# On the Minimal Assumptions of Group Signature Schemes*

Michel Abdalla[1] and Bogdan Warinschi[2]

[1] Departement d'Informatique
École Normale Supérieure
45 rue d'Ulm, 75230 Paris Cedex 05, France
E-mail: Michel.Abdalla@ens.fr
URL: http://www.michelabdalla.net
[2] Computer Science Department
University of California at Santa Cruz
1156 High Street, Santa Cruz, CA 95064, USA
E-mail: bogdan@cse.ucsc.edu
URL: http://www.cs.ucsd.edu/~bogdan

**Abstract.** One of the central lines of cryptographic research is identifying the weakest assumptions required for the construction of secure primitives. In the context of group signatures the gap between what is known to be necessary (one-way functions) and what is known to be sufficient (trapdoor permutations) is quite large. In this paper, we provide the first step towards closing this gap by showing that the existence of secure group signature schemes implies the existence of secure public-key encryption schemes. Our result shows that the construction of secure group signature schemes based solely on the existence of one-way functions is unlikely. This is in contrast to what is known for standard signature schemes, which can be constructed from any one-way function.

**Keywords.** Group signatures, one-way functions, trapdoor permutations, minimal assumptions.

## 1 Introduction

Motivation. One of the central lines of cryptographic research is identifying the weakest assumptions required for the construction of secure primitives. This is important not only to better understand the different relations among existing primitives, but also to learn the minimal conditions without which a certain primitive cannot exist. Yet another reason for finding the weakest assumptions is that stronger assumptions may later be found to be false while weaker assumptions may still hold. Therefore, by closing the gap between which primitive is sufficient and what is necessary to build a given cryptographic function such

* In J. Lopez, S. Qing, and E. Okamoto, editors, International Conference on Information and Communications Security – ICICS 2004, Volume 3269 of LNCS, pages 1–13, Malaga, Spain, October 27–29, 2004. Springer-Verlag, Berlin, Germany.

as encryption or group signatures, one can determine the exact conditions that need be met for them to exist.

While several implications and separations are known in the literature for primitives such as standard signatures and public-key encryption, very little is known for group signatures despite the intuition that the latter appears to be a stronger primitive than standard signatures. Currently, group signatures are only known to be implied by trapdoor permutations [9] and to imply one-way functions [30], a quite large gap. Addressing this problem is the main goal of this paper.

Preliminaries. In order to better understand our results, let us briefly recall the definitions for the basic primitives given in Figure 1. The most basic of the cryptographic primitives is a *one-way function*. Loosely speaking, a function is said to be one-way if it is easy to compute (on any input) but hard to invert (on average), where easy means computable in polynomial time on the length of the input. Another basic primitive is a *trapdoor one-way function*, or simply trapdoor function, introduced by Diffie and Hellman [16] in the seminal work which laid out the foundations of public-key cryptography. Informally, a one-way function is said to be trapdoor if it has associated to it a secret trapdoor which allows anyone in its possession to easily invert it. The notions of *one-way permutations* and *trapdoor permutations* are defined in a similar manner. The notion of *trapdoor predicates*, introduced by Goldwasser and Micali [21], is slightly different. Approximately, trapdoor predicates are probabilistic functions over $\{0,1\}$ which are easy to compute given a public key but whose output distributions on inputs 0 and 1 are hard to distinguish by any algorithm not in possession of the trapdoor information.

Since we will be using terms such as implications and separations throughout the paper, we should also recall what we mean by that. Consider for example two cryptographic primitives $S$ and $P$. In order to properly relate their security, one usually makes use of reductions. More precisely, a primitive $P$ is said to *imply* a primitive $S$ if the security of $P$ has been demonstrated to imply the security of $S$. More precisely, we use this phrase when someone has formally defined the goals $G_P$ and $G_S$ for primitives $P$ and $S$, respectively, and then has proven that the existence of an adversary $A_S$ who breaks primitive $S$, in the sense of violating $G_S$, implies the existence of an adversary $A_P$ who breaks primitive $P$, in the sense of violating $G_P$.

Proving a separation between two primitives, however, is a more subtle problem since it is not clear what it means to say that a given primitive does not imply another primitive. To overcome this problem, one usually uses the method due to Impagliazzo and Rudich [25] of restricting the class of reductions for which the separation holds. More specifically, they noted the fact that the vast majority of the reductions in cryptography uses the underlying primitive as a black-box and based on that, they introduced a method for proving separations between primitives with respect to these types of reductions.

