \in \mathbb{G}_1$.

3 Cryptanalysis of Certificateless GIS and BGOS Schemes

In this section we show the weaknesses in two certificateless GIS schemes [5], [6] and a certificateless BGOS scheme [7].

3.1 Cryptanalysis of Certificateless GIS Scheme [5]

GIS scheme given in [5] allows the signer to designate his signature to be verified by a group of people who belong to the signer's group. Members other than the designated group should not be able to verify the signature generated by him. The scheme in [5] is not secure against Type-I and Type-II attacks.

Type-I Attack. On seeing a valid signature by an user on some message, anyone can commit a forgery on any message. During the unforgeability game between the challenger $\mathcal{C}$ and adversary $\mathcal{A}_I$, $\mathcal{C}$ gives $\mathcal{A}_I$ the public parameters *params* and $\mathcal{A}_I$ gives to $\mathcal{C}$ a target identity ID^*. $\mathcal{A}_I$ is supposed to generate a valid forgery for the target identity ID^* on some message and $\mathcal{A}_I$ is not allowed to query partial private key for the target identity ID^*. $\mathcal{A}_I$ interacts with $\mathcal{C}$ and access all the oracles with the restrictions given in the model. $\mathcal{A}_I$ can query signature on any message and user identity pair $\langle m, ID \rangle$. $\mathcal{A}_I$ can replace the public keys of suppose any user including user with identity ID^*. During the training-phase $\mathcal{A}_I$ receives a valid signature $\sigma = \langle m, U, V \rangle$ on a message m with target identity ID^* using the **Sign** oracle. Now we show how $\mathcal{A}_I$ can generate a valid signature σ^* on an arbitrary message m^* for the target identity ID^*, such that σ^* is not the output of previous queries to **Sign** oracle. This can be shown by the following computation done by $\mathcal{A}_I$

- Computes $U^* = U + hP_{i1}$ - $H_0(ID^*)$, where h= $H_1(m, U)$ computed from σ.
- Computes $h^* = H_1(m^*, U^*)$
- Replaces public keys of ID^* as $P^*_{i1} = \frac{1}{h^*}\ H_0(ID^*)$ and $P^*_{i2} = \frac{1}{h^*}\ P$.
- $V^* = V$.

Now we claim that $\sigma^* = \langle m^*, U^*, V^*\rangle$ is a valid signature on the message m^* by the user with identity ID^* (with respect to its newly replaced public key). $\mathcal{C}$ can check the validity of the forged signature σ^* as follows.

Correctness of public keys. It is clear that $\langle P^*_{i1},\ P^*_{i2}\rangle$ satisfies the verification $e(P^*_{i1}, P) \overset{?}{=} e(P^*_{i2}, H_0(ID_i))$.

Correctness of forged signature. Note that $\mathcal{C}$ will use the current public key of ID^* that was set by $\mathcal{A}_I$.

- $\mathcal{C}$ has to check whether $e(V^*, P_{j1}) \overset{?}{=} e(U^*, D_{j1})\ e(h^* P^*_{i1}, D_{j1})$. In fact

$$\begin{aligned} R.H.S &= e(U^*, D_{j1})\ e(h^* P^*_{i1}, D_{j1}) \\ &= e(U + hP_{i1} - H_0(ID^*), D_{j1})\ e(h^* P^*_{i1}, D_{j1}) \\ &= e(U + hP_{i1} - H_0(ID^*), D_{j1})\ e(H_0(ID^*), D_{j1}) \\ &= e(U, D_{j1})\ e(hP_{i1}, D_{j1}). \\ &= e(V, P_{j1}). \\ &= e(V^*, P_{j1}) \\ &= L.H.S \end{aligned}$$

Thus the forged signature σ^* passes the verification successfully.

Type-II Attack. Type-II attack is also possible on the same scheme. During the unforgeability game between the challenger $\mathcal{C}$ and adversary $\mathcal{A}_{II}$, $\mathcal{A}_{II}$ can interacts with $\mathcal{C}$ and access the **Sign** oracle with the restrictions given in the model. $\mathcal{A}_{II}$ can ask signature on any message and identity pair $\langle m, ID\rangle$. $\mathcal{A}_{II}$ has access to the master private key. So it can compute the private key of any user from its public keys $\langle P_{i1}P_{i2}\rangle$ as $D_i = kP_{i1}$. Since the public key P_{i1}=$x_i H_0(ID_i)$, so $\mathcal{A}_{II}$ can generate signature on behalf of any user and $\mathcal{A}_{II}$ can verify the signature of any user. Here, we can visualize $\mathcal{A}_{II}$ as the KGC because it knows the master private key in the scheme.

