How to convert any ID-based Signature Schemes into a Group Signature Scheme

Claude Castelluccia *INRIA Rhône-Alpes*
655, avenue de l'Europe
38330 Montbonnot, France
claude.castelluccia@inrialpes.fr

August 2002

Abstract

This paper describes how any Identity Based Signature schemes can be used to implement a Group Signature scheme. The performance of the generated Group Signature scheme is similar to the performance of the underlying ID-based Signature scheme. This makes our proposal very attractive since most of existing group signature schemes that have been proposed so far are grossly inefficient. In contrast, ID-based signature schemes can be very efficient especially if they use elliptic curves and pairing.

keywords: Group Signature, ID-based Signature schemes

1 Introduction

An Identity based crypto-system [1, 2] is a system that allows a publicly known identifier (email address, IP address, name,...) to be used as the public key component of a public/private key pair in a crypto-system. The scheme assumes the existence of a *Private Key Generator* whose sole purpose is to compute for each user the private key associated with the identifier they want to use as Public Key. The scheme is ideal for closed groups of users. Several ID-based signature schemes have been proposed in the last 20 years [1, 3, 4, 5]. Some of these schemes use Elliptic Curve (EC) algorithms and are therefore particurly efficient.
A Group Signature, introduced by Chaum and van Heijst [6], allows any member of a group to digitally sign a document such that a verifier can confirm that it came from the group but does not know which individual in the group signed the document. The scheme assumes the existence of a *Group Controller* whose sole purpose is to compute for each user a private key that the user should use when signing a message on behalf on the group. A user verifies a signature with the *Group Public Key* that is usually constant and unique for the whole group (i.e. independent of the members). Many group signature schemes have been proposed [6, 7, 8, 9]. However all of them are much less efficient that regular signature schemes (such as DSA or RSA). Designing an efficient group signature scheme is still an open research problem.

In this paper we outline the similarities that exist between ID-based signature and Group signature schemes. We also show that any ID-based signature schemes can be used to implement a group signature scheme. Such group signature has the same performance than the performance of the ID-based signature scheme it is derived from. This makes our proposal very attractive since it is probably the most efficient group signature scheme that exists today. The rest of the paper is organized as follows. Section 2 presents the formal model of a secure ID-based signature scheme. Section 3 presents the formal model of a secure group signature scheme. Section 4 describes how any ID-based signature schemes could be used to implement a group signature scheme. This section also access the security of the new group signature scheme. Finally, the paper concludes in Section 5.

2 Identity-Based Signature

An Identity based crypto-system [1, 2] is a system that allows a publicly known identifier (email address, IP address, name,...) to be used as the public key component of a public/private key pair for the purposes of digital signature [1, 3, 4, 5], encryption [2] and key agreement [10]. The private key component is computed by the *Private Key Generator* (PKG) and sends to the corresponding node via a secure and authentic channel.

Definition 1. *An Identity based signature scheme is a digital signature scheme specified by the following four algorithms*:

- **ID_SETUP:** An algorithm, executed by the PKG, that takes a (random) parameter k as input and generates from it *params* (system parameters) and *master-key*. *Params* is publicly known, while *master-key* is only known to the PKG.
- **ID_EXTRACT:** An algorithm, executed by the PKG, that takes as input *params*, *master-key* and an arbitrary $ID_i \in \{0,1\}^*$, provided by a user, $User_i$, and returns a private key d_i. ID_i is an arbitrary string that is used as a public key and d_i is the corresponding private key.
- **ID_SIGN:** An algorithm that takes as input *params*, d_i and a message, m and returns a signature Sig defined as follows:
  $Sig = ID\_SIGN(params, d_i, m)$.
- **ID_VERIFY:** An algorithm that takes as input a message m and its signature Sig, the system params *params* and a public key ID_i and performs
  $valid = ID\_VERIFY(Sig, ID_i, params, m)$. *Valid* is a binary value that is set to 0 if the signature is invalid and to 1 if the signature is valid.