Background on group signatures. The notion of group signatures was introduced by Chaum and van Heyst [14] and describes a setting in which indi-

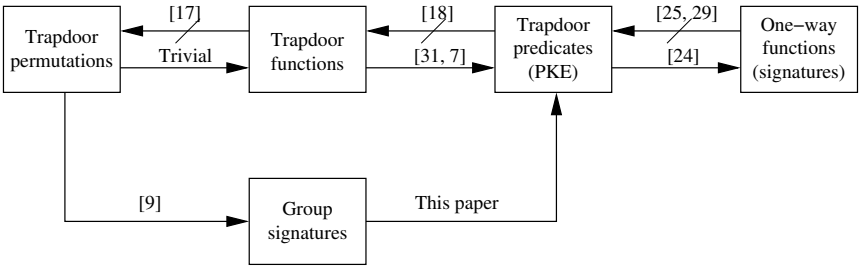


**Fig. 1.** Implications and black-box separations between primitives.

viduals within a group can sign messages with respect to the group. According to [14], a secure group signature scheme should satisfy two basic requirements, anonymity and traceability. While the former says that the identity of the signer should remain unknown to anyone verifying the signature including other group elements, the latter asks that there should exist an entity, called the group manager, capable of revoking the anonymity of signer whenever necessary.

Since the original work of Chaum and van Heyst [14], several other schemes have been proposed in the literature (e.g., [1, 3, 2, 15, 13, 12, 26]), each with its own set of security properties and requirements. It was only recently, however, that a formal model of security for group signatures was put forward [9], combining the increasing set of security requirements into two basic properties, called full-anonymity and full-traceability. These two basic properties were shown to imply in the case of static groups all of the existing security properties of previous scheme. Subsequent works also give formal definitions for dynamic groups [27, 10].

Such formal definitions have many benefits. They not only allow for concrete and simpler proofs of security (only two properties need be satisfied), but they also allow us to better understand what it means to be a secure group signature scheme and its implications. It also allows us to draw precise relations between group signatures and other cryptographic primitives. In fact, the implications proven in this paper are only possible in the presence of such formal models of security.

Contributions. In this paper, we provide the first step towards closing the gap between what is known to be sufficient to construct secure group signatures and what is known to be necessary. We do so by showing that group signatures imply public-key encryption and thus are unlikely to be constructed based solely on the existence of one-way functions (see Figure 1).

The separation between group signatures and one-way functions is a direct consequence of our work and that of Impagliazzo and Rudich [25] which showed that any such construction would either make use of non-black-box reduction techniques or prove along the way that $P \neq NP$. Recently, in [29], Reingold, Trevisan, and Vadhan improved on that by removing the condition that $P \neq$

$NP$. In other words, such construction would definitely have to rely on non-black-box reduction techniques. The implications of such results are of great importance since almost all reductions in cryptography are black-box.

Related work. Over the years, several results proving either implications or separations among different primitives appeared in the literature. Among the results that are more relevant to our work are those for signatures and public-key encryption.

Since the work of Goldwasser, Micali, and Rivest [22] proposing the construction of a secure signature scheme based on claw-free pairs and laying out the foundations of standard signatures, several other works followed aiming at establishing the weakest computational assumptions on which signature schemes could be based. The first of these works was the one of Bellare and Micali [8] showing how to construct signature schemes based on any trapdoor permutations. Their work was soon followed by the work of Naor and Yung [28] showing how to build signatures from any universal one-way hash functions and by the work of Rompel [30] showing how to build signatures from any one-way function. The latter is in fact also known to be a necessary assumption.

The picture in the case of public-key encryption and other primitives that are known to be implied by it (e.g., key exchange) is not as clear as in the case of standard signatures and is still the subject of active research [29, 18, 17, 7]. Several of these results are discussed in Section 4,

Another work that is similar in spirit to our work is the one of Halevi and Krawczyk [23] which shows that password-based authentication protocols imply public-key cryptography.

Organization. In Section 2 we recall the formal models and security definitions for (static) group signatures and public-key encryption schemes. Next, in Section 3, we show how to build a secure public-key encryption scheme from a secure group signature scheme. We then prove the security of our construction based on the anonymity property of group signatures. Finally, we conclude our paper by discussing the implications of our result in Section 4.

## 2 Definitions

### 2.1 Preliminaries

We will denote by $|m|$ the bit-length of a bit-string $m$. For any two arbitrary bit-strings $m_0$ and $m_1$ with $|m_0| = |m_1|$ we denote by $\mathsf{diff}(m_0, m_1) = \{i | m_0[i] \neq m_1[i]\}$, i.e. the set of bit positions on which $m_0$ and $m_1$ are different.

As usual, a function $f(\cdot)$ is said to be negligible if for any polynomial $p$, there exists a natural number $n_p$ such that $f(n) \leq \frac{1}{p(n)}$ for all $n_p \leq n$. We will say that a function of two arguments $f(\cdot, \cdot)$ is negligible, if for all polynomials $p$, the function $g$ defined by $g(k) = f(k, p(k))$ is negligible.

### 2.2 Public Key Encryption Schemes

Encryption schemes. A public-key encryption scheme $\mathcal{AE} = (\mathsf{K_e}, \mathsf{Enc}, \mathsf{Dec})$ is specified, as usual, by algorithms for key generation, encryption and decryption. The security property that is most relevant for the results of this paper is *indistinguishability under chosen-plaintext attack*, in short IND-CPA.

