The Fiat–Shamir Transform for Group and Ring Signature Schemes

M.-F. Lee, N.P. Smart, and B. Warinschi

Dept. Computer Science,
University of Bristol,
Merchant Venturers Building,
Woodland Road,
Bristol, BS8 1UB,
United Kingdom.
{lee,nigel,bogdan}@cs.bris.ac.uk

Abstract. The Fiat-Shamir (FS) transform is a popular tool to produce particularly efficient digital signature schemes out of identification protocols. It is known that the resulting signature scheme is secure (in the random oracle model) if and only if the identification protocol is secure against passive impersonators. A similar results holds for constructing ID-based signature schemes out of ID-based identification protocols.
The transformation had also been applied to identification protocols with additional privacy properties. So, via the FS transform, ad-hoc group identification schemes yield ring signatures and identity escrow schemes yield group signature schemes. Unfortunately, results akin to those above are not known to hold for these latter settings and the security of the resulting schemes needs to be proved from scratch, or worse, it is often simply assumed. Therefore, the security of the schemes obtained this way does not clearly follow from that of the base identification protocol and needs to be proved from scratch. Even worse, some papers seem to simply assume that the transformation works without proof.
In this paper we provide the missing foundations for the use of the FS transform in these more complex settings. We start with defining a formal security model for identity escrow schemes (a concept proposed earlier but never rigorously formalized). Our main result constists of necessary and sufficient conditions for an identity escrow scheme to yield (via the FS transform) a secure group signature schemes. In addition, we discuss several variants of this result that account for the constructions of group signatures that fulfill weaker notions of security. In addition, using the similarity between group and ring signature schemes we give analogous results for the latter primitive.

1 Introduction

Background. A canonical identification scheme is a three-move two-party protocol: the prover first sends a commitment Cmt to the verifier, the verifier picks and returns a random string Ch as a challenge. After receiving the challenge,

the prover outputs a response Rsp which is derived from the commitment, the challenge, and the secret of the prover. The verifier checks that the resulting transcript $(\textsc{Cmt}, \textsc{Ch}, \textsc{Rsp})$ satisfies a certain property, in which case we say that the transcript is accepting and the verifier outputs one, otherwise the verifier outputs zero. The Fiat-Shamir transform [19] takes as input a canonical identification protocol and produces a digital signature scheme. The transform essentially removes the interaction in such protocols, and in doing so it involves an arbitrary message M. This results in the following signing algorithm. To sign a message M the signer computes the commitment Cmt as the prover does in the identification scheme, then hashes Cmt and the message M using a hash function H to obtain a challenge $\textsc{Ch} = H(\textsc{Cmt}||M)$. The signer finally computes a response Rsp according to the underlying identification protocol. The resulting signature is $(\textsc{Cmt}, \textsc{Rsp})$. To verify the signature, one recomputes Ch as $H(\textsc{Cmt}||M)$ and verifies that the transcript $(\textsc{Cmt}, \textsc{Ch}, \textsc{Rsp})$ is an accepting identification transcript.

The transform is particularly popular since it yields some of the most efficient digital signature schemes known to date. Unsurprisingly, the transformation had been extensively studied. There are negative results that explain the difficulty of instantiating the hash function used in the transformation in a way that ensures the security of signature scheme in the standard model [17, 21]. Also, there are positive results relating the security of the underlying identification protocol to that of the resulting signature scheme in the random oracle model [1, 32, 33]. The best known such result is due to Abdalla et al. [1] who prove that the resulting signature scheme is secure in the random oracle model, *if and only if* the starting identification protocol is secure against passive impersonators. An important consequence of such general results is that they entail modular security proofs for signature schemes. First, prove the security of the identification protocol (this step is sometimes quite simple – for example it may immediately follow from existing known result, e.g. the identification protocol being honest-verifier zero-knowledge). Then conclude, via the general result, the security of the signature scheme. This path was advocated and used by Bellare, Namprempre, and Neven in a later paper where they prove (among other results) the soundness of the FS transform when applied to ID-based identification schemes to obtain ID-based signature schemes [7].

The Fiat-Shamir transform had been used as a design tool in other contexts where canonical three-move identification protocol occur. Notable examples include the construction of group signature schemes out of group identification schemes and the construction of ring signature schemes out of ad-hoc group identification. Unfortunately, unlike for digital signatures [1] no results formally relate the security of the underlying group/ad-hoc group identification scheme with that of the resulting group/ring signature scheme. In this cases, *e.g.* [2–5, 12–15], the security of the signature scheme needs to be proved from scratch. Unfortunately, it is simply assumed that the transformation "works". In this paper we investigate the use of the transform in the construction of group and ring signature schemes. We detail our results next.

Our results. We start by formalizing the notion of group identification (or identity-escrow). The primitive had been proposed by Kilian and Petrank [23] but its security had never been rigorously defined. Recall that an identity escrow scheme allows users to (interactively) prove membership in a group in a way that hides their individual identities. In case of misuse however, anonymity can be revoked by a group opener who is in possession of a secret opening key. Such schemes are therefore the interactive counterpart of group signature schemes. We take advantage of progress on security models for group signatures [6, 9] and adapt existing security notions for this primitive to group identification. Our models consider the case of monotonic dynamic groups (where users can be added to the group by a group manager). We define three distinct security notions. Two notions refer to adversaries that attempt to impersonate group members and here we distinguish between impersonators that frame other honest group members, and impersonators who produce transcripts that cannot be traced. By analogy with the corresponding notions in group signatures the resulting notions are *non-frameability* and *traceability*. A third requirement, *anonymity* demands that executing the identification protocols hides the user identity: an adversary is not able to tell apart runs of the identification protocol of different users. Finally, we also formalize as a game the correctness of group identification schemes. We give the details of the models for the case of passive adversaries (adversaries that only observe executions of the identification protocols of honest parties, but are not allowed to interact with them). Furthermore, the models are for canonical identification schemes (schemes where the identification protocol has the three-move structure outlined at the beginning of the introduction). We make these restrictions for simplicity: our theorems are only for canonical identification protocols and relate the security of the resulting group signature scheme with that of the underlying identification protocol under passive attacks. Nevertheless, the extension of our definitions to arbitrary group identification schemes and active adversaries is immediate.

Our main result is that the group signature obtained via the FS transform from a canonical identity-escrow scheme is correct, anonymous, traceable, and non-frameable if and only if the underlying group identification is, respectively, correct, anonymous, traceable, and non-frameable under passive (i.e., eavesdropping only) attacks.

Our theorem yields group signature schemes that meet the strongest possible notion of security. However, the literature for group signatures contains a large number of variations on this security model. The reason is that weaker, but still quite reasonable security requirements often allow for significantly more efficient schemes. Examples of restrictions include considering static groups, merging the group opener and the group manager, or disallowing the adversary to ask openings of arbitrary transcripts. These weaker notions are usually obtained by simply imposing restrictions on how the adversary interacts with the oracles defining the security game. In the reduction that proves our main theorem we show how to build an adversary against the underlying group identification scheme out of an adversary for the resulting group signature scheme. In this reduction, the

restrictions that the former adversary has on using his oracles translate into similar restrictions on how the latter adversary is allowed to use his own oracles. We therefore obtain analogue versions of our result that relates correspondingly weaker notions of group identification and group signatures by observing how the restrictions between the two settings translate through our reduction.

We take advantage of the similarities between group and ring identification/signature schemes and extend our results to this latter primitive. In this extended abstract we give the theorem that we prove but leave the details for the full version.

Finally, we also investigate the use of the FS transform to obtain ring signature schemes out of ad-hoc group identification schemes. In an ad-hoc group identification scheme a user proves (interactively) that he has the secret key that corresponds to one of several public keys selected by the user. The group is thus chosen ad-hoc. It is desired that the privacy of the user is preserved (and notice that in this case there is no possibility of opening an identification transcript as there is not protocol through which users are added to a group.) The non-interactive version of ad-hoc group identification are thus ring signatures. From this brief description, it is clear that the security models for these two primitive are close to those for group signature and group identification, respectively. Indeed, the difference is that there is no need for a group manager (as the group is decided by the user that proves membership) and there is no opening manager (anonymity here is not revocable). Our main result then applies with little modification to the construction of ring signatures out of ad-hoc group identification schemes via the Fiat-Shamir transform.

NOTATION We end this introduction by covering some basic notation which will be used throughout this paper. If S is a set then $s \leftarrow S$ means that s is selected uniformly at random from S. Let $\mathcal{A}(\cdot, \cdots, \cdot; R)$ be a randomized algorithm with coins R, then $y \leftarrow \mathcal{A}(x_1, \ldots, x_n R)$ means on input of $x_1, \ldots, x_n$ and coins R, the value y is the unique output of the algorithm. The notation $y \leftarrow \mathcal{A}(x_1, \ldots, x_n)$ is shorthand for first selecting a random R and then setting $y \leftarrow \mathcal{A}(x_1, \ldots, x_n; R)$. We let $\mathsf{Coins}(\mathcal{A})$ denote the space which R is drown from for the algorithm $\mathcal{A}$. An algorithm $\mathcal{A}$ run on input $x_1, \ldots, x_n$ with access to oracles $\mathcal{O}_1, \ldots, \mathcal{O}_m$ will be denoted by $\mathcal{A}(x_1, \ldots, x_n : \mathcal{O}_1, \ldots, \mathcal{O}_m)$, so as to avoid too many superscripts and subscripts.

2 Group Identification Schemes

In this section we formalize group identification schemes. These schemes were introduced by Kilian and Petrank [23] under the name identity escrow schemes. We use these two names interchangeably.

SYNTAX. Group identification schemes allow a user to prove membership in a group in such a way that his personal identity is protected. Using special secret keys, a group manager can add users whereas a group opener can revoke anonymity of any identification transcript. Since group identification schemes are the interactive counterparts of group signature schemes we make use of progress

in the formalization of the latter concept. In particular, we follow the model proposed by Bellare, Shi, and Zhang [9].

A group identification scheme is given by the tuple of algorithms $\mathcal{GID} = (\mathsf{GKg}^{\mathcal{GID}}, \mathsf{UKg}^{\mathcal{GID}}, \mathsf{Join}^{\mathcal{GID}}, \mathsf{Iss}^{\mathcal{GID}}, (\mathsf{P}^{\mathcal{GID}}, \mathsf{V}^{\mathcal{GID}}), \mathsf{Open}^{\mathcal{GID}}, \mathsf{Judge}^{\mathcal{GID}})$, where the functionality of these algorithms is as follows:

- $\mathsf{GKg}^{\mathcal{GID}}$: A setup program running a probabilistic key generation algorithm. It takes a security parameter 1^k and outputs the secret-public key pair $(\mathfrak{gmsk}, \mathfrak{gmpk})$ for the group manager $\mathcal{M}$, and a secret key $\mathfrak{osk}$ for the opener $\mathcal{O}p$. The key $\mathfrak{gmpk}$ is the public key for the group.
- $\mathsf{UKg}^{\mathcal{GID}}$: This is a probabilistic algorithm to generate user public/private key pairs. When run by user i, on input of 1^k, this outputs a user's key pair $(\mathfrak{usk}_i, \mathfrak{upk}_i)$.
- $(\mathsf{Join}^{\mathcal{GID}}, \mathsf{Iss}^{\mathcal{GID}})$: This is an interactive protocol between a new group member i and the group manager $\mathcal{M}$. Each of the algorithms take as input a state and produce a new state plus a decision $\{\mathsf{accept}, \mathsf{reject}, \mathsf{cont}\}$. The initial state of $\mathsf{Join}^{\mathcal{GID}}$ is the private key of the user $\mathfrak{usk}_i$, whilst that of $\mathsf{Iss}^{\mathcal{GID}}$ is $\mathfrak{gmsk}$ and the public key of the user. If the issuer group manager (running $\mathsf{Iss}^{\mathcal{GID}}$) accepts then the final output is assigned to $\mathfrak{Inf}_i$ (where i is the index/identity of the user). This is information that is to be passed to the group opener (who will later use it to open transcripts produced by user i). If the user i accepts then the final state of $\mathsf{Join}^{\mathcal{GID}}$ is assigned to $\mathfrak{gsk}_i$.
- $(\mathsf{P}^{\mathcal{GID}}, \mathsf{V}^{\mathcal{GID}})$: An interactive protocol between a prover and a verifier. The prover's input a value $\mathfrak{gsk}_i$, whereas the verifier's input is $\mathfrak{gmpk}$.
- $\mathsf{Open}^{\mathcal{GID}}$: A deterministic algorithm, on input of a transcript $\mathcal{T}$ of the $(\mathsf{P}^{\mathcal{GID}}, \mathsf{V}^{\mathcal{GID}})$ protocol, the values $\mathfrak{Inf}_*$ and the opening key $\mathfrak{osk}$. The algorithm outputs a pair (i, τ), where $i \geq 0$. If $i = 0$ then the algorithm is claiming that no group member was authenticated using the transcript $\mathcal{T}$, when $i \geq 1$ the algorithm is claiming that the group member with identity i was the prover in the transcript $\mathcal{T}$. In the latter case the value τ is a proof of this claim.
- $\mathsf{Judge}^{\mathcal{GID}}$: This algorithm takes as input $\mathfrak{gmpk}$, an integer j, the public key $\mathfrak{upk}_j$, a transcript $\mathcal{T}$ and a proof τ. It's goal is to check whether τ is a proof that j produced $\mathcal{T}$.

The above syntax is for general group identification scheme. Our results are for a special class of such schemes which we call (following [1]) *canonical*. For ease of exposition we give the security definition for group identification schemes for these class of schemes. The extension to general group identification is immediate.