3.2 Cryptanalysis of Another Certificateless GIS Scheme [6]

Chunbo Ma et al. have proposed another GIS [6] scheme. In this section, we present *Type-I* forgery on the scheme [6]. Here adversary $\mathcal{A}_I$ who considered to be inside the group can sign on behalf of any user on any message. During the unforgeability game between the challenger $\mathcal{C}$ and adversary $\mathcal{A}_I$, $\mathcal{C}$ gives $\mathcal{A}_I$ the public parameters *params* and a target identity ID_A. $\mathcal{A}_I$ is supposed to generate a valid forgery for the target identity ID_A on some message and it is not allowed to query partial private key for the target identity ID_A. $\mathcal{A}_I$ interacts with $\mathcal{C}$ and access all the oracles with the restrictions given in the model. $\mathcal{A}_I$

can query signature on any message and user identity pair $\langle m, ID\rangle$. $\mathcal{A}_I$ can replace the public keys of any user including user with identity ID_A. During the training-phase $\mathcal{A}_I$ receives a valid signature $\sigma = \langle m, U, V\rangle$ on a message m with target identity ID_A as the signer from the **Sign** oracle and also obtains the private key of some other user say ID_B from the **Key Extract** oracle. Now $\mathcal{A}_I$ can generate a valid signature σ^* on a message m^* for the target identity ID_A by using the private key of ID_B, such that σ^* is not the output of previous queries to **Sign** oracle. This can be shown by the following computation done by $\mathcal{A}_I$.First $\mathcal{A}_I$ computes the value $e(g, g^k)$ even though $\mathcal{A}_I$ may not know the value $e(g, g^k)$ directly, it can compute $e(g, g^k)$ as follows.

$$\begin{aligned} e(D_B, P_{pub2})e(D_B, (P_{pub1})^{H_1(ID_B)}) &= e(g^{\frac{k^2}{k+H_1(ID_B)}}, g)e(g^{\frac{kH_1(ID_B)}{k+H_1(ID_B)}}, g) \\ &= e(g^{\frac{k^2}{k+H_1(ID_B)}} g^{\frac{kH_1(ID_B)}{k+H_1(ID_B)}}, g) \\ &= e(g, g^k) \end{aligned}$$

Hence, $e(g, g^k)$ can be computed by $\mathcal{A}_I$ and subsequently $\mathcal{A}_I$ generates the forgery by performing the following:

- Computes $r^* = e(g, g^k)^{a^*}$.
- Computes $V^* = H_0(m^*||r^*)$.
- Computes $U^* = SK_B^{(a^*+v^*)}$.
- Replaces ID_A's public keys $X_A^* = X_A$ and $Y_A^* = X_A^{(-H_1(ID_A))}\ X_B^{H_1(ID_B)}\ Y_B$.
- Broadcasts the signature σ^* (m^*, U^*, V^*, ID_A).

Now challenger $\mathcal{C}$ can verify the validity of the signature using the private key of any group member say C as follows:

Computes $r^{'}$ as

$$\begin{aligned} & e(U^*,(X_A^*)^{H_1(ID_A)}.Y_A^*)e(S_C, X_C^{H_1(ID_C)}Y_C)^{-V^*} = \\ & \quad = e(U^*, X_A^{H_1(ID_A)}.X_A^{-H_1(ID_A)}X_B^{H_1(ID_B)}Y_B)\ e(S_C, X_C^{H_1(ID_C)}.Y_C)^{-V^*} \\ & \quad = e(g,g)^{k(a^*+V^*)}e(g,g)^{-V^*k}. \\ & \quad = e(g,g)^{ka^*} \\ & \quad = r' \end{aligned}$$

Checks $V^* \stackrel{?}{=} H_0(m^*||r^{'})$ if it holds σ^* is a valid forgery other wise not.