For completeness we now recall the definition. An (IND-CPA) adversary against $\mathcal{AE}$ is an algorithm $A$ that operates in two stages, a choose stage and a guess stage. For a a fixed bit $b$, the adversary works as follows. In the first stage the algorithm is given a public key $pk_e$ for encryption, and at the end of this stage it outputs a pair of messages $M_0$ and $M_1$. The input of the algorithm to the second stage is some state information, also produced at the end of the first stage, and a challenge ciphertext $C$ that is an encryption of $M_b$. At the end of the second stage the adversary outputs a guess bit $d$ that selects one or the other message. The adversary wins if he guesses successfully which of the messages was encrypted.

Let $\mathbf{Exp}^{\text{ind-cpa\_}b}_{\mathcal{AE},A}(k)$ denote the random variable representing the output of $A$ in the above experiment, when $pk_e$ is obtained by running the key generation algorithm (with fresh coins) on security parameter $k$. The advantage function of $A$ is defined as:

$$\mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{AE},A}(k) = \Pr\left[\mathbf{Exp}^{\text{ind-cpa\_1}}_{\mathcal{AE},A}(k) = 1\right] - \Pr\left[\mathbf{Exp}^{\text{ind-cpa\_0}}_{\mathcal{AE},A}(k) = 1\right]$$

An encryption scheme $\mathcal{AE}$ is said to be IND-CPA secure if the advantage function $\mathbf{Adv}^{\text{ind-cca}}_{\mathcal{AE},A}(\cdot)$ is negligible for any polynomial-time adversary $A$.

### 2.3 Group Signatures

In this section we recall the relevant definitions regarding group signatures. The presentation in this section follows [9].

Syntax of group signature schemes. A *group signature scheme* $\mathcal{GS} = (\mathsf{GKg}, \mathsf{GSig}, \mathsf{GVf}, \mathsf{Open})$ consists of four polynomial-time algorithms:

- The randomized *group key generation* algorithm GKg takes input $1^k, 1^n$, where $k \in \mathbb{N}$ is the security parameter and $n \in \mathbb{N}$ is the group size (ie. the number of members of the group), and returns a tuple $(gpk, gmsk, \boldsymbol{gsk})$, where $gpk$ is the *group public key*, $gmsk$ is the *group manager's secret key*, and $\boldsymbol{gsk}$ is an $n$-vector of keys with $\boldsymbol{gsk}[i]$ being a *secret signing key* for player $i \in [n]$.
- The randomized *group signing* algorithm GSig takes as input a secret signing key $\boldsymbol{gsk}[i]$ and a message $m$ to return a signature of $m$ under $\boldsymbol{gsk}[i]$ ($i \in [n]$).
- The deterministic *group signature verification* algorithm GVf takes as input the group public key $gpk$, a message $m$, and a candidate signature $\sigma$ for $m$ to return either 1 or 0.
- The deterministic *opening* algorithm Open takes as input the group manager secret key $gmsk$, a message $m$, and a signature $\sigma$ of $m$ to return an identity $i$ or the symbol $\perp$ to indicate failure.

Experiment $\mathbf{Exp}^{\text{anon-b}}_{\mathcal{GS},A}(k,n)$
$(gpk, gmsk, \boldsymbol{gsk}) \xleftarrow{\$} \mathsf{GKg}(1^k, 1^n)$
$(\mathrm{St}, i_0, i_1, m) \xleftarrow{\$} A^{\mathsf{Open}(gmsk,\cdot,\cdot)}(\mathsf{choose}, gpk, \boldsymbol{gsk})$ ; $\sigma \xleftarrow{\$} \mathsf{GSig}(\boldsymbol{gsk}[i_b], m)$
$d \xleftarrow{\$} A^{\mathsf{Open}(gmsk,\cdot,\cdot)}(\mathsf{guess}, \mathrm{St}, \sigma)$
If $A$ did not query its oracle with $m, \sigma$ in the guess stage then return $d$ EndIf
Return 0

**Fig. 2.** Experiment used to define full-anonymity of a group signature scheme $\mathcal{GS} = (\mathsf{GKg}, \mathsf{GSig}, \mathsf{GVf}, \mathsf{Open})$. Here $A$ is an adversary, $b \in \{0,1\}$, and St denotes state information passed by the adversary between stages.

Correctness. A group signature scheme must satisfy the following correctness requirement: For all $k, n \in \mathbb{N}$, all $(gpk, gmsk, \boldsymbol{gsk}) \in [\mathsf{GKg}(1^k, 1^n)]$, all $i \in [n]$ and all $m \in \{0,1\}^*$

$$\mathsf{GVf}(gpk, m, \mathsf{GSig}(\boldsymbol{gsk}[i], m)) = 1 \text{ and } \mathsf{Open}(gmsk, m, \mathsf{GSig}(\boldsymbol{gsk}[i], m)) = i \ .$$

In [9], the authors identify two security notions which are sufficient for defining security of group signature schemes. Out of the two notions, termed in [9] *full-anonymity* and *full-traceability* respectively we recall the formalization of the first and only informally discuss the second.

Full-anonymity. Informally, anonymity requires that an adversary not in possession of the group manager's secret key find it hard to recover the identity of the signer from its signature. The formalization of [9] uses a strong indistinguishability-based formulation. Roughly an adversary is allowed to interact with the group signature by asking for signatures, and openings of signatures of its own choosing. At the end of this interaction which represents the choose stage, the adversary has to output a message $m$ and two identities $i_0$ and $i_1$. As input to its second stage, the adversary receives state information it had output at the end of the choose stage and a challenge signature on $m$, created using one of the two identities chosen at random. The goal of the adversary is to determine which of the two users created the signature.