Canonical Group Identification Scheme. A canonical group identification scheme is a group identification scheme as above, except that now the $(\mathsf{P}^{\mathcal{GID}}, \mathsf{V}^{\mathcal{GID}})$ protocol is given by a three-move protocol of the commit-challenge-response variety. In the first move the prover sends a commitment Cmt to verifier, the verifier then responds with a random string $\mathrm{Ch} \in \{0,1\}^c$ as the challenge. Then the prover outputs a response Rsp which is derived from the commitment Cmt, the challenge Ch and their key $\mathfrak{gsk}_i$. Finally, the verifier verifies

the response and outputs a final decision to decide whether i is in the authorized group. In this case a transcript of the execution is given by $\mathcal{T} = (\textsc{Cmt}, \textsc{Ch}, \textsc{Rsp})$. The verifier algorithm is then of the simplified form $\mathsf{V}^{\mathcal{GID}}(\mathcal{T})$ and it returns a single value in $\{0, 1\}$.

Security of Canonical Group Identification Scheme. Following the treatment of [9] for group signatures we present notions of security, which we call anonymity, traceability and non-frameability for canonical group identification schemes. All our security models are for *passive* adversaries: while the adversary can obtain transcripts of the identification protocol run by honest users, he cannot directly interact with these users playing the role of the verifier. As explainer earlier, we focus on this setting since our theorems require security in this weaker sense. The extension of the definitions to active adversaries who can also interact with honest users is immediate.

Our definition use a set of oracles which we define in Figure 1. All oracles (and the underlying experiments) maintain the following global variables: a set HU of honest users, a set CU of corrupted users and a set TL of transcripts, all of which are assumed to be initially empty. Figure 1 shows what and how these oracles work in detail. Informally, the adversarial abilities that these oracles model are as follows.

- $\mathsf{AddU}(i)$: The adversary can use this oracle to add an honest user i to the group.
- $\mathsf{CrptU}(i, \mathfrak{pk})$: The adversary can create a corrupt user i and set the users public key to $\mathfrak{pk}$.
- $\mathsf{SndToI}(i, M)$: The adversary can use this oracle to engage as a corrupt user in a group-join protocol with the honest, Iss-executing issuer.
- $\mathsf{SndToU}(i, M)$: This oracle models the situation that the adversary has corrupted the issuer. The adversary can use this oracle to engage in the group-join protocol with the honest, Join-executing user.
- $\mathsf{USK}(i)$: The adversary can call this oracle and obtain both the private secret key and group signing key of an honest user i.
- $\mathsf{Exec}(i)$: This oracle allows the adversary to obtain transcripts of runs of the identification protocol between the honest prover i and an honest verifier.
- $\mathsf{CH}_b(i_0, i_1)$: This oracle is a left-right oracle for defining anonymity. The adversary sends a couple of honest identities (i_0, i_1) to the oracle and gets back a transcript $\mathcal{T}$ of the identification protocol executed by user i_b.
- $\mathsf{Open}(\mathcal{T})$: The adversary can query this oracle to obtain the output of the opening algorithm on $\mathcal{T}$, as long as $\mathcal{T}$ was not returned as a response to the CH_b oracle.

Using these oracles we can now define our security and correctness notions for canonical group identification scheme. We note that we only require security under passive attacks for our application, i.e. the attacker can obtain valid transcripts, but is not able to interact with individual provers. Hence, security is defined for this restricted notion of attack, the generalisation to active attacks is obvious. We also assume that the adversary is not able to read or write the

$\mathsf{AddU}(i)$:
- If $i \in \mathsf{HU} \cup \mathsf{CU}$ then return $\perp$.
- $\mathsf{HU} \leftarrow \mathsf{HU} \cup \{i\}$.
- $(\mathfrak{sk}_i, \mathfrak{pk}_i) \leftarrow \mathsf{UKg}(1^k)$.
- $\mathsf{dec}^i \leftarrow \mathsf{cont}$, $\mathfrak{gsk}_i \leftarrow \perp$.
- $\mathsf{St}_J^i \leftarrow (\mathfrak{gmpk}, \mathfrak{pk}_i, \mathfrak{sk}_i)$.
- $\mathsf{St}_I^i \leftarrow (\mathfrak{gmpk}, \mathfrak{gmsk}, \mathfrak{pk}_i)$, $M_J \leftarrow \perp$.
- $(\mathsf{St}_J^i, M_I, \mathsf{dec}^i) \leftarrow \mathsf{Join}^{\mathcal{GID}}(\S_J^i, M_J)$.
- While $(\mathsf{dec}^i = \mathsf{cont})$ do
  - $(\mathsf{St}_I^i, M_J, \mathsf{dec}^i) \leftarrow \mathsf{Iss}^{\mathcal{GID}}(\mathsf{St}_I^i, M_I, \mathsf{dec}^i)$.
  - If $\mathsf{dec}^i = \mathsf{accept}$ then $\mathfrak{Inf}_i \leftarrow \mathsf{St}_I^i$.
  - $(\mathsf{St}_J^i, M_I, \mathsf{dec}^i) \leftarrow \mathsf{Join}^{\mathcal{GID}}(\mathsf{St}_J^i, M_J)$.
- $\mathfrak{gsk}_i \leftarrow \mathsf{St}_J$.
- Return $\mathfrak{sk}_i$.

$\mathsf{SndToI}(i, M)$:
- If $i \notin \mathsf{CU}$ then return $\perp$.
- $(\mathsf{St}_I^i, M', \mathsf{dec}^i) \leftarrow \mathsf{Iss}^{\mathcal{GID}}(\mathsf{St}_I^i, M, \mathsf{dec}^i)$.
- If $\mathsf{dec}^i = \mathsf{accept}$ then $\mathfrak{Inf}_i \leftarrow \mathsf{St}_I^i$.
- Return M'.

$\mathsf{SndToU}(i, M)$:
- If $i \notin \mathsf{HU}$ then
  - $\mathsf{HU} \leftarrow \mathsf{HU} \cup \{i\}$.
  - $(\mathfrak{sk}_i, \mathfrak{pk}_i) \leftarrow \mathsf{UKg}(1^k)$.
  - $\mathfrak{gsk}_i \leftarrow \perp$, $M \leftarrow \perp$.
  - $\mathsf{St}_J^i \leftarrow (\mathfrak{gmpk}, \mathfrak{pk}_i, \mathfrak{sk}_i)$.
- $(\mathsf{St}_J^i, M', \mathsf{dec}^i) \leftarrow \mathsf{Join}^{\mathcal{GID}}(\mathsf{St}_J^i, M)$
- if $\mathsf{dec}^i = \mathsf{accept}$ then $\mathfrak{gsk}_i \leftarrow \mathsf{St}_J^i$.
- Return (M', dec^i).

$\mathsf{CH}_b(i_0, i_1)$:
- If $i_0 \notin \mathsf{HU}$ or $\mathfrak{gsk}_{i_0} = \perp$ then return $\perp$.
- If $i_1 \notin \mathsf{HU}$ or $\mathfrak{gsk}_{i_1} = \perp$ then return $\perp$.
- $\mathcal{T} \leftarrow \mathsf{Exec}(i_b)$.
- $\mathsf{TL} \leftarrow \mathsf{TL} \cup \{\mathcal{T}\}$.
- Return $\mathcal{T}$.

$\mathsf{CrptU}(i, \mathfrak{pk})$:
- If $i \in \mathsf{HU} \cup \mathsf{CU}$ then return $\perp$.
- $\mathsf{CU} \leftarrow \mathsf{CU} \cup \{i\}$.
- $\mathfrak{pk}_i \leftarrow \mathfrak{pk}$.
- $\mathsf{dec}^i \leftarrow \mathsf{cont}$
- $\mathsf{St}_I^i \leftarrow (\mathfrak{gmpk}, \mathfrak{gmsk}, \mathfrak{pk}_i)$.
- Return 1.

$\mathsf{USK}(i)$:
- If $i \notin \mathsf{HU}$ then return $\perp$.
- Return $(\mathfrak{gsk}_i, \mathfrak{sk}_i)$.

$\mathsf{Open}(\mathcal{T})$:
- If $\mathcal{T} \in \mathsf{TL}$ then return $\perp$
- Return $\mathsf{Open}^{\mathcal{GID}}(\mathcal{T}, \mathfrak{osk}, \mathfrak{Inf}_*)$.

$\mathsf{Exec}(i)$:
- If $i \notin \mathsf{HU}$ or $\mathfrak{gsk}_i = \perp$ then return $\perp$.
- $R \leftarrow \mathsf{Coins}(\mathsf{P}^{\mathcal{GID}})$.
- $\textsc{Cmt} \leftarrow \mathsf{P}^{\mathcal{GID}}(\mathfrak{gsk}_i; R)$.
- $\textsc{Ch} \leftarrow \{0,1\}^c$.
- $\textsc{Rsp} \leftarrow \mathsf{P}^{\mathcal{GID}}(\mathfrak{gsk}_i, \textsc{Cmt}, \textsc{Ch}, R)$.
- $\mathcal{T} \leftarrow (\textsc{Cmt}, \textsc{Ch}, \textsc{Rsp})$.
- Return $\mathcal{T}$.

Fig. 1. Oracles defining security for canonical group identification schemes

table $\mathfrak{Inf}_*$ which the opener uses to identify provers (this corresponds to the RReg and WReg oracles of [9]). This is purely for syntactic convenience, and this assumption can be removed in the standard way. We do not describe this in detail. To know the detail, please refer to [9].

Experiment $\mathsf{Exp}^{corr}_{\mathcal{GID},\mathcal{A}}(k)$
- $(\mathfrak{gmpk}, \mathfrak{gmsk}, \mathfrak{osk}) \leftarrow \mathsf{GKg}^{\mathcal{GID}}(1^k)$.
- $\mathsf{CU}, \mathsf{HU} \leftarrow \emptyset$.
- $i \leftarrow \mathcal{A}(\mathfrak{gmpk} : \mathsf{AddU}(\cdot))$.
- If $i \notin \mathsf{HU}$ then return 0.
- If $\mathfrak{gsk}_i = \perp$ then return 0.
- $\mathcal{T} \leftarrow \mathsf{Exec}(i)$.
- If $\mathsf{V}^{\mathcal{GID}}(\mathfrak{gmpk}, \mathcal{T}) = 0$ then return 1.
- $(j, \tau) \leftarrow \mathsf{Open}^{\mathcal{GID}}(\mathcal{T}, \mathfrak{osk}, \mathfrak{Inf}_*)$.
- If $i \neq j$ then return 1.
- If $\mathsf{Judge}^{\mathcal{GID}}(\mathfrak{gmpk}, i, \mathfrak{pk}_i, \mathcal{T}, \tau) = 0$ then return 1.
- Return 0.

Experiment $\mathsf{Exp}^{anon\text{-}b}_{\mathcal{GID},\mathcal{A}}(k)$
- $(\mathfrak{gmpk}, \mathfrak{gmsk}, \mathfrak{osk}) \leftarrow \mathsf{GKg}^{\mathcal{GID}}(1^k)$.
- $\mathsf{CU}, \mathsf{HU}, \mathsf{TL} \leftarrow \emptyset$.
- $d \leftarrow \mathcal{A}(\mathfrak{gmpk}, \mathfrak{gmsk} : \mathsf{SndToU}(\cdot,\cdot), \mathsf{CrptU}(\cdot,\cdot), \mathsf{USK}(\cdot), \mathsf{Open}(\cdot), \mathsf{CH}_b(\cdot,\cdot))$.
- Return d.

Experiment $\mathsf{Exp}^{trace}_{\mathcal{GID},\mathcal{A}}(k)$
- $(\mathfrak{gmpk}, \mathfrak{gmsk}, \mathfrak{osk}) \leftarrow \mathsf{GKg}^{\mathcal{GID}}(1^k)$.
- $\mathsf{CU}, \mathsf{HU} \leftarrow \emptyset$.
- $(\textsc{Cmt}, \mathsf{state}) \leftarrow \mathcal{A}_1(\mathfrak{gmpk}, \mathfrak{osk} : \mathsf{AddU}(\cdot), \mathsf{SndToI}(\cdot,\cdot), \mathsf{CrptU}(\cdot,\cdot), \mathsf{USK}(\cdot))$.
- $\textsc{Ch} \leftarrow \{0,1\}^c$.
- $\textsc{Rsp} \leftarrow \mathcal{A}_2(\textsc{Ch}, \mathsf{state} : \mathsf{AddU}(\cdot), \mathsf{SndToI}(\cdot,\cdot), \mathsf{CrptU}(\cdot,\cdot), \mathsf{USK}(\cdot))$.
- $\mathcal{T} \leftarrow (\textsc{Cmt}, \textsc{Ch}, \textsc{Rsp})$.
- If $\mathsf{V}^{\mathcal{GID}}(\mathfrak{gmpk}, \mathcal{T}) = 0$ then return 0.
- $(i, \tau) \leftarrow \mathsf{Open}^{\mathcal{GID}}(\mathcal{T}, \mathfrak{osk}, \mathfrak{Inf}_*)$.
- If $i = 0$ or $\mathsf{Judge}^{\mathcal{GID}}(\mathfrak{gmpk}, i, \mathfrak{pk}_i, \mathcal{T}, \tau) = 0$ then return 1.
- Return 0.

Experiment $\mathsf{Exp}^{non\text{-}frame}_{\mathcal{GID},\mathcal{A}}(k)$
- $(\mathfrak{gmpk}, \mathfrak{gmsk}, \mathfrak{osk}) \leftarrow \mathsf{GKg}^{\mathcal{GID}}(1^k)$.
- $\mathsf{CU}, \mathsf{HU} \leftarrow \emptyset$.
- $(\textsc{Cmt}, \mathsf{state}) \leftarrow \mathcal{A}_1(\mathfrak{gmpk}, \mathfrak{gmsk}, \mathfrak{osk} : \mathsf{SndToU}(\cdot,\cdot), \mathsf{CrptU}(\cdot,\cdot), \mathsf{USK}(\cdot), \mathsf{Exec}(\cdot))$.
- $\textsc{Ch} \leftarrow \{0,1\}^c$.
- $(\textsc{Rsp}, i, \tau) \leftarrow \mathcal{A}_2(\textsc{Ch}, \mathsf{state} : \mathsf{SndToU}(\cdot,\cdot), \mathsf{CrptU}(\cdot,\cdot), \mathsf{USK}(\cdot), \mathsf{Exec}(\cdot))$.
- $\mathcal{T} \leftarrow (\textsc{Cmt}, \textsc{Ch}, \textsc{Rsp})$.
- If $\mathsf{V}^{\mathcal{GID}}(\mathfrak{gmpk}, \mathcal{T}) = 0$ then return 0.
- If the following are all true then return 1 else return 0.
  - $\mathsf{Judge}^{\mathcal{GID}}(\mathfrak{gmpk}, i, \mathfrak{pk}_i, \mathcal{T}, \tau) = 1$ and $i \in \mathsf{HU}$ and $\mathfrak{gsk}_i \neq \perp$.
  - $\mathcal{A}$ did not query $\mathsf{USK}(i)$ and $\mathcal{T}$ was not produced by $\mathsf{Exec}(i)$.