A secure ID-based signature scheme must at least satisfy the following properties:

- *Correctness:* Signatures produced by a user using ID_SIGN must be accepted by ID_VERIFY (provided that the correct parameters are used).
- *Unforgeability:* It is computationally hard for everyone that do know the secret key d_i of $User_i$ to forge his signatures. As a consequence, it must be computationally hard for everyone to retrieve from $params$ the corresponding $master - key$.
- *Coalition-resistance:* A colluding subset of users, that have received their private key from the same PKG and $params$, cannot generate a valid signature that the PKG cannot link to one of the colluding users.

3 Group Signature

A group-signature, introduced by Chaum and van Heijst [6], allows any member of a group to digitally sign a document such that a verifier can confirm that it came from the group but does not know which individual in the group signed the document. More formally, a group signature is defined as follows [7, 8]:

Definition 2. A group-signature scheme is *a digital signature scheme comprised of the following five procedures:*

- **G_SETUP:** On input a security parameter k, the algorithm outputs the initial group public key, *GroupPK* (including all system parameters) and the secret, *master-key*, for the group manager. This algorithm is executed by the group manager upon creation of a new group.
- **G_JOIN:** A protocol between the group manager and a user, $User_i$ that results in the user becoming a new group member. The user's output is a membership secret, S_i.
- **G_SIGN:** An algorithm that on input a group public key, *GroupPK*, a membership secret, S_i, and a message, m, outputs the group signature, $GSig$, of m.
$GSig = G\_SIGN(GroupPK, S_i, m)$
- **G_VERIFY:** An algorithm for establishing the validity of an alleged group signature of a message with respect to a group public key. This algorithm takes as input the group public key, $GroupPK$, the message, m, and its group signature, $GSign_m$ and output 1 is the signature is valid as follows:
$valid = G\_VERIFY(GSig, GroupPK, m)$.
- **G_OPEN:** An algorithm that, given a message, a valid group signature on it, a group public key and a group manager's secret key, determines the identity of the signer. Note that only the group manager is able to perform this operation.

A secure group signature scheme must satisfy the following properties:

- *Correctness:* Signatures produced by a group member using *G_SIGN* must be accepted by G_VERIFY.
- *Unforgeability:* Only group members are able to sign messages on behalf of the group.
- *Anonymity:* Given a valid signature, identifying the actual signer is computationally hard for everyone but the group manager.
- *Unlinkability:* Deciding whether two different valid signatures were computed by the same group member is computationally hard.
- *Exculpability:* Neither a group member nor the group manager can sign on behalf of other members.
- *Traceability:* The group manager is always able to open a valid signature and identify the actual signer.
- *Coalition-resistance:* A colluding subset of group members cannot generate a valid signature that the group manager cannot link to one of the colluding group members.

4 Building a Group Signature from a Identity-Based Signature

If we consider that, in the ID-based signature scheme, all users that get a private key (from their ID) from the same $params$ and $master - key$ parameters form a group (identified by the parameter $params$), the concepts of ID-based signatures and group signatures are very similar.
The *ID_SETUP* and *G_SETUP* algorithms perform similar operations. In fact, both of them take as input a random parameter, k, and generate from it:

- a public parameter, respectively $params$ and $GroupPK$, that are used by a user to verify the signatures generated by a member of a group or a ID-based system.
- a private parameter, $master - key$, used by the PKG or the group controller to generated each member's private key component.

The *ID_EXTRACT* and *G_JOIN* phases are also very similar. They both generate from some input provided by a user, some cryptographic materials, respectively d_i and S_i, that are needed by the users to sign their messages and prove that they have been authorized by a central authority (namely the PKG or the Group Controller). Furthermore, if we set that:

- $Params = GroupPK$,
- $d_i = S_i$
- $GSig =< Sig, ID_i >$, i.e. $GSig$ is the concatenation of Sig and ID_i.

