Group Signatures *á la carte*[*]

Giuseppe Ateniese Gene Tsudik[†]

Abstract

A group signature scheme allows any member of a potentially large group to sign on behalf of the group. Group signatures are anonymous and unlinkable. Only a designated group manager can co-relate signatures and/or reveal the identity of the actual signer. At the same time, no one (including a group manager) can misattribute a valid signature. In this paper we construct a very efficient and provably secure group signature scheme.

1 Introduction

A group signature allows any group member to sign on behalf of the group. Group signatures are publicly verifiable but anonymous in that, no one, with the exception of a designated group manager, can establish the identity of the signer. Furthermore, group signatures are unlinkable which makes it computationally hard to establish whether or not multiple signatures are produced by the same group member. In exceptional cases (such as a legal dispute) any group signature can be "opened" by a group manager to reveal unambiguously the identity of the actual signer. At the same time, no one (including the group manager) can misattribute a valid group signature.

Definition 1.1. *A* group signature scheme *is a digital signature scheme comprised of the following procedures:*

- `SETUP` *an algorithm for generating the initial group public key* $\mathcal{Y}$.
- `JOIN` *a protocol between the group manager and a user that results in the user becoming a new group member.*
- `SIGN` *a protocol between a group member and a user whereby a group signature on a user-supplied message is computed by the group member.*
- `VERIFY` *an algorithm for establishing the validity of a group signature given a group public key and a signed message.*
- `OPEN` *an algorithm that, given a signed message and a group secret key, determines the identity of the signer.*

Group digital signatures are a relatively new concept introduced in 1991 [4]. A number of improvements and enhancements followed. Some of the notable results are due to Camenisch [1], Petersen [5] and Camenisch/Stadler [2]. This paper presented a very efficient and provably secure group signature scheme. Both its performance and its security appear to significantly surpass those of prior art. Both the size of the group public key and the length of a group signature are constant. However, we do not address hostile coalitions in the main scheme. Briefly, this means that a subset of group members can pool together their secrets and generate perfectly valid group signatures. Such signatures have the property that even the group manager is unable to open them. However, it should be noted that this property does not contradict the definition of a secure group signature scheme. The emphasis in this work is on a *single* member being unable to generate group signatures that are (in case of a dispute) not traceable to her by the group manager.

2 Preliminaries

In this section we briefly describe two types of building blocks necessary in the subsequent design of our group signature scheme.

So-called "proof-of-knowledge" systems allow a prover to demonstrate knowledge of a secret such that no useful information is revealed in the process. In this paper, we take advantage of Schnorr signature [6] based schemes in order to show knowledge of secrets without knowing the order of the group in which one operates. Our primary motivation for this choice is its computational efficiency. Substantially, these are signature schemes based on proofs of knowledge performed non-interactively making use of an ideal hash function (a la Fiat-Shamir). According to [2], we will call these schemes *signatures of knowledge*. Let $n = pq$ be a RSA-like modulus, $a, b \in \mathbb{Z}_n^*$ elements of large multiplicative order modulo both p and q and m a generic message. Throughout the paper we will use the following notation: $\mathrm{SKLOG}(m, a^x, a)$ denotes the Schnorr signature (allowing to prove the knowledge of x) and $\mathrm{SKRNG}(m, a^x, b^y, a, b,]i_1, i_2[)$ denotes an instance of a technique allowing a prover to convince a verifier that $x - y \in]i_1, i_2[$ (related to a protocol presented in [3]).

[*]Research supported by the Defense Advanced Research Project Agency, Information Technology Office, under contract DABT63-97-C-0031.

[†]USC Information Sciences Institute, 4676 Admiralty Way, Marina Del Rey, CA 90292, USA. Email:{ateniese,gts}@ISI.EDU

3 Our Group Signature Scheme

We now turn to the actual construction of our group signature scheme. The initial phase involves the group manager ($GRMGR$) setting the group public key $\mathcal{Y}$.

SETUP:

1. Select secret random RSA-like primes p, q, p' and q' such that $p = 2p' + 1$, $q = 2q' + 1$ achieving a modulus $n = pq$;
2. Select two secret random exponents y, z, and a public prime v;
3. Finally, choose an element $a \in \mathbb{Z}_n^*$ of large multiplicative order modulo both prime factors of n. and compute: $Y = a^{-y}$ and $Z = a^z$
4. The group public key is: $\mathcal{Y} = (n, v, a, Y, Z)$.