Fig. 2. Security experiments for canonical group identification schemes

Correctness: We require that transcripts produced by honest users are accepted by the verifiers, and that the opening algorithm correctly identifies the

user that produced a transcript. To formalise this we associate to the group identification scheme $\mathcal{GID}$, any adversary $\mathcal{A}$ and any $k \in \mathbb{N}$ the experiment $\mathsf{Exp}^{corr}_{\mathcal{GID},\mathcal{A}}(k)$ defined in Figure 2 where the adversary may want to make a valid transcript cannot be accepted by the verifiers, or make opener cannot correctly identify the prover, or let the proof τ cannot be correctly judge. We define

$$\mathsf{Adv}^{corr}_{\mathcal{GID},\mathcal{A}}(k) = \Pr[\mathsf{Exp}^{corr}_{\mathcal{GID},\mathcal{A}}(k) = 1],$$

and we say that the scheme is *correct* if $\mathsf{Adv}^{corr}_{\mathcal{GID},\mathcal{A}}(k) = 0$ for all adversaries $\mathcal{A}$ and all $k \in \mathbb{N}$.

Anonymity: Let $\mathcal{A}$ be an adversary performing anonymity experiment given in Figure 2 for $b \in \{0, 1\}$. The goal of the adversary is to determine which of two identities has engaged in a run of the identification protocol. In this experiment, the adversary can access the SndToU, CrptU, USK, and Open oracles to get some state information. The adversary uses queries to the CH_b oracle to determine the hidden bit b and hence break the anonymity of $\mathcal{GID}$. We define

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

and we say that the scheme has *anonymity* if $\mathsf{Adv}^{anon}_{\mathcal{GID},\mathcal{A}}(k)$ is a negligible function of k for any polynomial time adversary $\mathcal{A}$.

Traceability: Let $\mathcal{A}$ be an adversary, running in two stages, performing the traceability experiment given in Figure 2. The goal of the adversary is to produce a transcript that is either declared by the opener to be un-openable, or the opener believes they have identified the opener but they cannot produce a valid proof of this. In this experiment, the adversary can first access the AddU, SndToI, CrptU, USK oracles to obtain state information and then output a commitment $\textsc{Cmt}$. After the verifier has outputted the challenge $\textsc{Ch}$, the adversary queries the above oracles and finally outputs a response $\textsc{Rsp}$ associated with $\textsc{Cmt}$ and $\textsc{Rsp}$. The transcript $\mathcal{T}$ is $(\textsc{Cmt}, \textsc{Ch}, \textsc{Rsp})$. We define

$$\mathsf{Adv}^{trace}_{\mathcal{GID},\mathcal{A}}(k) = \Pr[\mathsf{Exp}^{trace}_{\mathcal{GID},\mathcal{A}}(k) = 1],$$

and we say that the scheme has *traceability* if $\mathsf{Adv}^{trace}_{\mathcal{GID},\mathcal{A}}(k)$ is a negligible function of k for any polynomial time adversary $\mathcal{A}$.

Non-Frameability: Let $\mathcal{A}$ be an adversary, also running in two stages, performing the non-frameability experiment given in Figure 2. The goal of the adversary is to output a new transcript which the judge will accept as belonging to an honest user i, where i did not produce this transcript. In this experiment, the adversary can first access the SndToU, CrptU, USK and Exec oracles to obtain state information and it then outputs a commitment $\textsc{Cmt}$. After the verifier has outputted the challenge $\textsc{Ch}$, the adversary queries the above oracles and

finally outputs a response Rsp associated with Cmt and Rsp. The transcript $\mathcal{T}$ is (Cmt, Ch, Rsp). We define

$$\mathsf{Adv}^{non\text{-}frame}_{\mathcal{GID},\mathcal{A}}(k) = \Pr[\mathsf{Exp}^{non\text{-}frame}_{\mathcal{GID},\mathcal{A}}(k) = 1],$$

and we say that the scheme has *non-frameability* if $\mathsf{Adv}^{non\text{-}frame}_{\mathcal{GID},\mathcal{A}}(k)$ is a negligible function of k for any polynomial time adversary $\mathcal{A}$.

Model Variations. Our main results relate the security of group signature schemes with the security of the group identification schemes from which they are obtained via the FS transform. The notions that we use are those defined above.

The group signature literature contains other, still reasonable security notions that we weaker. Our results extend to this setting. Via the FS transform one obtains group signature schemes that satisfies weaker notions of security from, correspondingly weakend group identification schemes. Below we sketch these weaker notions by analogy with those for group signatures.

First we note that the above definitions capture the notion of dynamic groups. In the case of a static groups we may have no $\mathsf{UKg}^{\mathcal{GID}}$ algorithm and no $(\mathsf{Join}^{\mathcal{GID}}, \mathsf{Iss}^{\mathcal{GID}})$ protocol for joining a group. Instead, the generation of user secret keys $\mathfrak{gsk}_i$ is assumed to be done by the setup algorithm $\mathsf{GKg}^{\mathcal{GID}}$, and is done once and for all on system setup. The experiments then need to be altered slightly in the obvious way, mainly to remove adversarial calls to the $\mathsf{AddU}, \mathsf{SndToU}$ and SndToI oracles. In analogy with the definitions from [6] in many schemes the openers secret key $\mathfrak{osk}$ is identical to the group managers secret key $\mathfrak{gmsk}$. We say that such system have an *opener-manager*. In another variant, also considered in [6], the algorithm Open^{GID} does not output a proof of correctness of the opening (to be verified by a judge) but simply outputs the identity i. These are schemes with *non-verified opening*. Again the security experiments need to be slightly modified with respect to how $\mathsf{Open}^{\mathcal{GID}}$ and $\mathsf{Judge}^{\mathcal{GID}}$ work, since there is now no proof and so no need of the $\mathsf{Judge}^{\mathcal{GID}}$ algorithm.

Finally, a scheme which does not allow the adversary to query the opening oracle in the anonymity experiment is said to be weakly secure, or simply CPA secure. One can think of the identity information within the transcript used by the opener as an encryption of this identity. Thus giving the adversary access or not to the opening oracle is akin to giving access to a decryption oracle in the security model for encryption schemes – hence the name.

3 Group Signature Schemes

In this section we describe the syntax and security notions for group signature schemes. The presentation follows closely [9], which also served as guidance for our model for group identification schemes. As such, there is a lot of commonality between the two presentations so we only stress the main differences.

Syntax. A group signature scheme is given by a tuple of algorithms: $\mathcal{GS} = (\mathsf{GKg}^{\mathcal{GS}}, \mathsf{UKg}^{\mathcal{GS}}, \mathsf{Join}^{\mathcal{GS}}, \mathsf{Iss}^{\mathcal{GS}}, \mathsf{GSig}, \mathsf{GVf}, \mathsf{Open}^{\mathcal{GS}}, \mathsf{Judge}^{\mathcal{GS}})$. The functionality

the algorithms $\mathsf{GKg}^{\mathcal{GS}}$, $\mathsf{UKg}^{\mathcal{GS}}$, $\mathsf{Join}^{\mathcal{GS}}$ and $\mathsf{Iss}^{\mathcal{GS}}$ is identical to those for the group identification schemes considered earlier. What is different is that the prover and verifier interactive algorithms are replaced with a signing algorithm GSig and a verification algorithm GVf. The syntax demanded from the algorithms for opening and judging $\mathsf{Open}^{\mathcal{GS}}$ and $\mathsf{Judge}^{\mathcal{GS}}$ is slightly modified to take this into account. Specifically:

- GSig: Is a probabilistic signing algorithm taking input a group signing key $\mathfrak{gsk}_i$ and a message m, returning a signature σ.
- GVf: Is a deterministic verifying algorithm which takes input the group public key $\mathfrak{gmpk}$, a group signature σ a message m. It then returns a Boolean decision to demonstrate whether the group signature is accepted or rejected.
- $\mathsf{Open}^{\mathcal{GS}}$: This is as before except it takes as input a message and a signature instead of a transcript.
- $\mathsf{Judge}^{\mathcal{GS}}$: Again, this is as before except it takes as input a message and a signature instead of a transcript.

Security notions. The games that define correctness, anonymity, traceability and non-frameability for group signature schemes are essentially the non-interactive versions of the games we have defined for identification schemes. The schemes make use of a modified set of oracles. The modifications are as follows:

The oracles used by the adversary are changed from that for canonical group identification schemes in the following ways:

- $\mathsf{Open}(\sigma, m)$: This oracle takes as input a signature σ and a message m and returns the result of running the opening algorithm (i.e. the identity of the user plus the associated proof).
- $\mathsf{CH}_b(i_0, i_1, m)$: This oracle is a left-right oracle for defining anonymity. The adversary sends a couple of honest identities (i_0, i_1) and a message m to the oracle and gets back a signature σ of the signature scheme executed by signer i_b. In addition, in CH_b and the game for anonymity we replace the list of transcripts TL by a list of signatures SL issued by the oracle CH_b.
- $\mathsf{Sign}(i, m)$: This oracle allows the adversary to obtain signatures of the signature scheme executed by a valid group member. This oracle takes as input the identity of the group member i and message m, and finally outputs a group signature of the member i.

The changes in games account for the fact that we replace identification with signing. In addition, we are only concerned with schemes secure in the random oracle model (as those obtained via the FS transform) so the algorithms and the adversary have access to oracle H defined in the standard way. Specifically, the oracle $\mathsf{H}(\cdot)$ maintains an internal list, H-List, of pairs (x, h) with the meaning that h was the answer that the oracle returned when it was previously queried with input x. When H receives an input x, it then returns h if (x, h) in H-List. Otherwise, it selects a random $h \in \{0,1\}^c$, adds the entry (x, h) to H-List, and returns h.

The formal games for security are in Figure 3, with the associated advantage functions being defined in the obvious manner.

Experiment $\mathsf{Exp}^{corr}_{\mathcal{GS},\mathcal{A}}(k)$
- $(\mathfrak{gmpk}, \mathfrak{gmsk}, \mathfrak{osk}) \leftarrow \mathsf{GKg}^{\mathcal{GS}}(1^k)$.
- $\mathsf{CU}, \mathsf{HU} \leftarrow \emptyset$.
- $(i, m) \leftarrow \mathcal{A}(\mathfrak{gmpk} : \mathsf{AddU}(\cdot))$.
- If $i \notin \mathsf{HU}$ then return 0.
- If $\mathfrak{gsk}_i = \perp$ then return 0.
- $\sigma \leftarrow \mathsf{GSig}(\mathfrak{gsk}_i, m)$.
- If $\mathsf{GVf}(\mathfrak{gmpk}, \sigma, m) = 0$ then return 1.
- $(j, \tau) \leftarrow \mathsf{Open}^{\mathcal{GS}}(\sigma, m, \mathfrak{osk}, \mathfrak{Inf}_*)$.
- If $i \neq j$ then return 1.
- If $\mathsf{Judge}^{\mathcal{GS}}(\mathfrak{gmpk}, i, \mathfrak{pk}_i, \sigma, m, \tau) = 0$ then return 1.
- Return 0.

Experiment $\mathsf{Exp}^{anon\text{-}b}_{\mathcal{GS},\mathcal{A}}(k)$
- $(\mathfrak{gmpk}, \mathfrak{gmsk}, \mathfrak{osk}) \leftarrow \mathsf{GKg}^{\mathcal{GS}}(1^k)$.
- $\mathsf{CU}, \mathsf{HU}, \mathsf{SL} \leftarrow \emptyset$.
- $d \leftarrow \mathcal{A}(\mathfrak{gmpk}, \mathfrak{gmsk} : \mathsf{H}(\cdot), \mathsf{SndToU}(\cdot,\cdot), \mathsf{CrptU}(\cdot,\cdot), \mathsf{USK}(\cdot), \mathsf{Open}(\cdot), \mathsf{CH}_b(\cdot,\cdot))$.
- Return d.

Experiment $\mathsf{Exp}^{trace}_{\mathcal{GS},\mathcal{A}}(k)$
- $(\mathfrak{gmpk}, \mathfrak{gmsk}, \mathfrak{osk}) \leftarrow \mathsf{GKg}^{\mathcal{GS}}(1^k)$.
- $\mathsf{CU}, \mathsf{HU} \leftarrow \emptyset$.
- $(\sigma, m) \leftarrow \mathcal{A}(\mathfrak{gmpk}, \mathfrak{osk} : \mathsf{H}(\cdot), \mathsf{AddU}(\cdot), \mathsf{SndToI}(\cdot,\cdot), \mathsf{CrptU}(\cdot,\cdot), \mathsf{USK}(\cdot))$.
- If $\mathsf{GVf}(\mathfrak{gmpk}, \sigma, m) = 0$ then return 0.
- $(i, \tau) \leftarrow \mathsf{Open}^{\mathcal{GS}}(\sigma, m, \mathfrak{osk}, \mathfrak{Inf}_*)$.
- If $i = 0$ or $\mathsf{Judge}^{\mathcal{GS}}(\mathfrak{gmpk}, i, \mathfrak{pk}_i, \sigma, m, \tau) = 0$ then return 1.
- Return 0.