Since σ^* is a valid forgery which we showed now, we can claim that the scheme given in [6] is having *Type-I* forgery.

3.3 Cryptanalysis of Broadcast Group Oriented Signature [7]

In BGOS, an user from one group can designate its signature to be verifiable by members of other group. In this section we present the cryptanalysis of BGOS scheme, which too has both Type-I and Type-II attacks.

Type-I Attack on BGOS Scheme [7]. On seeing a valid signature by an user on some message, anyone can commit a forgery on any message. During the unforgeability game between the challenger $\mathcal{C}$ and adversary $\mathcal{A}_I$, $\mathcal{C}$ gives $\mathcal{A}_I$ the public parameters *params* and $\mathcal{A}_I$ gives to $\mathcal{C}$ a target identity ID^*_{bi}. $\mathcal{A}_I$ is supposed to generate a valid forgery for the target identity ID^*_{bi} on some message and it is not allowed to query partial private key for target identity ID^*_{bi}. $\mathcal{A}_I$ interacts with $\mathcal{C}$ and access all the oracles with the restrictions given in the model. $\mathcal{A}_I$ can query signature on any message and user identity pair $\langle m, ID \rangle$. $\mathcal{A}_I$ can replace the public keys of suppose any user including user with identity ID^*_{bi}. During the training-phase $\mathcal{A}_I$ receive a valid signature $\sigma = \langle m, U_1, U_2, V \rangle$ on a message m with target identity ID^*_{bi} using the **Sign** oracle. Now we show how $\mathcal{A}_I$ can generate a valid signature σ^* on an arbitrary message m^* for the target identity ID^*_{bi}, such that σ^* is not the output of previous queries to **Sign** oracle. This can be shown by the following computation done by $\mathcal{A}_I$

- Computes $U_1^* = U_1 + hP_{bi}$-$H_0(ID^*_{bi})$ and $U_2^* = U_2 + hP_A^{(2)}$ - P.
- Computes $h^* = H_1(m^*, U_1^*)$.
- Replaces ID^*_{bi}'s public keys as $P^*_{bi} = \frac{1}{h^*} H_0(ID^*_{bi})$ and $Q^*_{bi} = \frac{1}{h^*} P$.
- Replaces group $\mathcal{A}$'s public keys as $P_A^{(2)*} = \frac{1}{h^*} P$ and $Q_A^{(2)*} = \frac{1}{h^*} H_0(ID_A)$.
- $V^* = V$.

Now we claim that $\sigma^* = \langle m^*, U_1^*, U_2^*, V^* \rangle$ is a valid signature on the message m^* by the user with identity ID^*. $\mathcal{C}$ can check the validity of the forged signature σ^* as follows.

Correctness of Public Keys: The replaced public keys of group $\mathbb{A}$ $\langle P_A^{(2)*}, Q_A^{(2)*} \rangle$ passes the verification

$$e(P_A^{(2)^*}, H_0(ID_A)) \stackrel{?}{=} e(P, Q_A^{(2)^*})$$

The replaced public keys of user b_i $\langle P^*_{bi}, Q^*_{bi} \rangle$ also passes the following verification:

$$e(P^*_{bi}, P) \stackrel{?}{=} e(Q^*_{bi}, H_0(ID_{bi}))$$

Correctness of forged signature: Note that $\mathcal{C}$ will use the current public key of ID^* that was set by $\mathcal{A}_I$. $\mathbb{C}$ has to check $e(V^*, P_{ai}) \stackrel{?}{=} e(h^* P^*_{bi} + U_1^*, D_{aj}^{(2)})$ $e(h^* P_A^{(2)*} + U_2^*, D_{aj}^{(1)})$. Now,

$$
\begin{aligned}
&R\ .H.S = \\
&\quad = e(h^* P^*_{bi} + U_1^*, D_{aj}^{(2)}) e(h^* P_A^{(2)*} + U_2^*, D_{aj}^{(1)}) \\
&\quad = e(h^* P^*_{bi} + U_1 + hP_{bi} - H_0(ID^*_{bi}), D_{aj}^{(2)}) e(h^* P_A^{2*} + U_2 + hP_A^{(2)} - P, D_{aj}^{(1)}) \\
&\quad = e(hP_{bi} + U_1, D_{aj}^{(2)}) e(hP_A^{(2)} + U_2, D_{aj}^{(1)}) \\
&\quad = e(V, P_{ai}) \\
&\quad = e(V^*, P_{ai}) \\
&\quad = L.H.S
\end{aligned}
$$

Thus the forged signature σ^* passes the verification successfully.