The experiment defining full-anonymity is given in Figure 2.3.

The advantage of an adversary $A$ in breaking the full-anonimity of a group signature scheme $\mathcal{GS}$ is denoted by

$$\mathbf{Adv}^{\text{anon}}_{\mathcal{GS},A}(k,n) = \Pr\left[\mathbf{Exp}^{\text{anon-1}}_{\mathcal{GS},A}(k,n) = 1\right] - \Pr\left[\mathbf{Exp}^{\text{anon-0}}_{\mathcal{GS},A}(k,n) = 1\right] .$$

A group signature scheme $\mathcal{GS}$ is said to be *fully-anonymous* if for any polynomial-time adversary $A$, the two-argument function $\mathbf{Adv}^{\text{anon}}_{\mathcal{GS},A}(\cdot,\cdot)$ is negligible (as defined in Section 2.1.)

Full-traceability. Full-traceability refers to the ability of the group manager to revoke anonymity of signers. Informally it requires that no colluding set $S$ of group members, comprised potentially of the whole group, can create signatures

that cannot be traced back to some member of $S$. A formalization of this property appears in [9], and we omit it here since is not relevant to the results of this paper.

The main result of [9] is to show that if trapdoor functions exist then group signature schemes that are fully-anonymous and fully-traceable also exist.

## 3 Group Signature Schemes Imply Public Key Cryptography

In this section, we show how to construct a secure public key encryption scheme given any secure group signature scheme.

### 3.1 Construction

Fix an arbitrary group signature scheme $\mathcal{GS} = (\mathsf{GKg}, \mathsf{GSig}, \mathsf{GVf}, \mathsf{Open})$. The idea of our construction is the following. Consider an instance of $\mathcal{GS}$ in which the group of signers has size 2, i.e. it only contains users 0 and 1. Consider the following encryption scheme, $\mathcal{AE}[\mathcal{GS}]$: the public key consists of the signature verification key of the group $gpk$, together with the signing keys of users 0 and 1, i.e. the vector $\boldsymbol{gsk} = (\boldsymbol{gsk}[0], \boldsymbol{gsk}[1])$. The associated secret key consists of the group verification key together with the group manager secret key. The encryption of message $M = b_0 b_1 \dots b_n$ with $b_i \in \{0, 1\}$ is done bit by bit, where the encryption of the bit $b$ is a signature on some fixed message $\mathbf{0}$ using the group signing key of user $b$. The decryption is immediate: to decrypt the encryption $\sigma$ of a bit $b$, simply verify that $\sigma$ is a valid group signature, and if so use the group manager's secret key to recover the identity of the signer (i.e. $b$). This immediately extends to arbitrary length messages.

We give the full details of our construction in Figure 3.

### 3.2 Security Proof

Let $\mathcal{B}$ be an adversary attacking the $\mathsf{IND\text{-}CPA}$ security of the encryption scheme $\mathcal{AE}[\mathcal{GS}]$. We show how to construct an adversary $\mathcal{A}$ against the group signature scheme $\mathcal{GS}$ such that

$$\mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{AE},\mathcal{A}}(k) \leq p_{\mathcal{A}}(k) \cdot \mathbf{Adv}^{\text{anon}}_{\mathcal{GS},\mathcal{B}}(k, 2) \;, \tag{1}$$

where $p_{\mathcal{A}}(k)$ is some polynomial bounding the running time of adversary $\mathcal{A}$. Since we assumed that $\mathcal{GS}$ is fully-anonymous, the function on the right-hand side of the inequality is negligible so $\mathcal{AE}$ is an $\mathsf{IND\text{-}CPA}$ secure encryption scheme.

The algorithm $\mathcal{A}$ is given in Figure 4. In the $\mathsf{guess}$ stage, $\mathcal{A}$ runs the $\mathsf{guess}$ stage of algorithm $\mathcal{B}$ for encryption scheme $\mathcal{AE}$ and obtains two messages $m_0$ and $m_1$. These messages, together with the state information output by $\mathcal{B}$ is forwarded to the $\mathsf{choose}$ stage of $\mathcal{A}$. In this stage, $\mathcal{A}$ selects at random a position $j$ on which $m_0$ and $m_1$ are different, and creates a challenge ciphertext for $\mathcal{B}$. The challenge ciphertext is an encryption $(gpk, \boldsymbol{gsk})$ of a word which on its first