The group public key $\mathcal{Y}$ is made available via the usual means (i.e., embedded in some form of a public key certificate signed by a trusted authority.) We note that, in practice, components of $\mathcal{Y}$ must be verifiable to prevent framing attacks. There are efficient methods providing this kind of proofs and, moreover, they must be done only once. For instance, to verify that a has large order in $\mathbb{Z}_n^*$, it is enough to test whether $a \neq -1, 1$ and that $\gcd(a-1, n) = 1$. This proves that a has order at least $p'q'$. $GRMGR$ also needs to provide a proof of n being a product of two primes p, q with $p = 2p' + 1$ and $q = 2q' + 1$.

Suppose now that an user wants to join the group. We assume that communication between the user and the group manager is secure, i.e. private and authentic.

JOIN:

1. User generates a secret exponent ℓ_1 and sends a^{ℓ_1} to $GRMGR$;
2. $GRMGR$ sends a random number ℓ_2 to the user
3. User computes $x = \ell_1 \cdot \ell_2$ and sends $GRMGR$: a^x and SKLOG(“$join$”, a^x, a)
4. $GRMGR$ verifies SKLOG and that $(a^{\ell_1})^{\ell_2} \equiv a^x \bmod n$ and sends the new membership certificate $A = a^{(x+y)v^{-1}}$ to the user.

As part of `JOIN`, the new member must also prove to $GRMGR$ that the exponent x is such that $0 < x < v$ (see [3]). Moreover, $GRMGR$ creates a new entry in the membership table, computes $ID = A^z$ and stores $\{ID, A, \texttt{JOIN}$ transcript$\}$ in the new entry. (`JOIN` transcript is formed by the messages received from the user in the steps above. It is assumed to be signed by the user with some form of a long-term credential.)

Armed with a membership certificate, a group member can generate anonymous and unlinkable group signatures on a generic message m:

SIGN:

1. Generate random value $r \in \mathbb{Z}_n^*$ and compute:
   $T_1 = A \cdot a^r$
   $T_2 = A^{-1} \cdot (Z \cdot a^{-1})^r$
2. Construct:
   $\mathrm{SKLOG}_1 = \mathrm{SKLOG}(m, a^{x+rv}, a)$
   $\mathrm{SKLOG}_2 = \mathrm{SKLOG}(m, a^{zr}, a^z)$
   $\mathrm{SKRNG} = \mathrm{SKRNG}(m, a^{x+rv}, a^{zrv}, a, a^z,]0, v[)$
3. Send: $\{T_1, T_2, \mathrm{SKLOG}_1, \mathrm{SKLOG}_2, \mathrm{SKRNG}\}$

Note that:
$T_1 = a^{(x+y)v^{-1}+r}$ and $T_2 = a^{-(x+y)v^{-1}+r(z-1)}$.

Verifier checks the validity of the signature on message m as follows:

VERIFY:

1. Compute:
   a) $W_1 = (T_1^v) \cdot Y \; (= a^{x+rv})$
   b) $W_2 = (T_1 \cdot T_2) \; (= a^{rz} = Z^r)$
2. Verify the signatures of knowledge:
   a) SKLOG(m, W_1, a) which shows that the prover is a group member.
   b) SKLOG(m, W_2, Z) which proves knowledge of r and assures the verifier that the signature can be `OPEN`-ed later if necessary.
   c) SKRNG($m, W_1, W_2^v, a, a^z,]0, v[$) which proves the signer is using a value signed by $GRMGR$.

In the event that the actual signer must be subsequently identified (e.g., in case of a dispute) $GRMGR$ executes the following procedure:

OPEN:

1. Check signature validity via `VERIFY` procedure.
2. Compute $ID = (T_1)^{z-1} \cdot T_2^{-1} = A^z$.
3. Look up ID in the membership table to identify the signer.

References

[1] J. Camenisch. Efficient and generalized group signatures. In *Advances in Cryptology – EUROCRYPT*, 1997.

[2] J. Camenisch and M. Stadler. Efficient group signature schemes for large groups. In *Advances in Cryptology – CRYPTO*, 1997.

[3] A. Chan, Y. Frankel and Y. Tsiounis. Easy come - easy go divisible cash. In *Advances in Cryptology – EUROCRYPT*, 1998.

[4] D. Chaum and E. van Heyst. Group signatures. In *Advances in Cryptology – EUROCRYPT*, 1991.

[5] H. Petersen. How to convert any digital signature scheme into a group signature scheme. In *Security Protocols Workshop*, 1997.

[6] C. Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, 1991.