Type-II Attack on BGOS Scheme [7]. Type-II attack is also possible on BGOS [7] scheme. During the Unforgeability game between the challenger $\mathcal{C}$ and adversary $\mathcal{A}_{II}$, $\mathcal{A}_{II}$ can interact with $\mathcal{C}$ and can access **Sign** oracle with the restrictions given in the model. $\mathcal{A}_{II}$ can ask signature on any message and identity pair $\langle m, ID \rangle$. The adversary $\mathcal{A}_{II}$ can access the master private key. So, $\mathcal{A}_{II}$ can compute the full private key of any user from group $\mathbb{A}$ using the public keys $\langle P_{ai}, Q_{ai} \rangle$ as $\langle \{D_{ai}^{(1)}, D_{ai}^{(2)}\} \rangle = \langle sP_{ai}, tP_{ai} \rangle$ and any user from group B with public keys $\langle P_{bi}, Q_{bi} \rangle$ as $\langle \{D_{bi}^{(1)}, D_{bi}^{(2)}\} \rangle = \langle sP_{bi}, tP_{bi} \rangle$. As a result the KGC can generate signature on behalf of any user and also verify the signature of any user in any group, which contradicts the statement of the authors.

4 Cryptanalysis of Identity Based ADGS Scheme [8]

In this section, we present the cryptanalysis of an identity based ADGS scheme [8]. We show that the ADGS scheme in [8], is universally forgeable by demonstrating two different ways to proceed with the attack.

Universal Forgery Without Having Access to Any Previous Signature. The scheme ADGS described above is universally forgeable. The adversary $\mathcal{A}$ can forge the signature of any user without seeing any valid signature previously signed by any user. $\mathcal{A}$ selects r^*, k^*, $t^* \in_R \mathbb{Z}_q^*$, computes $T_i^* = k^* Q_i \ for (i = 1$ *to* n $a_i \in \mathbb{U}$). and then computes the following values.

- $V_0^* = t^* s^* P$.
- $V_1^* = t^* k^* P$.
- $V_2^* = r^* k^* P$.
- $h^* = H_1(m^*)$.
- $T_0^* = \frac{1}{h^*} k^* P$.
- $V^* = r^* P + P_{pub}$.

  $\mathcal{A}$ produces $\sigma^* = (m^*, V^*, V_0^*, V_1^*, V_2^*, T_0^*, ..., T_n^*)$ as a valid signature on message m^*.

Now the correctness of the forged signature σ^* can be shown as follows:

Correctness: The $L.H.S$ is

$$\begin{aligned} e(V^*, T_i^*) &= e(r^* P + P_{pub}, k^* Q_i) \\ &= e(r^* P, k^* Q_i) e(P_{pub}, k^* Q_i) \\ &= e(r^* k^* P, Q_i) e(k^* P, sQ_i) \\ &= e(V_2^*, Q_i) e(\tfrac{1}{h^*} T_0^*, D_i) \\ &= R.H.S \end{aligned}$$

Thus, we show that $\mathcal{A}$ is capable of generating a valid ADGS on behalf of user with out knowing users secret key.

Universal Forgery on Seeing a Signature of an User. On seeing a valid signature by an user on some message, anyone can commit a forgery on any message. During the unforgeability game between the challenger $\mathcal{C}$ and adversary $\mathcal{A}$, $\mathcal{C}$ gives $\mathcal{A}$ the public parameters *params* and a target identity ID^*. $\mathcal{A}$ is supposed to generate a valid forgery for the target identity ID^* on some message and it is restricted to query private key for the target identity ID^*. $\mathcal{A}$ interacts with $\mathcal{C}$ and accesses all the oracles with the restrictions given in the model. $\mathcal{A}$ can query signature on any message and user identity pair $\langle m, ID \rangle$. $\mathcal{A}$ can replace the public keys of any user including user with identity ID^*. During the training-phase on receiving a valid signature $\sigma = \langle m, V, V_0, V_1, V_2, T_0, ..., T_n \rangle$ on a message m with target identity ID^* from the **Sign** oracle, $\mathcal{A}$ can generate a valid signature σ^* on a message m^* for the target identity ID^*, such that σ^* is not the output of previous queries to **Sign** oracle. This can be shown by the following computation done by $\mathcal{A}$