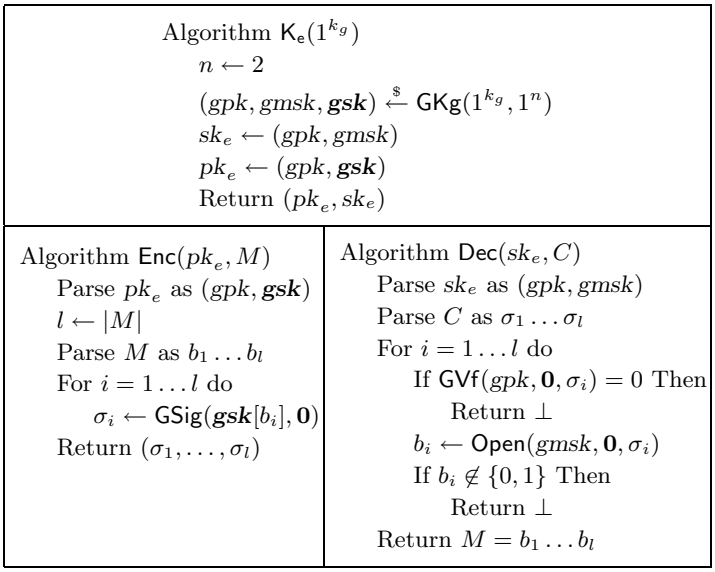


**Fig. 3.** Construction of an IND-CPA secure public-key bit-encryption scheme $\mathcal{AE}[\mathcal{GS}] = (\mathsf{K_e}, \mathsf{Enc}, \mathsf{Dec})$ based on any secure group signature scheme $\mathcal{GS} = (\mathsf{GKg}, \mathsf{GSig}, \mathsf{GVf}, \mathsf{Open})$.

$j-1$ positions coincides with $m_1$ and on its last $n-j$ positions coincides with $m_0$, where $n = |m_0| = |m_1|$. The bit $b$ on position $j$ in the plaintext encrypted by the challenge ciphertext is precisely the identity of the player that generated the challenge signature $\sigma$ which $\mathcal{A}$ received from its environment.

For some fixed messages $m_0$ and $m_1$, let us denote by $s_0, \dots, s_{\mathrm{p}}$ the sequence of $\mathrm{p} = |\mathsf{diff}(m_0, m_1)|$ words such that $s_0 = m_0$, $s_{\mathrm{p}} = m_1$, and any two consecutive words $s_{i-1}$ and $s_i$ differ exactly in one bit position. More precisely, let $j$ be the element of rank $i$ in $\mathsf{diff}(m_0, m_1)$. We can construct word $s_i$ from word $s_{i-1}$ by flipping the $j$-th bit of $s_{i-1}$, for $i = 1, \dots, \mathrm{p}$. Now, let $i$ be the rank of the value $j$ selected by $\mathcal{A}$ during the choose stage of $\mathcal{A}$. Therefore, adversary $\mathcal{B}$ receives as challenge either the encryption of $s_{i-1}$ or the encryption of $s_i$, depending on the key used to create challenge signature $\sigma$. With this in mind, notice that in the experiment $\mathbf{Exp}^{\text{anon-b}}_{\mathcal{GS},\mathcal{A}}(k, 2)$ (for $b \in \{0, 1\}$), adversary $\mathcal{A}$ successfully guesses the bit $b$ whenever adversary $\mathcal{B}$ correctly identifies if the challenge ciphertext is the encryption of $s_{i-1}$ or that of $s_i$. To simplify notation, we will write $\mathcal{B}(\mathsf{Enc}(pk, s_i))$ for $\mathcal{B}(\mathsf{guess}, \mathrm{St}, \mathsf{Enc}((gpk, \boldsymbol{gsk}), s_i))$. It follows from the above discussion that

$$\Pr\left[\,\mathbf{Exp}^{\text{anon-0}}_{\mathcal{GS},A}(k, 2) = 1\,\right] = \frac{1}{|\mathsf{diff}(m_0, m_1)|} \sum_{i=1}^{|\mathsf{diff}(m_0,m_1)|} \Pr\left[\,\mathcal{B}(\mathsf{Enc}(pk, s_{i-1})) = 1\,\right]$$

| | |
|---|---|
| Adversary $\mathcal{A}(\mathsf{choose}, gpk, \boldsymbol{gsk})$<br>$(\mathrm{St}, m_0, m_1) \leftarrow \mathcal{B}(\mathsf{choose}, (gpk, \boldsymbol{gsk}))$<br>$j \leftarrow \mathsf{diff}(m_0, m_1)$<br>$\mathrm{St}' \leftarrow (\mathrm{St}, m_0, m_1, gpk, \boldsymbol{gsk}, j)$<br>Return $(\mathrm{St}', m_0[j], m_1[j], \mathbf{0})$ | Adversary $\mathcal{A}(\mathsf{guess}, \mathrm{St}', \sigma)$<br>Parse $\mathrm{St}'$ as $(\mathrm{St}, m_0, m_1, gpk, \boldsymbol{gsk}, j)$<br>For $i \leftarrow 1, \ldots, j-1$<br>$\sigma_i \leftarrow \mathsf{GSig}(\boldsymbol{gsk}[m_0[i]], \mathbf{0})$<br>For $i \leftarrow j+1, \ldots, n$<br>$\sigma_i \leftarrow \mathsf{GSig}(\boldsymbol{gsk}[m_1[i]], \mathbf{0})$<br>$\sigma_j \leftarrow \sigma$<br>Let $d \leftarrow \mathcal{B}(\mathsf{guess}, \mathrm{St}, (\sigma_1, \ldots, \sigma_l))$<br>Output $d$ |