Experiment $\mathsf{Exp}^{non\text{-}frame}_{\mathcal{GS},\mathcal{A}}(k)$
- $(\mathfrak{gmpk}, \mathfrak{gmsk}, \mathfrak{osk}) \leftarrow \mathsf{GKg}^{\mathcal{GS}}(1^k)$.
- $\mathsf{CU}, \mathsf{HU} \leftarrow \emptyset$.
- $(\sigma, m, i, \tau) \leftarrow \mathcal{A}(\mathfrak{gmpk}, \mathfrak{gmsk}, \mathfrak{osk} : \mathsf{H}(\cdot), \mathsf{SndToU}(\cdot,\cdot), \mathsf{CrptU}(\cdot,\cdot), \mathsf{USK}(\cdot), \mathsf{Sign}(\cdot))$.
- If $\mathsf{GVf}(\mathfrak{gmpk}, \sigma, m) = 0$ then return 0.
- If the following are all true then return 1 else return 0.
  - $\mathsf{Judge}^{\mathcal{GS}}(\mathfrak{gmpk}, i, \mathfrak{pk}_i, \sigma, m, \tau) = 1$ and $i \in \mathsf{HU}$ and $\mathfrak{gsk}_i \neq \perp$.
  - $\mathcal{A}$ did not query $\mathsf{USK}(i)$ and σ was not produced by a call to $\mathsf{Sign}(i, m)$.

Fig. 3. Security experiments for group signature schemes

As for group identification, one can define weaker notions of security for group signature schemes by appropriate restrictions and syntactic modifications. The standard examples from the literature include moving to static groups, CPA security, non-verified opening and schemes with an opener-manager.

4 From Group Identification to Group Signature Schemes

In this section we formally define the Fiat-Shamir transform for group signature and prove that it leads to secure schemes.

THE FIAT-SHAMIR TRANSFORM. The Fiat-Shamir transform for standard digital signature schemes works if the underlying identification protocol is such that the first message (the commitment) has sufficient entropy. To make the transformation applicable to a larger class of identification protocols, one workaround is to "artificially" append a random string to the commitment. Abdalla et al. [1] call this the extended Fiat-Shamir transform. We adapt this more general transformation to the setting of group identification/signatures. Similarly to [1], our security results would then subsume the case when the commitment of the original scheme has sufficient entropy.

The transformation essentially removes the interaction in the identification protocol of the group identification scheme, very much like it does when applied to standard identification protocols.

Let $\mathcal{GID} = (\mathsf{GKg}^{\mathcal{GID}}, \mathsf{UKg}^{\mathcal{GID}}, \mathsf{Join}^{\mathcal{GID}}, \mathsf{Iss}^{\mathcal{GID}}, (\mathsf{P}^{\mathcal{GID}}, \mathsf{V}^{\mathcal{GID}}), \mathsf{Open}^{\mathcal{GID}}, \mathsf{Judge}^{\mathcal{GID}})$ be a canonical group identification scheme, and $s : \mathbb{N} \to \mathbb{N}$ be a function which defines a *seed length* $s(k)$ given the security parameter k. We select a hash function $H : \{0,1\}^* \to \{0,1\}^c$ at random from the set of all maps $\{0,1\}^* \to \{0,1\}^c$, where c is the bit length of the challenge CH in the canonical group identification scheme we will be using. From these we construct a group signature scheme $\mathcal{GS} = (\mathsf{GKg}^{\mathcal{GS}}, \mathsf{UKg}^{\mathcal{GS}}, \mathsf{Join}^{\mathcal{GS}}, \mathsf{Iss}^{\mathcal{GS}}, \mathsf{GSig}, \mathsf{GVf}, \mathsf{Open}^{\mathcal{GS}}, \mathsf{Judge}^{\mathcal{GS}})$ as follows. We let $\mathsf{GKg}^{\mathcal{GS}} = \mathsf{GKg}^{\mathcal{GID}}$, $\mathsf{UKg}^{\mathcal{GS}} = \mathsf{UKg}^{\mathcal{GID}}$, $\mathsf{Join}^{\mathcal{GS}} = \mathsf{Join}^{\mathcal{GID}}$ and $\mathsf{Iss}^{\mathcal{GS}} = \mathsf{Iss}^{\mathcal{GID}}$. The functions $\mathsf{GSig}, \mathsf{GVf}, \mathsf{Open}^{\mathcal{GS}}$ and $\mathsf{Judge}^{\mathcal{GS}}$ are defined as in Figure 4. We call the resulting group signature scheme $FS(\mathcal{GID})$.

$\mathsf{GSig}(\mathfrak{gsk}_i, m)$:
- $R_P \leftarrow \mathsf{Coins}(\mathsf{P}^{\mathcal{GID}})$.
- $\textsc{Cmt} \leftarrow \mathsf{P}^{\mathcal{GID}}(\mathfrak{gsk}_i; R_P)$.
- $R \leftarrow \{0,1\}^{s(k)}$.
- $\textsc{Ch} \leftarrow H(R \| \textsc{Cmt} \| m)$.
- $\textsc{Rsp} \leftarrow \mathsf{P}^{\mathcal{GID}}(\mathfrak{gsk}_i, \textsc{Cmt}, \textsc{Ch}, R_P)$.
- $\sigma \leftarrow (R, \textsc{Cmt}, \textsc{Rsp})$.
- Return σ.

$\mathsf{GVf}(\mathfrak{gmpk}, \sigma, m)$:
- Parse σ as $(R, \textsc{Cmt}, \textsc{Rsp})$.
- $\textsc{Ch} \leftarrow H(R \| \textsc{Cmt} \| m)$.
- $\mathcal{T} \leftarrow (\textsc{Cmt}, \textsc{Ch}, \textsc{Rsp})$.
- Return $\mathsf{V}^{\mathcal{GID}}(\mathfrak{gmpk}, \mathcal{T})$.

$\mathsf{Open}^{\mathcal{GS}}(\sigma, m, \mathfrak{osk}, \mathfrak{Inf}_*)$:
- Parse σ as $(R, \textsc{Cmt}, \textsc{Rsp})$.
- $\textsc{Ch} \leftarrow H(R \| \textsc{Cmt} \| m)$.
- $\mathcal{T} \leftarrow (\textsc{Cmt}, \textsc{Ch}, \textsc{Rsp})$.
- Return $\mathsf{Open}^{\mathcal{GID}}(\mathcal{T}, \mathfrak{osk}, \mathfrak{Inf}_*)$.

$\mathsf{Judge}^{\mathcal{GS}}(\mathfrak{gmpk}, i, \mathfrak{pk}_i, \sigma, m, \tau)$:
- Parse σ as $(R, \textsc{Cmt}, \textsc{Rsp})$.
- $\textsc{Ch} \leftarrow H(R \| \textsc{Cmt} \| m)$.
- $\mathcal{T} \leftarrow (\textsc{Cmt}, \textsc{Ch}, \textsc{Rsp})$.
- Return $\mathsf{Judge}^{\mathcal{GID}}(\mathfrak{gmpk}, i, \mathfrak{pk}_i, \mathcal{T}, \tau)$.

Fig. 4. Construction of a group signature scheme from a group identification scheme

SECURITY RESULTS. Since the security of the resulting group signature schemes relies on the entropy of the commitment we recall the necessary notion. Security of the above construction relies on the random oracle model. In addition it relies on the values of the constants $s(k)$ and c. In particular the associated min-

entropy, defined below, of the commitment generated by the prover needs to be large enough.

Definition 1 (Min-Entropy of Commitments). *Let $\mathcal{GID}$ be a canonical group identification scheme. Let $k \in N$ and $(\mathfrak{sk}_i, \mathfrak{pk}_i)$ be the key pair generated by key generation algorithm $\mathsf{UKg}^{\mathcal{GID}}$ on input of 1^k. We denote by $\mathcal{C}(\mathfrak{sk}_i) = \left\{\text{CMT} = \mathsf{P}^{\mathcal{GID}}(\mathfrak{sk}_i, R_P)\right\}$ be the set of all possible commitments associated with $\mathfrak{sk}_i$. We define the maximum probability that a commitment takes on a particular value via*

$$\alpha(\mathfrak{sk}_i) = \max_{\text{CMT} \in \mathcal{C}(\mathfrak{sk}_i)} \left\{ Pr\left[\mathsf{P}^{\mathcal{GID}}(\mathfrak{sk}_i, R_P) = \text{CMT} : R_P \leftarrow \mathsf{Coins}(\mathsf{P}^{\mathcal{GID}})\right]\right\}.$$

Then the min-entropy function associated with $\mathcal{GID}$ is defined as follows:

$$\beta(k) = \min_{\mathfrak{sk}_i} \left\{ \log_2 \left(\frac{1}{\alpha(\mathfrak{sk}_i)} \right) \right\}$$

where minimum is taken over all key pairs $(\mathfrak{sk}_i, \mathfrak{pk}_i)$ generated by $\mathsf{UKg}^{\mathcal{GID}}(1^k)$. We say that $\mathcal{GID}$ is non-trivial if $\beta(\cdot) = \omega(\log(\cdot))$ is super-logarithmic.

Our results show a tight connection between the security of the underlying group identification schemes and the group signature scheme obtained via the FS transform. If the starting group signature scheme is secure (it has the four properties that we have defined earlier), then the resulting group signature scheme is also secure. This result is captured by the following theorem.

Theorem 1. *(Secure $\mathcal{GID} \Rightarrow$ secure $\mathcal{GS}$) Let $\mathcal{GID}$ be a canonical group identification scheme and $\mathcal{GS} = FS(\mathcal{GID})$. If $\mathcal{GID}$ has the properties of correctness, anonymity, traceability and non-frameability under passive attacks, then $\mathcal{GS}$ also has the above properties.*

We also show that security against passive adversaries for the underlying group identification scheme is also necessary. Specifically, we have the following theorem.

Theorem 2. *(Secure $\mathcal{GID} \Leftarrow$ secure $\mathcal{GS}$) Let $\mathcal{GID}$ be a canonical group identification scheme and $\mathcal{GS} = FS(\mathcal{GID})$. If $\mathcal{GS}$ has the properties of correctness, anonymity, traceability and non-frameability, then $\mathcal{GID}$ is correct, anonymous, traceable and non-frameable under passive attacks.*

MODEL VARIATIONS. As remarked earlier different authors have used different notions of security for group signature schemes. Each of these different notions is obtained by appropriate restrictions on the powers of the adversary in the standard security games. Unsurprisingly for both group identification and signature schemes, the restrictions are essentially the same (modulo the parts that are different). For example, in both cases, CPA-security is obtained by not providing the adversary with an oracle for opening transcripts and signatures, respectively. Since this is true for all of the oracles which our reductions

preserve essentially unchanged, our proofs easily extend to these variations in models. Specifically, we have the following: If X is one of the properties in the set $\{\mathsf{correctness}, \mathsf{anonymity}, \mathsf{traceability}, \mathsf{non\text{-}frameability}, \mathsf{CPA\text{-}secure}, \mathsf{CCA\text{-}secure}\}$ then, if $\mathcal{GID}$ has property X, then $\mathcal{GS} = FS(\mathcal{GID})$ has property X. This is true for both static and dynamic groups. In the following table we summarize the security models used in prior work on group signatures that appeared after 2003.

	Dynamic	CCA Secure	Has Judge	Opener $\neq$ Manager	Standard Model
BMW[6]		✔			
BBS[5],BS[8]					
BW[10, 11]				✔	✔
MU[26],ZZW[35],NF[27],HWL[22]	✔				
LCSL[25]	✔			✔	✔
NKHF[30],NS[31]	✔			✔	
NFHF[28],KY[24]	✔	✔			
FI[18]	✔	✔		✔	
WYZ[34]	✔	✔	✔	✔	
BSZ[9],G[20]	✔	✔	✔	✔	✔

5 Proof of the Construction

5.1 Proof of Theorem 1

The concept of our proof for **Theorem 1** is as follow: if $\mathcal{GS} = FS(\mathcal{GID})$ is insecure (i.e., there exists an algorithm $\mathcal{A}$ which can break the security of $\mathcal{GS}$ with non-negligible advantage), then there exists a algorithm $\mathcal{B}$ which can break the security of $\mathcal{GID}$ with non-negligible advantage. We now prove **Theorem 1** via Lemma 1 to Lemma 4.

Lemma 1. *Let $\mathcal{GID}$ be a group identification scheme and $\mathcal{GS} = FS(\mathcal{GID})$. Let $\mathcal{A}$ be an adversary attacking the correctness of the group signature in the random oracle model. Then there is an adversary $\mathcal{B}$ against the correctness of $\mathcal{GID}$ such that $\mathsf{Adv}^{corr}_{\mathcal{GID},\mathcal{B}} \geq \mathsf{Adv}^{corr}_{\mathcal{GS},\mathcal{A}}$.*

Proof. Assume $\mathcal{B}$ is an algorithm attacking the correctness of $\mathcal{GID}$ and $\mathcal{A}$ is an algorithm against anonymity of $\mathcal{GS}$. The goal of $\mathcal{B}$ is to use $\mathcal{A}$ to gain advantage when it runs $\mathsf{Exp}^{corr}_{\mathcal{GID},\mathcal{B}}(k)$ and accesses the associated oracle $\mathsf{AddU}^{\mathcal{GID}}(i, M)$. To achieve this goal, $\mathcal{B}$ should run the algorithm $\mathcal{A}$, simulate the environment of $\mathcal{A}$ in $\mathsf{Exp}^{corr}_{\mathcal{GS},\mathcal{A}}(k)$ with the $\mathsf{AddU}^{\mathcal{GID}}(i, M)$ oracle in $\mathsf{Exp}^{corr}_{\mathcal{GID},\mathcal{B}}(k)$. We now construct the algorithm $\mathcal{B}$ running $\mathcal{A}$ to gain advantage against $\mathcal{GID}$. Suppose $\mathcal{B}$ plays the correctness game, runs $\mathsf{Exp}^{corr}_{\mathcal{GID},\mathcal{B}}(k)$ and accesses the oracle of $\mathsf{AddU}^{\mathcal{GID}}(i, M)$. First, $\mathcal{B}$ runs $\mathcal{A}$ and simulates the oracle for $\mathcal{A}$. Then $\mathcal{B}$ just needs to set $\mathsf{AddU}^{\mathcal{GS}}(i, M) = \mathsf{AddU}^{\mathcal{GID}}(i, M)$. Therefore, we can easily have $\mathsf{Adv}^{corr}_{\mathcal{GID},\mathcal{B}} \geq \mathsf{Adv}^{corr}_{\mathcal{GS},\mathcal{A}}$.