- Dividing V by h. $\frac{1}{h}V = (\frac{r}{h} + 1)D_0$ where $h = H_1(m)$.
- Computes $h^* = H_1(m^*)$.
- $V_0^* = V_0$ and $V_1^* = V_1$.
- $V_2^* = \frac{h^*}{h}V_2$.
- The remaining parameters $T_0, ..., T_n$, V_0 and V_1 are same as that of original signature.
- $V^* = h^*\frac{V}{h}$.

Now $\sigma^* = \sigma^*\ (m^*, V^*, V_0, V_1, V_2^*, T_0, ..., T_n)$ is a valid signature on the message by the user with identity ID^*. $\mathcal{C}$ can check the validity of the forged signature σ^* as follows.

Correctness: The $L.H.S$ is

$$\begin{aligned} e(V^*, T_i) &= e((\frac{h^*}{h}r + h^*)D_0, k^*Q_i) \\ &= e(\frac{h^*}{h}rD_0, kQ_i)e(h^*D_0, kQ_i) \\ &= e(\frac{h^*}{h}rkD_0, Q_i)e(h^*kQ_0, D_i) \\ &= e(V_2^*, Q_i)e(\frac{1}{h^*}T_0^*, D_i) \\ &= R.H.S \end{aligned}$$

Now, it is clear that the forged signature σ^* passes the verification successfully.

5 New ADGS Scheme(N-ADGS)

In this section we present a new identity based ADGS scheme. Assume that a signer a_0 has to designate his signature to be verified by n users namely $\{a_1, ..., a_n\}$. All the n users may be from different groups and are selected by a_0. The signer a_0 forms the set $\mathbb{U} = \{a_1, ..., a_n\}$ to generate the signature. In our scheme designated members of the group cannot simulate the signers signature.

- **N-ADGS Initialize:**
  The PKG initializes the system by executing this algorithm. This algorithm takes the security parameter 1^k as input and produces two groups $\mathbb{G}_1$ and

$\mathbb{G}_2$ of prime order q, where $|q| = k$, a generator P of $\mathbb{G}_1$, a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ and two cryptographic hash functions H_1 :$\{0,1\}^* \times \mathbb{G}_2 \times \mathbb{G}_1 \times \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{Z}_q^*$ and H_2 :$\{0,1\}^* \rightarrow \mathbb{G}_1$. The master private key is $s \in_R \mathbb{Z}_q^*$ and the master public key is set to be $P_{pub} = sP$. Sets $\theta = e(P_{pub}, R)$ where $R \in_R \mathbb{G}_1$. The public parameters are $\langle \mathbb{G}_1, \mathbb{G}_2, e, P, P_{pub}, P_{pub}, H_1, H_2, \theta, R \rangle$.

– **N-ADGS Key Generation/Extract:** This algorithm is executed by the PKG and on input of identity ID_i, PKG computes $Q_i = H_2(ID_i)$ and sets the private key as $D_i = sQ_i$. Now, D_i is sent to the user in a secure way.
– **N-ADGS Sign:** To sign a message m for a designated group of users $\mathbb{U} = (a_1, ..., a_n)$ with identities $(ID_1, ..., ID_n)$ the user with identity ID_0, private key D_0 and public key Q_0 performs the following steps:
  - Chooses r,k,t $\in_R Z_q^*$ and computes $T_i = \langle T_{i1}, T_{i2} \rangle$ as $\langle t(Q_i + R), kQ_i \rangle$ for$(i = 1$ to $n)$.
  - Computes $U_1 = rQ_0$, $U_2 = rkP$ and $U_3 = tP$.
  - Computes $\omega = e(D_0, U_3)$ and Computes $W = \theta^t \omega$.
  - Computes $h = H_1(m, \omega, U_1, U_2, U_3)$ and $V = rP_{pub} + hD_0$.