ID_SIGN can be rewritten as follows:

- $GSig = ID\_SIGN(GroupPK, S_i, m)$

ID_SIGN is then a function that takes as input a publicly available identifier of the group, *GroupPK*, a secret, S_i, and a message, m and outputs the group signature, $GSig$, of m. This new definition is actually the definition of a group signature, G_SIGN, as defined in Section 3. Therefore any ID_SIGN function can be used to implement a G_SIGN function.
Similarly ID_VERIFY can be rewritten as follows:

- $valid = ID\_VERIFY(GSig, GroupPK, m)$

ID_VERIFY is then a function that takes as input the group parameter, *GroupPK*, the message, m, and its signature, $GSig$ and outputs 1 is the signature is valid. This is actually the definition of a group signature verification, G_VERIFY, as defined in Section 3. Therefore any ID_VERIFY function can be used to implement a G_VERIFY function.

As a result of this comparison, we can conclude that by using $params$ as a group public key and by defining a group signature as the concatenation of a ID-based signature with the user's public key, ID_i, any ID-based signature scheme can be used to implement a group signature scheme.

4.1 Basic Scheme

In this section we present, in more details, how a ID-based signature scheme can be used to implement a group signature scheme. In this description, the Group Controller is also a PKG. The protocol then works as follows:

- **G_SETUP:** the Group Controller (i.e the PKG) generates from some random k the system parameters, *params*, and the $master - key$. *Params* is thereafter used as the group public key, i.e. as *GroupPK*.

- **G_JOIN:** When a user, $User_i$, becomes part of the group it contacts the Group Controller and provides its public key ID_i. The Group Controller then generates from it and from *params* and $master - key$ the corresponding private key, d_i (according to the $ID\_EXTRACT$ algorithm). This private key is communicated secretly to $User_i$.

- **G_SIGN:** $User_i$ signs a message, m, by using d_i, and *params* in the algorithm described in Section 2. The group signature, $GSign$, is then the concatenation of the previously generated signature, Sig, and $User_i$'s public key, ID_i. In other words,
$GSig =< Sig, ID_i >$, and $G\_SIG = ID\_SIGN(params, d_i, m)$.

- **G_VERIFY:** A user verifies that the signature was generated by the group by using the algorithm specified in Section 2, i.e.:
$valid = ID\_VERIFY(Sig, ID\_i, params, m)$.
Note that only a host that has received its private key, d_i, from the PKG, i.e. that is an authorized member of the group, could have signed the message.

- **G_OPEN:** The group manager knows for each ID_j the identity of the user, $User_j$, that is associated with it. This binding is established during the G_JOIN phase. As a result, it is easy for a group manager, given a message and a valid group signature $< Sig, ID_j >$ to determine the identity of the signer.

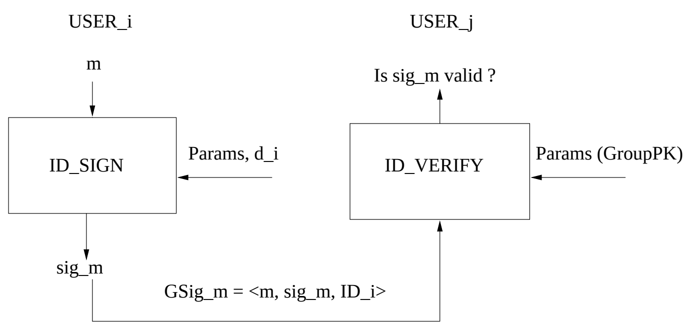

Figure 1: Group Signature Scheme

4.2 Security Analysis

In this section, we access the security of the group signature scheme defined in Section 4.1 according to the security properties defined in Section 3.

- *Correctness:* This property is guaranteed since any ID-based signature schemes must guaranteed it too.

- *Unforgeability:* This property is guaranteed since any ID-based signature schemes must guaranteed it too.

- *Anonymity:* In our scheme, a group signature is the concatenation of the identity based signature with the user's public key (i.e. ID). Therefore if the underlying Identity based signature provides anonymity and if the user's public key does not reveal any information about the user, anonymity is guaranteed by the group signature scheme.

- *Unlinkability:* In our scheme, a group signature is the concatenation of the identity based signature with the user's public key (i.e. ID). As a result, all the signatures generated by a user will contains his public key. Therefore unlinkability is not provided. However if the underlying identity-based signature provides unlinkability and if a user uses a different public/private key pair for each signature, unlinkability is then provided. This solution might not be very practical if the user has to sign a lot of messages (because it needs to get and store a lot of public/private key pairs) but is acceptable otherwise.