Lemma 2. *Let $\mathcal{GID}$ be a group identification scheme and $\mathcal{GS} = FS(\mathcal{GID})$. Let $s(\cdot)$ be a seed length and $\beta(\cdot)$ be the min-entropy function associated with $\mathcal{GID}$. Let $\mathcal{A}$ be an adversary attacking the anonymity of the group signature in the random oracle model, making q_h hash-oracle queries. Then there is an adversary $\mathcal{B}$ against the anonymity of $\mathcal{GID}$ such that* $\mathsf{Adv}^{anon}_{\mathcal{GID},\mathcal{B}} \geq \mathsf{Adv}^{anon}_{\mathcal{GS},\mathcal{A}} - \frac{q_h}{2^{s(k)+\beta(k)}}$.

Proof. Assume $\mathcal{B}$ is an algorithm attacking the anonymity of $\mathcal{GID}$ and $\mathcal{A}$ is an algorithm against anonymity of $\mathcal{GS}$. The goal of $\mathcal{B}$ is to use $\mathcal{A}$ to gain advantage when it runs $\mathsf{Exp}^{anon\text{-}b}_{\mathcal{GID},\mathcal{B}}(k)$ and accesses the associated oracles. To achieve this goal, $\mathcal{B}$ should run the algorithm $\mathcal{A}$, simulate the environment of $\mathcal{A}$ in $\mathsf{Exp}^{anon\text{-}b}_{\mathcal{GS},\mathcal{A}}(k)$ with the oracles in $\mathsf{Exp}^{anon\text{-}b}_{\mathcal{GID},\mathcal{B}}(k)$. We now construct the algorithm $\mathcal{B}$ running $\mathcal{A}$ to gain advantage against $\mathcal{GID}$. Suppose $\mathcal{B}$ plays the anonymity game, runs $\mathsf{Exp}^{anon\text{-}b}_{\mathcal{GID},\mathcal{B}}(k)$ and accesses the oracles of $\mathsf{SndToU}^{\mathcal{GID}}(i, M)$, $\mathsf{SndToI}^{\mathcal{GID}}(i, M)$, $\mathsf{CrptU}^{\mathcal{GID}}(i, \mathfrak{pk})$, $\mathsf{USK}^{\mathcal{GID}}(i)$, $\mathsf{Open}^{\mathcal{GID}}(\mathcal{T})$ and $\mathsf{CH}^{\mathcal{GID}}_b(i_0, i_1)$. First, $\mathcal{B}$ runs $\mathcal{A}$ and simulates the oracles for $\mathcal{A}$. Then $\mathcal{B}$ defines $\mathsf{SndToU}^{\mathcal{GS}}(i, M)$, $\mathsf{SndToI}^{\mathcal{GS}}(i, M)$, $\mathsf{CrptU}^{\mathcal{GS}}(i, \mathfrak{pk})$, and $\mathsf{USK}^{\mathcal{GS}}(i)$. to be the equivalent oracles in the $\mathcal{GID}$ game. Then $\mathcal{B}$ constructs $\mathsf{H}(x)$ and $\mathsf{Open}^{\mathcal{GS}}(\sigma, m)$ oracles for $\mathcal{A}$ as follows:

$\mathsf{H}(x)$:

- Maintain the H-List of pairs (x, h).
- When $\mathcal{A}$ queries x, return $H(x)$ if it is defined.
- Pick y at random from $\{0,1\}^c$.
- $H(x) \leftarrow y$, return $H(x)$.

$\mathsf{Open}^{\mathcal{GS}}(\sigma, m)$

- Parse σ as R, C_{MT}, R_{SP}.
- Look-up $H(R\|\textsc{Cmt}\|m)$, if it is not in the table then return $\perp$.
- $\textsc{Ch} \leftarrow H(R\|\textsc{Cmt}\|m)$.
- $\mathcal{T} \leftarrow (\textsc{Cmt}, \textsc{Ch}, \textsc{Rsp})$
- Return $\mathsf{Open}(\mathcal{T})$.

When $\mathcal{A}$ makes one of the above oracles queries, Algorithm $\mathcal{B}$ answers $\mathcal{A}$ with its own queries according the above simulation. Then $\mathcal{A}$ makes one challenge oracle query, which $\mathcal{B}$ answers by calling its oracle and return a signature according to the following $\mathsf{CH}^{\mathcal{GS}}_b(i_0, i_1, m)$ simulation:

- $\mathcal{T} \leftarrow \mathsf{CH}^{\mathcal{GID}}_b(i_0, i_1)$.
- Parse $\mathcal{T}$ as $\textsc{Cmt}^*$, $\textsc{Ch}^*$, $\textsc{Rsp}^*$.
- $R^* \leftarrow \{0,1\}^{s(k)}$.
- If $H(R^*\|\textsc{Cmt}^*\|m)$ is in H-List then abort.
- Add $(H(R^*\|\textsc{Cmt}^*\|m), \textsc{Ch}^*)$ to the H-List.
- Patch H-list such that $\textsc{Ch} = H(R^*\|\textsc{Cmt}^*\|m)$.
- $\sigma \leftarrow (R, \textsc{Cmt}, \textsc{Rsp})$.
- Return σ.

Finally, $\mathcal{A}$ outputs a decision bit d for the Experiment $\mathsf{Exp}^{anon\text{-}b}_{\mathcal{GS},\mathcal{A}}(k)$. Algorithm $\mathcal{B}$ returns d as the answer to its own challenge for the Experiment $\mathsf{Exp}^{anon\text{-}b}_{\mathcal{GID},\mathcal{B}}(k)$. Let F be the event that $\mathcal{A}$ wins the anonymity game and S be the event that $\mathcal{A}$ aborts during the simulation of the challenge oracle for $\mathcal{A}$. In other words, S is the event that $\mathcal{A}$ had queried $R^*\|\textsc{Cmt}^*\|m$ to the oracle before. We first give the upper bound on $Pr[S]$. If q_h is the number of oracle queries made by $\mathcal{A}$, the H contains at most q_h execution times. Since $\textsc{Ch}^*$ is chosen uniformly at random from $\{0,1\}^{\beta(k)}$ and R^* is chosen uniformly at random from $\{0,1\}^{s(k)}$, a simple union bound gives that $Pr[S] \leq \frac{q_h}{2^{s(k)_{\beta(k)}}}$. If the simulation is perfect, $\mathcal{B}$ wins the anonymity game whenever $\mathcal{A}$ wins in the Experiment $\mathsf{Exp}^{anon\text{-}b}_{\mathcal{GS},\mathcal{A}}(k)$ simulation. If S occurs, then the simulation aborts. Therefore, the advantage of $\mathcal{B}$ can be bounded by

$$\mathsf{Adv}^{anon}_{\mathcal{GID},\mathcal{B}} \geq \Pr[F \wedge \neg S]$$

So, we can derive the advantage of $\mathcal{B}$

$$\begin{aligned}\mathsf{Adv}^{anon}_{\mathcal{GID},\mathcal{B}} &\geq \Pr[F \wedge \neg S] \geq \Pr[F](1 - Pr[S]) \geq \Pr[F] - \Pr[S] \\ &\geq \mathsf{Adv}^{anon}_{\mathcal{GS},\mathcal{A}} - \frac{q_h}{2^{s(k)+\beta(k)}}.\end{aligned}$$

Lemma 3. *Let $\mathcal{GID}$ be a group identification scheme and $\mathcal{GS} = FS(\mathcal{GID})$. Let $s(\cdot)$ be a seed length and $\beta(\cdot)$ be the min-entropy function associated with $\mathcal{GID}$. Let $\mathcal{A}$ be an adversary attacking the traceability of the group signature in the random oracle model, making q_h hash-oracle queries. Then there is an adversary $\mathcal{B}$ against the traceability of $\mathcal{GID}$ such that $\mathsf{Adv}^{trace}_{\mathcal{GID},\mathcal{B}} \geq \mathsf{Adv}^{trace}_{\mathcal{GS},\mathcal{A}} - \frac{q_h}{2^{s(k)+\beta(k)}}$.*

Proof. Let $\mathcal{B}$ be an algorithm attacking traceability of GID by running algorithm $\mathcal{A}$ and simulating the $\mathsf{Exp}^{trace}_{\mathcal{GS},\mathcal{A}}(k)$ environment for $\mathcal{A}$. We assume that $\mathcal{B}$ has access to the oracles $\mathsf{AddU}^{\mathcal{GID}}(i,M)$, $\mathsf{SndToI}^{\mathcal{GID}}(i,M)$, $\mathsf{CrptU}^{\mathcal{GID}}(i,\mathfrak{pk})$ and $\mathsf{USK}^{\mathcal{GID}}(i)$ for the $\mathcal{GID}$ game. We now construct the algorithm $\mathcal{B}$ as follows: $\mathcal{B}$ begins with the initialization: $hc \leftarrow 0$, $fp \leftarrow \{1,...,q_h\}$. Next, $\mathcal{B}$ runs $\mathcal{A}$ and simulates the following oracles for $\mathcal{A}$, $\mathsf{AddU}^{\mathcal{GS}}(i,M)$, $\mathsf{SndToI}^{\mathcal{GS}}(i,M)$, $\mathsf{CrptU}^{\mathcal{GS}}(i,\mathfrak{pk})$, and $\mathsf{USK}^{\mathcal{GS}}(i)$. using the equivalent oracles in the $\mathcal{GID}$ game. When $\mathcal{A}$ makes hash oracle queries, $\mathcal{B}$ constructs $\mathsf{H}(x)$ and answers $\mathcal{A}$ as follows:

$\mathsf{H}(x)$:

- Maintain the H-List of pairs (x,h).
- If $hc \neq fp$
  - If there exists (x,h) in H-List then return h.
  - $hc \leftarrow hc + 1$.
  - Select y at random from $\{0,1\}^c$.
  - Add (x,y) to H-List.
  - Return y.
- Parse x as R, $\textsc{Cmt}^*$, $\textsc{Rsp}$.
- Send $\textsc{Cmt}^*$ to the verifier and get back $\textsc{Ch}^*$.
- Add $(R||\textsc{Cmt}^*||\textsc{Ch}^*)$ to H-List.

– Return $\textsc{Ch}^*$.

Finally, $\mathcal{A}$ outputs a forgery $(R, \textsc{Cmt}, \textsc{Rsp})$. $\mathcal{B}$ sends $\textsc{Rsp}$ to the verifier as the response. Let j be that $Q[j] = R||\textsc{Cmt}||m$. If $j = fp$, then the transcript between $\mathcal{B}$ and the verifier is $\textsc{Cmt}||\textsc{Ch}^*||\textsc{Rsp}$. In this case, $\textsc{Ch} = \textsc{Ch}^*$. Algorithm $\mathcal{B}$ wins the traceability game when either

$$\mathsf{V}^{\mathcal{GID}}(gmpk, (\textsc{Cmt}, \textsc{Ch}^*, \textsc{Rsp})) = 1$$

or

$$\mathsf{Judge}^{\mathcal{GID}}(\mathfrak{gmpk}, i, \mathfrak{pk}_i, (\textsc{Cmt}, \textsc{Ch}^*, \textsc{Rsp}), \tau) = 1.$$

Let F be the event that $\mathcal{A}$ wins the traceability game and S be the event of hash collusion. We give the upper bound on $Pr[S]$. If q_h is the number of oracle queries made by $\mathcal{A}$, the H colludes at most q_h execution times. Since $\textsc{Ch}^*$ is chosen uniformly at random from $\{0,1\}^{\beta(k)}$ and R^* is chosen uniformly at random from $\{0,1\}^{s(k)}$, a simple union bound gives that $Pr[S] \leq \frac{q_h}{2^{s(k)\beta(k)}}$. Therefore, the advantage of $\mathcal{B}$ can be bounded by $\mathcal{B}$ can be bounded by

$$\mathsf{Adv}^{trace}_{\mathcal{GID},\mathcal{B}} \geq \Pr[F \wedge \neg S] \geq \Pr[F](1 - Pr[S]) \geq \Pr[F] - \Pr[S]$$

So, we can derive the advantage of $\mathcal{B}$

$$\mathsf{Adv}^{trace}_{\mathcal{GID},\mathcal{B}} \geq \Pr[F] - \Pr[S] \geq \mathsf{Adv}^{trace}_{\mathcal{GS},\mathcal{A}} - \frac{q_h}{2^{s(k)+\beta(k)}}.$$

Lemma 4. *Let $\mathcal{GID}$ be a group identification scheme and $\mathcal{GS} = FS(\mathcal{GID})$. Let $s(\cdot)$ be a seed length and $\beta(\cdot)$ be the min-entropy function associated with $\mathcal{GID}$. Let $\mathcal{A}$ be an adversary attacking the non-frameability of the group signature in the random oracle model, making q_h hash-oracle queries and q_s signature oracles. Then there is an adversary $\mathcal{B}$ against the non-frameability of $\mathcal{GID}$ such that $\mathsf{Adv}^{non\text{-}frame}_{\mathcal{GID},\mathcal{B}} \geq \mathsf{Adv}^{non\text{-}frame}_{\mathcal{GS},\mathcal{A}}(k) - \frac{1}{q_h} \cdot \frac{q_s(q_h+q_s-1)}{2^{s(k)+\beta(k)}}$.*

Proof. Let $\mathcal{B}$ be an algorithm attacking non-frameability of $\mathcal{GID}$ by running algorithm $\mathcal{A}$ and simulating the $\mathsf{Exp}^{non\text{-}frame}_{\mathcal{GS},\mathcal{A}}(k)$ environment for $\mathcal{A}$. Suppose $\mathcal{B}$ accesses the oracles of $\mathsf{SndToU}^{\mathcal{GID}}(i, M)$, $\mathsf{CrptU}^{\mathcal{GID}}(i, \mathfrak{pk})$ and $\mathsf{USK}^{\mathcal{GID}}(i)$, $\mathsf{USK}^{\mathcal{GID}}(i)$ and $\mathsf{Exec}^{\mathcal{GID}}(i)$. $\mathcal{B}$ begins with the initialization: $hc \leftarrow 0$, $sc \leftarrow 0$, $fp \leftarrow \{1, ..., q_h\}$. Next, $\mathcal{B}$ runs $\mathcal{A}$ and sets $\mathsf{SndToU}^{\mathcal{GS}}(i, M) = \mathsf{SndToU}^{\mathcal{GID}}(i, M)$, $\mathsf{CrptU}^{\mathcal{GS}}(i, \mathfrak{pk}) = \mathsf{CrptU}^{\mathcal{GID}}(i, \mathfrak{pk})$, and $\mathsf{USK}^{\mathcal{GS}}(i) = \mathsf{USK}^{\mathcal{GID}}(i)$.