**Fig. 4.** Construction of an adversary $\mathcal{A}$ against $\mathcal{GS}$ from an adversary $\mathcal{B}$ against $\mathcal{AE}[\mathcal{GS}]$.

and

$$\Pr\left[\,\mathbf{Exp}^{\text{anon-1}}_{\mathcal{GS},A}(k,2)=1\,\right] = \frac{1}{|\mathsf{diff}(m_0,m_1)|}\sum_{i=1}^{|\mathsf{diff}(m_0,m_1)|}\Pr\left[\,\mathcal{B}(\mathsf{Enc}(pk,s_i))=1\,\right],$$

where the first factor represents the probability that the value $j$ selected by $\mathcal{A}$ has rank $i$. Let $\mathrm{p} = |\mathsf{diff}(m_0, m_1)|$. We can now bound the advantage of $\mathcal{A}$ by:

$$\begin{aligned}
&\mathbf{Adv}^{\text{anon}}_{\mathcal{GS},\mathcal{A}}(k,2)\\
&\quad= \Pr\left[\,\mathbf{Exp}^{\text{anon-1}}_{\mathcal{GS},\mathcal{A}}(k,2)=1\,\right] - \Pr\left[\,\mathbf{Exp}^{\text{anon-0}}_{\mathcal{GS},\mathcal{A}}(k,2)=1\,\right]\\
&\quad= \frac{1}{\mathrm{p}}\cdot\sum_{i=1}^{\mathrm{p}}\Pr\left[\,\mathcal{B}(\mathsf{Enc}(pk,s_i))=1\,\right] - \frac{1}{\mathrm{p}}\cdot\sum_{i=1}^{\mathrm{p}}\Pr\left[\,\mathcal{B}(\mathsf{Enc}(pk,s_{i-1}))=1\,\right]\\
&\quad= \frac{1}{\mathrm{p}}\cdot\sum_{i=1}^{\mathrm{p}}\left(\Pr\left[\,\mathcal{B}(\mathsf{Enc}(pk,s_i))=1\,\right] - \Pr\left[\,\mathcal{B}(\mathsf{Enc}(pk,s_{i-1}))=1\,\right]\right)\\
&\quad= \frac{1}{\mathrm{p}}\cdot\left(\Pr\left[\,\mathcal{B}(\mathsf{Enc}(pk,s_{\mathrm{p}}))=1\,\right] - \Pr\left[\,\mathcal{B}(\mathsf{Enc}(pk,s_0))=1\,\right]\right)\\
&\quad= \frac{1}{\mathrm{p}}\cdot\left(\Pr\left[\,\mathcal{B}(\mathsf{Enc}(pk,m_1))=1\,\right] - \Pr\left[\,\mathcal{B}(\mathsf{Enc}(pk,m_0))=1\,\right]\right)\\
&\quad= \frac{1}{\mathrm{p}}\cdot\mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{AE},\mathcal{B}}(k)\\
&\quad\geq \frac{1}{|m_0|}\cdot\mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{AE},\mathcal{B}}(k)
\end{aligned}$$

We can also bound the length of $m_0$ by the total running of algorithm $\mathcal{A}$, which is some polynomial $p_{\mathcal{A}}(\cdot)$ in the security parameter. As a result,

$$\mathbf{Adv}^{\text{anon}}_{\mathcal{GS},\mathcal{A}}(k,2) \geq \frac{1}{p_{\mathcal{A}}(k)}\cdot\mathbf{Adv}^{\text{ind-cpa}}_{\mathcal{AE},\mathcal{B}}(k)$$

which gives the result claimed in Equation 1 by rearranging the terms.

*Remark 1.* The encryption scheme $\mathcal{AE}[\mathcal{GS}]$ in Figure 3 can also be proven to be IND-CCA secure if we restrict the length of the messages being encrypted to 1 (i.e., the plaintext is just a *single* bit). Note that, in this special case, we can easily simulate the decryption oracle given to the adversary $\mathcal{B}$ using the oracle for the *opening* algorithm Open from the experiment for anonymity.

*Remark 2.* In [11], Boneh, Boyen, and Shacham define a weaker variant of the full-anonymity property, called CPA-full-anonymity, in which the Open oracle is not given to the adversary in the experiment for anonymity. Since the proof that secure group signatures imply IND-CPA public-key encryption does not rely on the Open oracle, the implication still stands even in their weaker security model.

## 4 Concluding Remarks

The main advantage of proving that the existence of secure group signature schemes implies public-key encryption schemes is that one can apply several of the results that are known for public-key encryption to the case of group signatures. Here we highlight the most important ones.

Group signatures from one-way functions. Given that standard signature schemes can be constructed from any one-way function, one may wonder whether the same is true for group signatures. Unfortunately, this does not seem to be the case. In particular, such construction would need to use non-black-box reduction techniques when proving its security [25, 29]. Loosely speaking, a non-black-box reduction from a cryptographic scheme to a given primitive is a reduction in which either the code of the primitive is used in a non-black-box manner by the cryptographic scheme or the code of the adversary against the cryptographic scheme is explicitly used when building an adversary against the primitive.