- *Exculpability:* In our proposal, a group member can not sign on behalf of other members because it does not know the other members' private keys. However the group manager (i.e. the PKG) knows each users' private key. It can therefore sign on behalf of any member. Exculpability is therefore not provided. As described in Section 4.3, the basic protocol can be extended to provide this property at the cost of adding an extra signature.

- *Traceability:* Since, in our proposal, the group manager generates each member private keys from their public keys, it can easily identify the actual signer of a valid signature by looking at the public key component in the group signature. Note that this property is guaranteed even if the exculpability extension (as described in Section 4.3 is used). Traceability is therefore provided.

- *Coalition-resistance:* This property is guaranteed since any ID-based signature schemes must guaranteed it too.

4.3 The Exculpability Extension

This section describes how to extend the protocol presented in Section 4.1 to provide *Exculpability.* The new protocol works as follows:

- *G_SETUP:* as in the basic scheme.

- *G_JOIN:* When a user, $User_i$, becomes part of the group it contacts the Group Controller and sends it its public key ID_i.
  Note that in the basic scheme, ID_i is an arbitrary string. In the proposed extension, ID_i is actually the public component of a (RSA or DSA) signature public/private key pair generated by the user itself [1]. This public/private key pair will be referred as (PK_i, SK_i) in the remainder of this paper. Therefore ID_i is set to PK_i. The Group Controller then generates from ID_i and from $params$ and $master - key$ the corresponding (group) private key, d_i (according to the extract algorithm). This private is communicated secretly to $User_i$.

- *G_SIGN:* $User_i$ signs a message, m, with its private key, i.e. SK_i, and the corresponding signature scheme (i.e. RSA or DSA). This generates $Sig0$.
  $Sig0 = RSA\_SIGN(SK_i, m)$.
  It then re-sign m with the algorithm described in Section 2 and the cryptographic paramaters d_i, and $params$. This generates Sig.
  $Sig = ID\_SIGN(params, d_i, m)$.
  The group signature, $GSign_m$, is then the concatenation of the previously generated signatures, $Sig0$ and Sig, with the $User_i$'s public key, ID_i. In other words,
  $GSig =< Sig0, Sig, ID_i >$.

- *G_VERIFY:* A user verifies that the signature was generated by the group by verifying using the algorithm specified in Section 2 that Sig is valid and therefore the $User_i$ is an authorized member of the group.
  $valid = ID\_VERIFY(Sig, ID_i, params, m)$. A user verifies that the signature was generated by $User_i$ and not by the group manager by verifying using the $User_i$'s public key (i.e. ID_i) and the corresponding signature scheme (i.e. DSA or RSA) that $Sig0$ is valid.

[1] It might be more convenient to use a hash of the Public Key, as in [11], instead of the Public Key itself.

$valid = RSA\_VERIFY(Sig0, ID_i)$. Since the Group Controller does not know the private key SK_i it will not be able to generate a valid $Sig0$. This property provides exculpability.

- *G_OPEN:* as in the basic scheme.