When $\mathcal{A}$ makes above oracle queries, $\mathcal{B}$ answers by calling its own appropriate queries. When $\mathcal{A}$ makes hash oracle queries and signing oracle queries, $\mathcal{B}$ constructs $\mathsf{H}(x)$ and $\mathsf{Sign}(i, m)$ for $\mathcal{A}$ as follows:

$\mathsf{H}(x)$:
- Maintain H-List (x, h).
- If $hc \neq fp$
  - If there exists (x, h) in H-List then return h.
  - $hc \leftarrow hc + 1$.

- Select y at random from $\{0,1\}^c$.
- Add (x, y) to H-List.
- Return y.

– Parse x as R, $\textsc{Cmt}^*$, $\textsc{Rsp}$.
– Send $\textsc{Cmt}^*$ to the verifier and get back $\textsc{Ch}^*$.
– Add $(R||\textsc{Cmt}^*||\textsc{Ch}^*)$ to H-List.
– Return C_H^*.

$\mathsf{Sign}^{\mathcal{GS}}(i, m)$

– Maintain S-List (i, m, σ).
– When $\mathcal{A}$ queries (i, m) then $hs \leftarrow hs + 1$.
– $\mathcal{T} \leftarrow \mathsf{Exec}(i)$.
– Parse $\mathcal{T}$ as $(\textsc{Cmt}, \textsc{Ch}, \textsc{Rsp})$.
– $R \leftarrow \{0,1\}^{s(k)}$.
– Patch $\textsc{Ch} \leftarrow H(R\|\textsc{Cmt}\|m)$.
– $\sigma \leftarrow (R, \textsc{Cmt}, \textsc{Rsp})$.
– Return σ.

Finally, $\mathcal{A}$ outputs a forgery $(R, \textsc{Cmt}, \textsc{Rsp})$. If the tuple $(i, m, (R, \textsc{Cmt}, \textsc{Rsp}))$ is not in the S-List, algorithm $\mathcal{B}$ sends $\textsc{Rsp}$ to the verifier as the response. Let j be that $Q[j] = R||\textsc{Cmt}||m$. If $j = fp$, then the transcript between $\mathcal{B}$ and the verifier is $\textsc{Cmt}||\textsc{Ch}^*||\textsc{Rsp}$ (In this case, $\textsc{Ch} = \textsc{Ch}^*$). If $\mathcal{B}$ does not abort during the simulation, then the algorithm $\mathcal{A}$'s view is identical to its view in the real attack. Suppose F be the event that $\mathcal{A}$ wins the non-frameability game. S be the event that $hc = fp$ and $H(Q[fp]) = H(R||\textsc{Cmt}^*||m)$ is already defined when querying sign oracle. The advantage of $\mathcal{B}$ can be bounded by

$$\mathsf{Adv}^{non\text{-}frame}_{\mathcal{GID},\mathcal{B}} \geq \Pr[F \wedge \neg S] \geq \Pr[F](1 - Pr[S]) \geq \Pr[F] - \Pr[S]$$

The probability of F in the $i-th$ signature query is at most

$$\frac{1}{q_h} \cdot \frac{q_h + (i-1)}{2^{s(k)+\beta(k)}},$$

because $\mathcal{A}$ has made q_h hash queries, and $(i-1)$ times signing queries. So,

$$\Pr[F] \leq \sum_{i=1}^{q_s} \frac{1}{q_h} \cdot \frac{q_h + (i-1)}{2^{s(k)+\beta(k)}} = \frac{1}{q_h} \cdot \frac{q_s q_h + q_s(q_s-1)/2}{2^{s(k)+\beta(k)}} \leq \frac{1}{q_h} \cdot \frac{q_s(q_h + q_s - 1)}{2^{s(k)+\beta(k)}}.$$

We now bound the advantage of $\mathcal{B}$,

$$\mathsf{Adv}^{non\text{-}frame}_{\mathcal{GID},\mathcal{B}} \geq Pr[F] - \Pr[S] \geq \mathsf{Adv}^{non\text{-}frame}_{\mathcal{GS},\mathcal{A}}(k) - \frac{1}{q_h} \cdot \frac{q_s(q_h + q_s - 1)}{2^{s(k)+\beta(k)}}.$$

By combining Lemmas 1, 2, 3 and 4, **Theorem 1** is proved.

5.2 Proof of Theorem 2

The idea behind the proof for **Theorem 2** is as follow: if $\mathcal{GID}$ is insecure (i.e., there exists an algorithm $\mathcal{A}$ which can break the security of $\mathcal{GID}$ with non-negligible advantage), then there exists a algorithm $\mathcal{B}$ which can break the security of $\mathcal{GS} = FS(\mathcal{GID})$ with non-negligible advantage. We now prove **Theorem 2** via Lemma 5 to Lemma 8.

Lemma 5. *Let $\mathcal{GID}$ be a group identification scheme and $\mathcal{GS} = FS(\mathcal{GID})$. Let $\mathcal{A}$ be an adversary attacking the correctness of the group identification in the random oracle model. Then there is an adversary $\mathcal{B}$ against the correctness of $\mathcal{GS}$ such that $\mathsf{Adv}^{corr}_{\mathcal{GS},\mathcal{B}} \geq \mathsf{Adv}^{corr}_{\mathcal{GID},\mathcal{A}}$.*

Proof. Assume $\mathcal{A}$ is an algorithm attacking the correctness of $\mathcal{GID}$ and $\mathcal{B}$ is an algorithm against correctness of $\mathcal{GS}$. The goal of $\mathcal{B}$ is to use $\mathcal{A}$ to gain advantage when it runs $\mathsf{Exp}^{corr}_{\mathcal{GS},\mathcal{B}}(k)$ and accesses the associated oracle $\mathsf{AddU}^{\mathcal{GS}}(i, M)$. To achieve this goal, $\mathcal{B}$ should run the algorithm $\mathcal{A}$, simulate the environment of $\mathcal{A}$ in $\mathsf{Exp}^{corr}_{\mathcal{GID},\mathcal{A}}(k)$ with the $\mathsf{AddU}^{\mathcal{GS}}(i, M)$ oracle in $\mathsf{Exp}^{corr}_{\mathcal{GS},\mathcal{B}}(k)$. We now construct the algorithm $\mathcal{B}$ running $\mathcal{A}$ to gain advantage against $\mathcal{GS}$. Suppose $\mathcal{B}$ plays the correctness game, runs $\mathsf{Exp}^{corr}_{\mathcal{GS},\mathcal{B}}(k)$ and accesses the oracle of $\mathsf{AddU}^{\mathcal{GS}}(i, M)$. First, $\mathcal{B}$ runs $\mathcal{A}$ and simulates the oracle for $\mathcal{A}$. Then $\mathcal{B}$ just needs to sets $\mathsf{AddU}^{\mathcal{GID}}(i, M) = \mathsf{AddU}^{\mathcal{GS}}(i, M)$. Therefore, we can easily have $\mathsf{Adv}^{corr}_{\mathcal{GS},\mathcal{B}} \geq \mathsf{Adv}^{corr}_{\mathcal{GID},\mathcal{A}}$.

Lemma 6. *Let $\mathcal{GID}$ be a group identification scheme and $\mathcal{GS} = FS(\mathcal{GID})$. Let $\mathcal{A}$ be an adversary attacking the anonymity of the group identification. Then there is an adversary $\mathcal{B}$ against the anonymity of $\mathcal{GS}$ such that $\mathsf{Adv}^{anon}_{\mathcal{GS},\mathcal{B}} \geq \mathsf{Adv}^{anon}_{\mathcal{GID},\mathcal{A}}$.*

Proof. Assume $\mathcal{A}$ is an algorithm attacking the anonymity of $\mathcal{GID}$ and $\mathcal{B}$ is an algorithm against anonymity of $\mathcal{GS}$. The goal of $\mathcal{B}$ is to use $\mathcal{A}$ to gain advantage when it runs $\mathsf{Exp}^{anon\text{-}b}_{\mathcal{GS},\mathcal{B}}(k)$ and accesses the associated oracles. To achieve this goal, $\mathcal{B}$ should run the algorithm $\mathcal{A}$, simulate the environment of $\mathcal{A}$ in $\mathsf{Exp}^{anon\text{-}b}_{\mathcal{GID},\mathcal{A}}(k)$ with the oracles in $\mathsf{Exp}^{anon\text{-}b}_{\mathcal{GS},\mathcal{B}}(k)$. We now construct the algorithm $\mathcal{B}$ running $\mathcal{A}$ to gain advantage against $\mathcal{GS}$. Suppose $\mathcal{B}$ plays the anonymity game, runs $\mathsf{Exp}^{anon\text{-}b}_{\mathcal{GS},\mathcal{B}}(k)$ and accesses the oracles of $\mathsf{SndToU}^{\mathcal{GS}}(i, M)$, $\mathsf{SndToI}^{\mathcal{GS}}(i, M)$, $\mathsf{CrptU}^{\mathcal{GS}}(i, \mathfrak{pk})$, $\mathsf{USK}^{\mathcal{GS}}(i)$, $\mathsf{Open}^{\mathcal{GS}}(\sigma, m)$ and $\mathsf{CH}^{\mathcal{GS}}_b(i_0, i_1, m)$. First, $\mathcal{B}$ runs $\mathcal{A}$ and simulates the oracles for $\mathcal{A}$. Then $\mathcal{B}$ sets $\mathsf{SndToU}^{\mathcal{GS}}(i, M)$, $\mathsf{SndToI}^{\mathcal{GS}}(i, M)$, $\mathsf{CrptU}^{\mathcal{GS}}(i, \mathfrak{pk})$, and $\mathsf{USK}^{\mathcal{GS}}(i)$ to their equivalent oracles in the $\mathcal{GID}$ game. Then $\mathcal{B}$ constructs $\mathsf{Open}^{\mathcal{GID}}(\mathcal{T})$ oracle for $\mathcal{A}$ as follow:

$\mathsf{Open}^{\mathcal{GID}}(\mathcal{T})$

- Parse $\mathcal{T}$ as Cmt, Ch, Rsp.
- Generate a message m and a random R.
- $\text{Ch} \leftarrow H(R \| \text{Cmt} \| m)$.
- $\sigma \leftarrow (R, \text{Cmt}, \text{Rsp})$
- Return $\mathsf{Open}^{\mathcal{GS}}(\sigma, m)$.

When $\mathcal{A}$ makes one of the above oracles queries, Algorithm $\mathcal{B}$ answers $\mathcal{A}$ with its own queries according the above simulation. Then $\mathcal{A}$ makes one challenge oracle query, which $\mathcal{B}$ answers by calling its oracle and return a transcript according to the following $\mathsf{CH}_b^{\mathcal{GID}}(i_0, i_1)$ simulation:

- $\sigma \leftarrow \mathsf{CH}_b^{\mathcal{GS}}(i_0, i_1, m)$.
- Parse σ as R^*, $\textsc{Cmt}^*$, $\textsc{Rsp}^*$.
- $\textsc{Ch}^* \leftarrow H(R^* \| \textsc{Cmt}^* \| m)$.
- $\mathcal{T} \leftarrow (\textsc{Cmt}^*, \textsc{Ch}^*, \textsc{Rsp}^*)$.
- Return $\mathcal{T}$.

Finally, $\mathcal{A}$ outputs a decision bit d for the Experiment $\mathsf{Exp}^{anon\text{-}b}_{\mathcal{GID},\mathcal{A}}(k)$. Algorithm $\mathcal{B}$ returns d as the answer to its own challenge for the Experiment $\mathsf{Exp}^{anon\text{-}b}_{\mathcal{GS},\mathcal{B}}(k)$. Therefore, we have $\mathsf{Adv}^{anon}_{\mathcal{GS},\mathcal{B}} \geq \mathsf{Adv}^{anon}_{\mathcal{GID},\mathcal{A}}$

Lemma 7. *Let $\mathcal{GID}$ be a group identification scheme and $\mathcal{GS} = FS(\mathcal{GID})$. Let $\mathcal{A}$ be an adversary attacking the traceability of the group identification. Then there is an adversary $\mathcal{B}$ against the traceability of $\mathcal{GS}$ such that $\mathsf{Adv}^{trace}_{\mathcal{GS},\mathcal{B}} \geq \mathsf{Adv}^{trace}_{\mathcal{GID},\mathcal{A}}$.*

Proof. Let $\mathcal{B}$ be an algorithm attacking traceability of $\mathcal{GS}$ by running algorithm $\mathcal{A}$ and simulating the $\mathsf{Exp}^{trace}_{\mathcal{GID},\mathcal{A}}(k)$ environment for $\mathcal{A}$. We assume that $\mathcal{B}$ has access to the oracles $\mathsf{AddU}^{\mathcal{GS}}(i, M)$, $\mathsf{SndToI}^{\mathcal{GS}}(i, M)$, $\mathsf{CrptU}^{\mathcal{GS}}(i, \mathfrak{pk})$ and $\mathsf{USK}^{\mathcal{GS}}(i)$ for the $\mathcal{GS}$ game. We now construct the algorithm $\mathcal{B}$ as follows: $\mathcal{B}$ begins with the initialization: $m \leftarrow 0$. Next, $\mathcal{B}$ runs $\mathcal{A}$ and simulates the following oracles for $\mathcal{A}$: $\mathsf{AddU}^{\mathcal{GS}}(i, M) = \mathsf{AddU}^{\mathcal{GID}}(i, M)$, $\mathsf{SndToI}^{\mathcal{GS}}(i, M) = \mathsf{SndToI}^{\mathcal{GID}}(i, M)$, $\mathsf{CrptU}^{\mathcal{GS}}(i, \mathfrak{pk}) = \mathsf{CrptU}^{\mathcal{GID}}(i, \mathfrak{pk})$, and $\mathsf{USK}^{\mathcal{GS}}(i) = \mathsf{USK}^{\mathcal{GID}}(i)$. When $\mathcal{A}$ makes above oracle queries, $\mathcal{B}$ answers by calling its own appropriate queries. When $\mathcal{A}$ outputs a commitment $\textsc{Cmt}$, $\mathcal{B}$ increases $m \leftarrow m + 1$, selects a random R and then sets $\textsc{Ch} \leftarrow H(R \| \textsc{Cmt} \| m)$. Next, $\mathcal{B}$ sends $\textsc{Ch}$ to $\mathcal{A}$, and $\mathcal{A}$ finally outputs a response $\textsc{Rsp}$. Algorithm $\mathcal{B}$ sets $\sigma \leftarrow (R, \textsc{Cmt}, \textsc{Rsp})$ and then outputs (σ, m). Since the messages in the algorithm are always new, the forgery has never been queried to the signing oracle in the past. Therefore, we have $\mathsf{Adv}^{trace}_{\mathcal{GS},\mathcal{B}} \geq \mathsf{Adv}^{trace}_{\mathcal{GID},\mathcal{A}}$.