As pointed out in [29], many of the examples of cryptographic schemes that make use of the primitive's code come from constructions making use of the general construction of zero-knowledge proofs for NP languages of Goldreich et al. [20, 19], as their construction is non-black-box. However, it was recently found [4, 6, 5] that reductions making use of the adversary's code in the proof of security were found and they are considered one of main breakthroughs in the area of zero-knowledge. Nevertheless, we would like to stress that almost all reductions in cryptography are black-box and the examples of non-black-box reductions are very few. Hence, it is unlikely that group signatures can be built from one-way functions.

On the minimal assumption for group signatures. Despite the difficulty of constructing group signature schemes from one-way functions, one may wonder whether it is possible to build group signature from apparently stronger assumptions such as trapdoor predicates or (poly-to-one) trapdoor functions. A poly-to-one trapdoor function is a trapdoor function where the number of pre-images for any point in the range is polynomially-bounded. However, the picture

in this case is not so clear and such constructions may or may not be possible. For this reason, we review some results which may be of importance to us.

The first of these results is the one of Bellare et al. [7] that shows the the restriction on the pre-image size of trapdoor functions is an important one since super-poly-to-one trapdoor functions can be constructed from one-way functions [7]. On the other hand, poly-to-one trapdoor functions are also known to imply trapdoor predicates [7, 31], which in turn are known to be equivalent to to secure public-key encryption [21].

Another relevant result is the one due to Gertner et al. [18] which shows that no black-box reductions exist from trapdoor predicates to poly-to-one trapdoor functions. In fact, their result shows that it might be possible to construct trapdoor predicates (i.e., public-key encryption) based on assumptions that are strictly weaker than (poly-to-one) trapdoor functions, with respect to black-box reductions.

Another separation that is important to our work is the one from Gertner et al. [17] which shows that there are no black-box constructions of trapdoor permutations from trapdoor functions. Their result seems to indicate that the latter assumption is stronger than (poly-to-one) trapdoor functions.

Apart from the fact that trapdoor permutations imply group signatures [9] and that the latter implies trapdoor predicates (this paper), the impossibility of black-box reductions from trapdoor predicates to trapdoor functions to trapdoor permutations leaves completely open the remaining relations between these primitives and group signatures. Therefore, constructions of group signatures based on trapdoor functions or trapdoor predicates may still be possible. Turning to the other side of the coin, the construction of any of these primitives from group signatures may also be possible.

## Acknowledgments

The first author has been supported in part by the European Commission through the IST program under the IST-2002-507932 ECRYPT contract and in part by a CNRS postdoctoral grant. The second author was supported by the NSF Career Award CCR-0208800.

## References

1. G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik. A practical and provably secure coalition-resistant group signature scheme. In M. Bellare, editor, *Advances in Cryptology – CRYPTO 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 255–270, Santa Barbara, CA, USA, Aug. 20–24, 2000. Springer-Verlag, Berlin, Germany.
2. G. Ateniese and G. Tsudik. Group signatures la carte. In *10th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 848–849, Baltimore, Maryland, USA, Jan. 17–19, 1999. ACM-SIAM.

3. G. Ateniese and G. Tsudik. Some open issues and new directions in group signatures. In M. Franklin, editor, *Financial Cryptography'99*, volume 1648 of *Lecture Notes in Computer Science*, pages 196–211, Anguilla, British West Indies, Feb. 1999. Springer-Verlag, Berlin, Germany.
4. B. Barak. How to go beyond the black-box simulation barrier. In *42nd Annual Symposium on Foundations of Computer Science*, pages 106–115, Las Vegas, Nevada, USA, Oct. 14–17, 2001. IEEE Computer Society Press.
5. B. Barak. Constant-round coin-tossing with a man in the middle or realizing the shared random string model. In *43rd Annual Symposium on Foundations of Computer Science*, pages 345–355, Vancouver, British Columbia, Canada, Nov. 16–19, 2002. IEEE Computer Society Press.
6. B. Barak, O. Goldreich, S. Goldwasser, and Y. Lindell. Resettably-sound zero-knowledge and its applications. In *42nd Annual Symposium on Foundations of Computer Science*, pages 116–125, Las Vegas, Nevada, USA, Oct. 14–17, 2001. IEEE Computer Society Press.
7. M. Bellare, S. Halevi, A. Sahai, and S. P. Vadhan. Many-to-one trapdoor functions and their ralation to public-key cryptosystems. In H. Krawczyk, editor, *Advances in Cryptology – CRYPTO'98*, volume 1462 of *Lecture Notes in Computer Science*, pages 283–298, Santa Barbara, CA, USA, Aug. 23–27, 1998. Springer-Verlag, Berlin, Germany.
8. M. Bellare and S. Micali. How to sign given any trapdoor function. *Journal of the ACM*, 39(1):214–233, 1992.
9. M. Bellare, D. Micciancio, and B. Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In E. Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 614–629, Warsaw, Poland, May 4–8, 2003. Springer-Verlag, Berlin, Germany.
10. M. Bellare, H. Shi, and C. Zhang. Foundations of group signatures: The case of dynamic groups. Cryptology ePrint Archive, Report 2004/077, 2004. `http://eprint.iacr.org/`.
11. D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In M. Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 41–55, Santa Barbara, CA, USA, Aug. 15–19, 2004. Springer-Verlag, Berlin, Germany.
12. E. Bresson and J. Stern. Efficient revocation in group signatures. In K. Kim, editor, *PKC 2001: 4th International Workshop on Theory and Practice in Public Key Cryptography*, volume 1992 of *Lecture Notes in Computer Science*, pages 190–206, Cheju Island, South Korea, Feb. 13–15, 2001. Springer-Verlag, Berlin, Germany.
13. J. Camenisch. Efficient and generalized group signatures. In W. Fumy, editor, *Advances in Cryptology – EUROCRYPT'97*, volume 1233 of *Lecture Notes in Computer Science*, pages 465–479, Konstanz, Germany, May 11–15, 1997. Springer-Verlag, Berlin, Germany.
14. D. Chaum and E. van Heyst. Group signatures. In D. W. Davies, editor, *Advances in Cryptology – EUROCRYPT'91*, volume 547 of *Lecture Notes in Computer Science*, pages 257–265, Brighton, UK, Apr. 8–11, 1991. Springer-Verlag, Berlin, Germany.
15. L. Chen and T. P. Pedersen. New group signature schemes. In A. D. Santis, editor, *Advances in Cryptology – EUROCRYPT'94*, volume 950 of *Lecture Notes in Computer Science*, pages 171–181, Perugia, Italy, May 9–12, 1994. Springer-Verlag, Berlin, Germany.