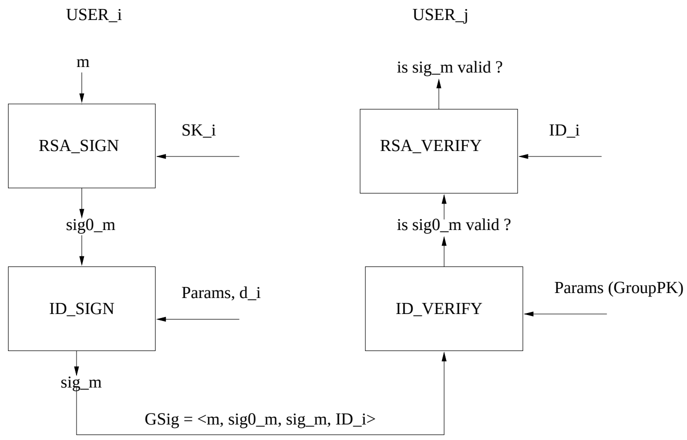

Figure 2: Group Signature Scheme with Excupability extension

This extension has a performance cost since it adds one (RSA or DSA) signature. However this extension is optional and should not be used if exulpability is not needed. Furthermore even with this extra cost, we believe that our scheme is still more efficient that any existing group signatures. Note that with this extension, a user can ulteriorly prove that he signed a message by revealing the RSA private key that was used. This is an interesting property for some applications. For example [12] explains how a group signature scheme can be used to submit tenders. In this example, all companies submitting a tender form a group and each company signs its tender anonymously using the group signature. Later when the preferred tender has been selected the signer can be identifier by the trusted authority, whereas the signers of all other tenders will remain anonymous. This example however requires that all members have a complete trust to the trusted authority. In fact, what prevents the trusted authority from revealing the identity of other members? With our scheme, users can get their signing key anonymously from the trusted authority. The winner can then prove that he was the signer of the selected tender by revealing its RSA private key.

5 Conclusion

This paper describes how any identity based signature can be converted into a group signature. The generated group signature can handle large groups since the group public key and parameters are constant and do not depend on the group members. The security of such a group signature depends on the security of the ID based signature scheme it was derived from. We show that the following properties are provided: correctness, anonymity, traceability, coalition-resistance, and optionally exculpability. Unlinkability is not provided, unless a group member uses a different keying material for each signature. This might be acceptable if the member signs few messages. The generated group signature performance is similar to the performance of the underlying ID based signature scheme. We believe this is a very good result since most of existing group signature scheme that have been proposed so far are grossly inefficient. ID-based signature schemes can be very efficient especially if they use elliptic curves and pairing [5, 3, 13, 4].

References

[1] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptography - CRYPTO'84*, Springer, Ed., 1984, pp. 47–53.

[2] D. Boneh and M. Franklin, "Identity based encryption from weil pairing," in *Advances in Cryptography - CRYPT0 2001*, Santa Barbara, CA, August 2001.

[3] J.C. Cja and J. H. Cheon, "An identity-based signature signature from gap diffie-hellman problem," Tech. Rep., IACR Cryptology ePrint Archive: Report 2002/018, http://eprint.iacr.org/2002/018/, 2002.

[4] K. Paterson, "Id-based signatures from pairings on elliptic curves," Tech. Rep., IACR Cryptology ePrint Archive: Report 2002/004, http://eprint.iacr.org/2002/004/, 2002.

[5] R. Sakai, K. Ohgishi, and M. Kasahara, "Cryptosystems based on pairing," in *SCIC2000*, Okinawa, Japan, January 2000.

[6] D. Chaum and E. Van Heyst, "Group signatures," in *Advances in Cryptography - EUROCRYPT'91*, Springer-Verlag, Ed., 1991, vol. 547, pp. 257–265.

[7] G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik, "A practical and provably secure coalition-resistant group signature scheme," in *Advances in Cryptography - CRYPT0 2000*, Santa Barbara, CA, August 2000.

[8] J. Camenish and M. Michels, "A group signature with improved efficiency," in *Advances in Cryptography - ASIACRYPT'98*, Springer-Verlag, Ed., 1998, vol. 1514, pp. 160–174.

[9] Y.-M. Tseng and J.-K. Jan, "A novel id-based group signature," *Information Sciences*, pp. 131–141, 1999.

[10] N. P. Smart, "An identity based authenticated key agreement protocol based on the weil pairing," Tech. Rep., IACR Cryptology ePrint Archive: Report 2001/111, http://eprint.iacr.org/2001/111/, 2001.

[11] G. Montenegro and C. Castelluccia, "Statistically Unique and Cryptographically Verifiable (SUCV) identifiers and addresses," in *NDSS'02*, February 2002.

[12] L. Chen and T.P. Pedersen, "New group signature schemes," in *Advances in Cryptography - EUROCRYPT'95*, Springer-Verlag, Ed., 1995, vol. 950, pp. 171–181.

[13] F. Hess, "Exponent group signature schemes and efficient identity based signature schemes based on pairings," Tech. Rep., IACR Cryptology ePrint Archive: Report 2002/012, http://eprint.iacr.org/2002/012/, 2002.