Lemma 8. *Let $\mathcal{GID}$ be a group identification scheme and $\mathcal{GS} = FS(\mathcal{GID})$. Let $\mathcal{A}$ be an adversary attacking the non-frameability of the group identification. Then there is an adversary $\mathcal{B}$ against the non-frameability of $\mathcal{GS}$ such that $\mathsf{Adv}^{non\text{-}frame}_{\mathcal{GS},\mathcal{B}} \geq \mathsf{Adv}^{non\text{-}frame}_{\mathcal{GID},\mathcal{A}}$.*

Proof. Let $\mathcal{B}$ be an algorithm attacking non-frameability of GS by running algorithm $\mathcal{A}$ and simulating the $\mathsf{Exp}^{non\text{-}frame}_{\mathcal{GID},\mathcal{A}}(k)$ environment for $\mathcal{A}$. Suppose $\mathcal{B}$ accesses the oracles of $\mathsf{SndToU}^{\mathcal{GS}}(i, M)$, $\mathsf{CrptU}^{\mathcal{GS}}(i, \mathfrak{pk})$ and $\mathsf{USK}^{\mathcal{GS}}(i)$, $\mathsf{USK}^{\mathcal{GS}}(i)$ and $\mathsf{Sign}^{\mathcal{GS}}(i, m)$. $\mathcal{B}$ begins with the initialization: $m \leftarrow 0$. Next, $\mathcal{B}$ runs $\mathcal{A}$ and sets $\mathsf{SndToU}^{\mathcal{GS}}(i, M) = \mathsf{SndToU}^{\mathcal{GID}}(i, M)$, $\mathsf{CrptU}^{\mathcal{GS}}(i, \mathfrak{pk}) = \mathsf{CrptU}^{\mathcal{GID}}(i, \mathfrak{pk})$, and $\mathsf{USK}^{\mathcal{GS}}(i) = \mathsf{USK}^{\mathcal{GID}}(i)$.

When $\mathcal{A}$ makes above oracle queries, $\mathcal{B}$ answers by calling its own appropriate queries. When $\mathcal{A}$ queries transcript oracles, $\mathcal{B}$ constructs $\mathsf{Exec}(i)$ for $\mathcal{A}$ as follow:

$\mathsf{Exec}(i)$
- $m \leftarrow m + 1$.
- $\sigma \leftarrow \mathsf{Sign}(i, m)$.
- Parse σ as $(R^*, \textsc{Cmt}^*, \textsc{Rsp}^*)$.
- $\textsc{Ch}^* \leftarrow H(R^* \| \textsc{Cmt}^* \| m)$.
- $\mathcal{T} \leftarrow (\textsc{Cmt}^*, \textsc{Ch}^*, \textsc{Rsp}^*)$.
- Return $\mathcal{T}$.

When $\mathcal{A}$ outputs a commitment $\textsc{Cmt}$, $\mathcal{B}$ increases $m \leftarrow m + 1$, selects a random R, and sets $\textsc{Ch} \leftarrow H(R \| \textsc{Cmt} \| m)$. Then, $\mathcal{B}$ sends $\textsc{Ch}$ to $\mathcal{A}$ and gets back a response $\textsc{Rsp}$. Finally, $\mathcal{B}$ sets $\sigma \leftarrow (R, \textsc{Cmt}, \textsc{Rsp})$ and outputs (σ, m). Since the messages in the algorithm are always new, the forgery has never been queried to the signing oracle in the past. Therefore, we have $\mathsf{Adv}^{non\text{-}frame}_{\mathcal{GS},\mathcal{B}} \geq \mathsf{Adv}^{non\text{-}frame}_{\mathcal{GID},\mathcal{A}}$.

By combining Lemmas 5, 6, 7 and 8, **Theorem 2** is proved.

6 From Ad-hoc Group Identification to Ring Signatures

An ad hoc group identification scheme is an identification protocol in which a prover can anonymously prove she is a valid number of an ad hoc group. Based on the underlying PKI, arbitrary ad hoc groups of a user population can be formed without the help of a group manager. In [16], the authors give a formal model of an ad hoc identification scheme which is a six-tuple of algorithms $(\mathsf{Setup}, \mathsf{Register}, \mathsf{Make\text{-}GPK}, \mathsf{Make\text{-}GSK}, \mathsf{Anon\text{-}ID}^{\mathsf{P}}, \mathsf{Anon\text{-}ID}^{\mathsf{V}})$. However, in this paper, we slightly modify the notations of the model of [16] in order to suit with the model and notations of our ring signature $\mathcal{RS}$.

An ad hoc group identification scheme is given by the tuple of algorithms $\mathcal{AHID} = (\mathsf{UKg}^{\mathcal{AHID}}, \mathsf{GPKg}^{\mathcal{AHID}}, \mathsf{GSKg}^{\mathcal{AHID}}, (\mathsf{P}^{\mathcal{AHID}}, \mathsf{V}^{\mathcal{AHID}}))$. The functionality of these algorithms is as follows:

- $\mathsf{UKg}^{\mathcal{AHID}}$: This is a probabilistic algorithm to generate user public/private key pairs. When run by user i, on input of 1^k, this outputs a user's key pair $(\mathfrak{sk}_i, \mathfrak{pk}_i)$.
- $(\mathsf{GPKg}^{\mathcal{AHID}}, \mathsf{GSKg}^{\mathcal{AHID}})$: The ad hoc group public key generation algorithm and the ad hoc group secret key generation algorithm. The algorithm $\mathsf{GPKg}^{\mathcal{AHID}}$ is a deterministic algorithm which combines a set of user public keys S into a single ad hoc group public key $\mathfrak{gpk}$. The deterministic algorithm $\mathsf{GSKg}^{\mathcal{AHID}}$ takes as input a user secret/public key pair $(\mathfrak{sk}_i, \mathfrak{pk}_i)$ and a set of user public keys S, it outputs an ad hoc group secret key $\mathfrak{gsk}_i$ which associates with the ad hoc group public key $\mathfrak{gpk}$.
- $(\mathsf{P}^{\mathcal{AHID}}, \mathsf{V}^{\mathcal{AHID}})$: An interactive protocol between a prover and a verifier. The prover's input is a value $\mathfrak{gsk}_i$, whereas the verifier's input is $\mathfrak{gpk}$.

As before we shall focus purely on canonical ad hoc group identification schemes, where the $(\mathsf{P}^{\mathcal{AHID}}, \mathsf{V}^{\mathcal{AHID}})$ protocol is given by three-move protocol of the commit-challenge-response variety.

We now present a security model for such ad-hoc group identification schemes, and an analogous model for ring signatures. We then formalise the construction of ring signatures from ad-hoc group identification schemes via the Fiat–Shamir transform. Finally, an analogous theorem to the earlier one can be proved, namely:

Theorem 3. *Let $\mathcal{AHID}$ be a canonical ad-hoc group identification scheme and $\mathcal{RS} = FS(\mathcal{AHID})$. The derived ring signature scheme $\mathcal{RS}$ has the properties of correctness, anonymity and unforgeability against chosen-message attacks if and only if $\mathcal{AHID}$ has the properties of correctness, anonymity, and non-impersonation under passive attacks.*

6.1 Security of Ad hoc Canonical Group Identification Scheme

We now present notions of security for canonical ad hoc group identification schemes, which we call anonymity and non-impersonation, under passive attacks. Before doing so we first define some oracles, in Figure 5, which will be used by our adversaries in attacking ad hoc canonical group identification schemes. All oracles (and the underlying experiments) maintain the following global variables: a set HU of honest users, a set CU of corrupted users, a set S of an arbitrary ad hoc group public keys set, and a set TL of transcripts, all of which are assumed to be initially empty.

Using these oracles we can now define our security notions and correctness notions for canonical ad hoc group identification scheme. This is done via the experiments in Figure 6. We note that we only require security under passive attacks for our application, i.e. the attacker can obtain valid transcripts, but is not able to interact with individual provers. Hence, security is defined for this restricted notion of attack, the generalisation to active attacks is obvious.

Correctness: We require that transcripts produced by honest users should be accepted by the verifiers. We define

$$\mathsf{Adv}^{corr}_{\mathcal{AHID},\mathcal{A}}(k) = \Pr[\mathsf{Exp}^{corr}_{\mathcal{AHID},\mathcal{A}}(k) = 1],$$

and we say that the scheme is *correct* if $\mathsf{Adv}^{corr}_{\mathcal{AHID},\mathcal{A}}(k) = 0$ for all adversaries $\mathcal{A}$ and all $k \in \mathbb{N}$.

Anonymity: Let $\mathcal{A}$ be an adversary performing anonymity experiment given below for $b \in \{0, 1\}$. The goal of the adversary is to determine which of two identities has engaged in a run of the identification protocol. We define

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

and we say that the scheme has *anonymity* if $\mathsf{Adv}^{anon}_{\mathcal{AHID},\mathcal{A}}(k)$ is a negligible function of k for any polynomial time adversary $\mathcal{A}$.

$\mathsf{AddU}(i)$:
- If $i \in \mathsf{HU} \cup \mathsf{CU}$ then return $\perp$.
- $\mathsf{HU} \leftarrow \mathsf{HU} \cup \{i\}$.
- $(\mathfrak{sk}_i, \mathfrak{pk}_i) \leftarrow \mathsf{UKg}(1^k)$.
- Return $(\mathfrak{sk}_i, \mathfrak{pk}_i)$.

$\mathsf{Exec}(i, S)$:
- If $i \notin \mathsf{HU}$ or $\mathfrak{pk}_i \notin S$ then return $\perp$
- $\mathfrak{gsk}_i \leftarrow \mathsf{GSKg}^{\mathcal{AHID}}(\mathfrak{sk}_i, \mathfrak{pk}_i, S)$.
- $R \leftarrow \mathsf{Coins}(\mathsf{P}^{\mathcal{AHID}})$
- $\textsc{Cmt} \leftarrow \mathsf{P}^{\mathcal{AHID}}(\mathfrak{gsk}_i; R)$
- $\textsc{Ch} \leftarrow \{0,1\}^c$
- $\textsc{Rsp} \leftarrow \mathsf{P}^{\mathcal{AHID}}(\mathfrak{gsk}_i, \textsc{Cmt}, \textsc{Ch}, R)$
- $\mathcal{T} \leftarrow (\textsc{Cmt}, \textsc{Ch}, \textsc{Rsp})$.
- Return $\mathcal{T}$

$\mathsf{CrptU}(i, \mathfrak{pk})$:
- If $i \in \mathsf{HU} \cup \mathsf{CU}$ then return $\perp$.
- $\mathsf{CU} \leftarrow \mathsf{CU} \cup \{i\}$.
- Return 1.

$\mathsf{USK}(i, S)$:
- If $\mathfrak{pk}_i \notin S$ then return $\perp$.
- $\mathfrak{gpk} \leftarrow \mathsf{GPKg}^{\mathcal{AHID}}(S)$.
- $\mathfrak{gsk}_i \leftarrow \mathsf{GSKg}^{\mathcal{AHID}}(\mathfrak{sk}_i, \mathfrak{pk}_i, S)$.
- Return $(\mathfrak{gsk}_i, \mathfrak{gpk}, \mathfrak{sk}_i)$.

$\mathsf{CH}_b(i_0, i_1, S)$:
- If $i_0 \notin \mathsf{HU}$ or $\mathfrak{sk}_{i_0} \notin S$ then return $\perp$.
- If $i_1 \notin \mathsf{HU}$ or $\mathfrak{sk}_{i_1} \notin S$ then return $\perp$.
- $\mathcal{T} \leftarrow \mathsf{Exec}(i_b, S)$.
- $\mathsf{TL} \leftarrow \mathsf{TL} \cup \{(S, \mathcal{T})\}$.
- Return $\mathcal{T}$.

Fig. 5. Oracles required to define security for canonical ad hoc group identification schemes

Non-impersonation: Let $\mathcal{A}$ be an adversary performing non-impersonation experiment given below. The goal of the adversary is to produce a valid transcript belongs to an ad hoc group, however, $\mathcal{A}$ is not in the group. We define

$$\mathsf{Adv}^{non\text{-}imp}_{\mathcal{AHID},\mathcal{A}}(k) = \Pr[\mathsf{Exp}^{non\text{-}imp}_{\mathcal{AHID},\mathcal{A}}(k) = 1],$$

and we say that the scheme is *secure against impersonation* if $\mathsf{Adv}^{non\text{-}imp}_{\mathcal{AHID},\mathcal{A}}(k)$ is a negligible function of k for any polynomial time adversary $\mathcal{A}$.