  Now $\sigma = (m, V, W, U_1, U_2, U_3, T_1, ..., T_n, \mathbb{U})$ is a valid signature on message m by ID_0, with the user group $\mathbb{U}$ as designated verifiers.
– **N-ADGS Verify:** Verification is a two step process. First step is to verify whether the verifier belongs to the group $\mathbb{U}$ and second step is to verify the validity of the signature.
  - *Judge Verifier:* Using the value $T_{i2} = kQ_i$, the verifier checks whether $e(T_{i2}, Q_0) \stackrel{?}{=} e(Q_i, U_1)$. If the verification holds then user with public key Q_i will do the next step in verification.
  - *Verify Signature:* Each designated verifier $a_i \in \mathbb{U}$ can verify the signature by performing the following steps.
    * Computes $\omega^{'} = We(D_i, U_3)e(P_{pub}, T_{i1})^{-1}$.
    * Computes $h^{'} = H_1(m, \omega^{'}, U_1, U_2, U_3)$.
    * Checks whether $e(V, T_{i2}) \stackrel{?}{=} e(h^{'} U_1, D_i)e(U_2, D_i)$.

  If the above check hold then the signature is valid. Otherwise the signature is invalid.

5.1 Security Proof for N-ADGS

Unforgeability Proof

Theorem 1. *Our N-ADGS scheme is existentially unforgeable under chosen message and identity attack if* ***CDHP*** *(Computational Diffie Hellman Problem) is hard in* $\mathbb{G}_1$.

This proof appears in the full version of the paper [10].

Unlinkability Proof

Theorem 2. *Our N-ADGS scheme is unlinkable in the sense that members outside the group cannot verify the signature if* ***DBDHP*** *(Decisional Bilinear Diffie Hellman Problem) is hard in* ($\mathbb{G}_1$, $\mathbb{G}_1$, $\hat{e}$).

This proof appears in the full version of the paper [10].

6 Conclusion

In this paper, we have presented attacks on two certificateless GIS schemes [5], [6], a certificateless BGOS scheme [7] and an identity based ADGS [8] scheme. We have proposed a new identity-based ADGS scheme. We leave as an open problem to construct efficient identity based ADGS with constant size signature independent of the number of designated verifiers. Our scheme is secure against existential forgery on adaptively chosen message and ID attack under the CDH assumption in the random oracle model and is unlinkable under the DBDH assumption.

References

1. Al-Riyami, S.S., Paterson, K.G.: Certificateless public key cryptography. In: Laih, C.-S. (ed.) ASIACRYPT 2003. LNCS, vol. 2894, pp. 452–473. Springer, Heidelberg (2003)
2. Boneh, D., Franklin, M.K.: Identity-based encryption from the weil pairing. SIAM J. Comput. 32(3), 586–615 (2003)
3. Hu, B.C., Wong, D.S., Zhang, Z., Deng, X.: Key replacement attack against a generic construction of certificateless signature. In: Batten, L.M., Safavi-Naini, R. (eds.) ACISP 2006. LNCS, vol. 4058, pp. 235–246. Springer, Heidelberg (2006)
4. Jakobsson, M., Sako, K., Impagliazzo, R.: Designated verifier proofs and their applications. In: Maurer, U.M. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 143–154. Springer, Heidelberg (1996)
5. Ma, C., Ao, F., He, D.: Certificateless group inside signature. In: Proceedings, April 2005, pp. 194–200 (2005)
6. Ma, C., Ao, J.: Certificateless group oriented signature secure against key replacement attack. Cryptology ePrint Archive, Report 2009/139 (2009), `http://eprint.iacr.org/`
7. Ma, C., He, D., Ao, J.: Broadcast group oriented signature. In: 2005 Fifth International Conference on Information, Communications and Signal Processing, pp. 454–458 (2005)
8. Ma, C., Li, J.: Adaptable designated group signature. In: Huang, D.-S., Li, K., Irwin, G.W. (eds.) ICIC 2006. LNCS, vol. 4113, pp. 1053–1061. Springer, Heidelberg (2006)
9. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)
10. Sree Vivek, S., Sharmila Deva Selvi, S., Gopinath, S., Pandu Rangan, C.: Breaking and building of group inside signature. Cryptology ePrint Archive, Report 2009/188 (2009), `http://eprint.iacr.org/`