16. W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22:644–654, 1978.
17. Y. Gertner, S. Kannan, T. Malkin, O. Reingold, and M. Viswanathan. The relationship between public key encryption and oblivious transfer. In *41st Annual Symposium on Foundations of Computer Science*, pages 325–335, Las Vegas, Nevada, USA, Nov. 12–14, 2000. IEEE Computer Society Press.
18. Y. Gertner, T. Malkin, and O. Reingold. On the impossibility of basing trapdoor functions on trapdoor predicates. In *42nd Annual Symposium on Foundations of Computer Science*, pages 126–135, Las Vegas, Nevada, USA, Oct. 14–17, 2001. IEEE Computer Society Press.
19. O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity and a methodology of cryptographic protocol design. In *27th Annual Symposium on Foundations of Computer Science*, pages 174–187, Toronto, Ontario, Canada, Oct. 27–29, 1986. IEEE Computer Society Press.
20. O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *Journal of the ACM*, 38(3):691–729, 1991.
21. S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28:270–299, 1984.
22. S. Goldwasser, S. Micali, and R. L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing*, 17(2):281–308, Apr. 1988.
23. S. Halevi and H. Krawczyk. Public-key cryptography and password protocols. *ACM Transactions on Information and System Security*, 2(3):230–268, Aug. 1999.
24. R. Impagliazzo and M. Luby. One-way functions are essential for complexity-based cryptography. In *30th Annual Symposium on Foundations of Computer Science*, pages 230–235, Research Triangle Park, North Carolina, Oct. 30 – Nov. 1, 1989. IEEE Computer Society Press.
25. R. Impagliazzo and S. Rudich. Limits on the provable consequences of one-way permutations. In *21st Annual ACM Symposium on Theory of Computing*, pages 44–61, Seattle, Washington, USA, May 15–17, 1989. ACM Press.
26. A. Kiayias and M. Yung. Extracting group signatures from traitor tracing schemes. In E. Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 630–648, Warsaw, Poland, May 4–8, 2003. Springer-Verlag, Berlin, Germany.
27. A. Kiayias and M. Yung. Group signatures: Provable security, efficient constructions and anonymity from trapdoor-holders. Cryptology ePrint Archive, Report 2004/076, 2004. `http://eprint.iacr.org/`.
28. M. Naor and M. Yung. Universal one-way hash functions and their cryptographic applications. In *21st Annual ACM Symposium on Theory of Computing*, pages 33–43, Seattle, Washington, USA, May 15–17, 1989. ACM Press.
29. O. Reingold, L. Trevisan, and S. P. Vadhan. Notions of reducibility between cryptographic primitives. In M. Naor, editor, *TCC 2004: 1st Theory of Cryptography Conference*, volume 2951 of *Lecture Notes in Computer Science*, pages 1–20, Cambridge, MA, USA, Feb. 19–21, 2004. Springer-Verlag, Berlin, Germany.
30. J. Rompel. One-way functions are necessary and sufficient for secure signatures. In *22nd Annual ACM Symposium on Theory of Computing*, pages 387–394, Baltimore, Maryland, USA, May 14–16, 1990. ACM Press.
31. A. C. Yao. Theory and applications of trapdoor functions. In *23rd Annual Symposium on Foundations of Computer Science*, pages 80–91, Chicago, Illinois, Nov. 3–5, 1982. IEEE Computer Society Press.