6.2 Ring Signature

A ring signature scheme has the form $\mathcal{RS} = (\mathsf{UKg}^{\mathcal{RS}}, \mathsf{RPKg}^{\mathcal{RS}}, \mathsf{RSKg}^{\mathcal{RS}}, \mathsf{RSig}, \mathsf{RVf})$. The functionality the algorithms $\mathsf{UKg}^{\mathcal{RS}}, \mathsf{RPKg}^{\mathcal{RS}}$ and $\mathsf{RSKg}^{\mathcal{RS}}$ are identical to those for the ad hoc group identification schemes considered earlier. However, the prover and verifier interactive algorithms have now been replaced with a signing algorithm RSig and a verification algorithm RVf. In particular we have:

- GSig: Is a probabilistic signing algorithm taking input a ring signing key $\mathfrak{gsk}_i$ and a message m, returning a signature σ.
- RVf: Is a deterministic verifying algorithm which takes input the ring public key $\mathfrak{gpk}$, a group signature σ and a message m. It then returns a Boolean decision to demonstrate whether the group signature is accepted or rejected.

We can define the properties of correctness, anonymity almost the same as for ad hoc identification schemes. Unforgeability under chosen-message attacks

Experiment $\mathsf{Exp}^{corr}_{\mathcal{AHID},\mathcal{A}}(k)$
- $\mathsf{CU}, \mathsf{HU} \leftarrow \emptyset$.
- $(i, S) \leftarrow \mathcal{A}(\mathsf{AddU}(\cdot))$.
- If $i \notin \mathsf{HU}$ or $\mathfrak{sk}_i \notin S$ then return 0.
- $\mathcal{T} \leftarrow \mathsf{Exec}(i, S)$.
- $\mathfrak{gpk} \leftarrow \mathsf{GPKg}^{\mathcal{AHID}}(S)$.
- If $\mathsf{V}^{\mathcal{AHID}}(\mathfrak{gpk}, \mathcal{T}) = 0$ then return 1.
- Return 0.

Experiment $\mathsf{Exp}^{anon\text{-}b}_{\mathcal{AHID},\mathcal{A}}(k)$
- $\mathsf{CU}, \mathsf{HU}, \mathsf{TL} \leftarrow \emptyset$.
- $d \leftarrow \mathcal{A}(\mathsf{CrptU}(\cdot,\cdot), \mathsf{USK}(\cdot), \mathsf{Exec}(\cdot,\cdot), \mathsf{CH}_b(\cdot,\cdot))$.
- Return d.

Experiment $\mathsf{Exp}^{non\text{-}imp}_{\mathcal{AHID},\mathcal{A}}(k)$
- $\mathsf{CU}, \mathsf{HU} \leftarrow \emptyset$.
- $(\textsc{Cmt}, S, \mathsf{state}) \leftarrow \mathcal{A}_1(\mathsf{CrptU}(\cdot,\cdot), \mathsf{USK}(\cdot))$.
- $\textsc{Ch} \leftarrow \{0,1\}^c$.
- $\textsc{Rsp} \leftarrow \mathcal{A}_2(\textsc{Ch}, \mathsf{state} : \mathsf{CrptU}(\cdot,\cdot), \mathsf{USK}(\cdot))$.
- $\mathcal{T} \leftarrow (\textsc{Cmt}, \textsc{Ch}, \textsc{Rsp})$.
- $\mathfrak{gpk} \leftarrow \mathsf{GPKg}^{\mathcal{AHID}}(S)$.
- If $\mathsf{V}^{\mathcal{AHID}}(\mathfrak{gpk}, \mathcal{T}) = 0$ then return 0.
- If the following are all true then return 1 else return 0.
  - $i \in \mathsf{HU}$ and $\mathfrak{gsk}_i \neq \perp$.
  - $\mathcal{A}$ did not query $\mathsf{USK}(i)$ and $\mathcal{T}$ was not produced by a call to $\mathsf{Exec}(i, S)$.

Fig. 6. Security experiments for canonical ad-hoc group identification schemes

is also like the non-impersonation for group identification schemes. Firstly, the oracles are changed in the following ways:

- The $\mathsf{Exec}(i, S)$ oracle is now replaced by a signature oracle $\mathsf{Sign}(i, m, S)$
- In CH_b and the game for anonymity we replace the list of transcripts TL by a list of signatures SL issued by the oracle CH_b. The oracle CH_b now also takes an additional input, which is an adversarially chosen message m.
- We need a hash oracle H which is the same as in the game for group signatures.

Secondly, the games are slightly changed as we no longer are talking about an interactive protocol and we need to sign a message. In particular the experiments become as in Figure 7, with the advantages being defined in the obvious manner.

6.3 Construction from Ad Hoc Group Identification to Ring Signatures

We now construct a ring signature scheme from a canonical ad hoc group identification scheme, using the generalised Fiat–Shamir transform. Let $\mathcal{AHID} =$

Experiment $\mathsf{Exp}^{corr}_{\mathcal{RS},\mathcal{A}}(k)$
- $\mathsf{CU}, \mathsf{HU} \leftarrow \emptyset$.
- $(i, m, S) \leftarrow \mathcal{A}(\mathsf{AddU}(\cdot))$.
- If $i \notin \mathsf{HU}$ or $\mathfrak{sk}_i \notin S$ then return 0.
- $\sigma \leftarrow \mathsf{Sign}(i, m, S)$.
- $\mathfrak{gpk} \leftarrow \mathsf{GPKg}^{\mathcal{AHID}}(S)$.
- If $\mathsf{RVf}\mathcal{RS}(\mathfrak{gpk}, \sigma, m) = 0$ then return 1.
- Return 0.

Experiment $\mathsf{Exp}^{anon\text{-}b}_{\mathcal{RS},\mathcal{A}}(k)$
- $\mathsf{CU}, \mathsf{HU}, \mathsf{TL} \leftarrow \emptyset$.
- $d \leftarrow \mathcal{A}(\mathsf{H}(\cdot), \mathsf{CrptU}(\cdot,\cdot), \mathsf{USK}(\cdot), \mathsf{Sign}(\cdot,\cdot,\cdot), \mathsf{CH}_b(\cdot,\cdot,\cdot))$.
- Return d.

Experiment $\mathsf{Exp}^{uf\text{-}cma}_{\mathcal{RS},\mathcal{A}}(k)$
- $\mathsf{CU}, \mathsf{HU} \leftarrow \emptyset$.
- $(\sigma, m, S) \leftarrow \mathcal{A}(\mathsf{H}(\cdot), \mathsf{CrptU}(\cdot,\cdot), \mathsf{USK}(\cdot), \mathsf{Sign}(\cdot,\cdot,\cdot))$.
- $\mathfrak{gpk} \leftarrow \mathsf{GPKg}^{\mathcal{AHID}}(S)$.
- If $\mathsf{RVf}(\mathfrak{gpk}, \sigma, m) = 0$ then return 0.
- If the following are all true then return 1 else return 0.
  - $i \in \mathsf{HU}$ and $\mathfrak{gsk}_i \neq \perp$.
  - $\mathcal{A}$ did not query $\mathsf{USK}(i)$ and σ was not produced by a call to $\mathsf{Sign}(i, m, S)$.

Fig. 7. Security experiments for ring signature schemes

$(\mathsf{UKg}^{\mathcal{AHID}}, \mathsf{GPKg}^{\mathcal{AHID}}, \mathsf{GSKg}^{\mathcal{AHID}}, (\mathsf{P}^{\mathcal{AHID}}, \mathsf{V}^{\mathcal{AHID}})$ be a canonical ad hoc group identification scheme, and $s : \mathbb{N} \to \mathbb{N}$ be a function which defines a *seed length* $s(k)$ given the security parameter k. We select a hash function $H : \{0,1\}^* \to \{0,1\}^c$ at random from the set of all maps $\{0,1\}^* \to \{0,1\}^c$, where c is the bit length of the challenge CH in the canonical group identification scheme we will be using. From these we construct ring signature scheme $\mathcal{RS} = (\mathsf{UKg}^{\mathcal{RS}}, \mathsf{RPKg}^{\mathcal{RS}}, \mathsf{RSKg}^{\mathcal{RS}}, \mathsf{RSig}, \mathsf{RVf})$ as follows. We let $\mathsf{UKg}^{\mathcal{RS}} = \mathsf{UKg}^{\mathcal{AHID}}$, $\mathsf{RPKg}^{\mathcal{RS}} = \mathsf{GPKg}^{\mathcal{AHID}}$, $\mathsf{RSKg}^{\mathcal{RS}} = \mathsf{GSKg}^{\mathcal{AHID}}$. Then the functions RSig and RVf are defined as in Figure 8

The Security of the Construction The security proof of the construction is very similar to the one of $\mathcal{GID}$-to-$\mathcal{GS}$ given in the main body of the paper. The main differences are that the $\mathcal{AHID}/\mathcal{RS}$ algorithms are setup-free and there are no opening and judge oracles in $\mathcal{AHID}/\mathcal{RS}$ models. Moreover, the oracles restrictions are almost the same between the security game for $\mathcal{AHID}$ and the security game for $\mathcal{RS}$. Hence, we do not present the proof in detail as it can be derived as an easy exercise for the reader.

$\mathsf{RSig}(\mathfrak{gsk}_i, m)$:
- $R_P \leftarrow \mathsf{Coins}(\mathsf{P}^{\mathcal{AHID}})$.
- $\textsc{Cmt} \leftarrow \mathsf{P}^{\mathcal{AHID}}(\mathfrak{gsk}_i; R_P)$.
- $R \leftarrow \{0,1\}^{s(k)}$.
- $\textsc{Ch} \leftarrow H(R \| \textsc{Cmt} \| m)$.
- $\textsc{Rsp} \leftarrow \mathsf{P}^{\mathcal{AHID}}(\mathfrak{gsk}_i, \textsc{Cmt}, \textsc{Ch}, R_P)$.
- $\sigma \leftarrow (R, \textsc{Cmt}, \textsc{Rsp})$.
- Return σ.

$\mathsf{RVf}(\mathfrak{gpk}, \sigma, m)$:
- Parse σ as $(R, \textsc{Cmt}, \textsc{Rsp})$.
- $\textsc{Ch} \leftarrow H(R \| \textsc{Cmt} \| m)$.
- $\mathcal{T} \leftarrow (\textsc{Cmt}, \textsc{Ch}, \textsc{Rsp})$.
- Return $\mathsf{V}^{\mathcal{AHID}}(\mathfrak{gpk}, \mathcal{T})$.

Fig. 8. Construction of a ring signature from a canonical ad hoc identification scheme

7 Acknowledgements

The work in this paper was partially funded by the European Commission through the ICT programme under contract ICT-2007-216676 ECRYPT II FP7. The second author was supported by a Royal Society Wolfson Merit Award and a grant from Google.

References

1. M. Abdalla, J.H. An, M. Bellare and C. Namprempre. From identification to signatures via the Fiat–Shamir transform: Minimizing assumptions for security and forward-security. In *Advances in Cryptology – Eurocrypt 2002*, Springer-Verlag LNCS 2332, 418–433, 2002.
2. G. Ateniese, J. Camenisch, M. Joye and G. Tsudik. A practical and provably secure coalition-resistant group signature scheme. In *Advances in Cryptology – Crypto 2000*, Springer-Verlag LNCS 1880, 255–270, 2000.
3. G. Ateniese and B. de Medeiros. Efficient group signatures without trapdoors. In *Advances in Cryptology – Asiacrypt 2002*, Springer-Verlag LNCS 2894, 246–268, 2002.
4. M. Abe, M. Ohkubo and K. Suzuki. 1-out-of-n signatures from a variety of keys. In *Advances in Cryptology – Asiacrypt 2002*, Springer-Verlag LNCS 2501, 415–432, 2002.
5. D. Boneh, X. Boyen and H. Shacham. Short group signatures. In *Advances in Cryptology – Crypto 2004*, Springer-Verlag LNCS 3152, 41–55, 2004.
6. M. Bellare, D. Micciancio and B. Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In *Advances in Cryptology – Eurocrypt 2003*, Springer-Verlag LNCS 2656, 614–629, 2003.
7. M. Bellare, C. Namprempre and G. Neven. Security proofs for identity-based identification and signature schemes. In *Advances in Cryptology – Eurocrypt 2004*, Springer-Verlag LNCS 3027, 268–286, 2004
8. D. Boneh and H. Shacham. Group signatures with verifier-local revocation. In *Proceedings of ACM CCS 2004*, 168–177, 2004.
9. M. Bellare, H, Shi and C. Zhang. Foundations of group signatures: The case of dynamic groups. In *Topics in Cryptology – CT-RSA 2005*, Springer-Verlag LNCS 3376, 136–153, 2005.

10. X. Boyen and B. Waters. Compact group signatures without random oracles. In *Advances in Cryptology – Eurocrypt 2006*, Springer-Verlag LNCS 4004, 427–444, 2006.
11. X. Boyen, B. Waters Full-domain subgroup hiding and constant-size group signatures. In *Public Key Cryptography – PKC 2007*, Springer-Verlag LNCS 4450, 1–15, 2007.
12. J. Camenisch. Efficient and generalized group signatures. In *Advances in Cryptology – Eurocrypt 1997*, Springer-Verlag LNCS 1233, 465–479, 1997.
13. J. Camenisch and M. Michels. A group signature scheme with improved efficiency. In *Advances in Cryptology – Asiacrypt 1998*, Springer-Verlag LNCS 1514, 160–174, 1998.
14. L. Chen, and T.P. Pedersen. New group signature schemes (extended abstract). In *Advances in Cryptology – Eurocrypt 1994*, Springer-Verlag LNCS 950, 171–181, 1994.
15. J. Camenisch and M. Stadler. Efficient group signature schemes for large group. In *Advances in Cryptology – Crypto 1997*, Springer-Verlag LNCS 1294, 410–424, 1997.
16. Y. Dodis, A. Kiayias, A. Nicolosi and V. Shoup. Anonymous identification in ad hoc groups. In *Advances in Cryptology – Eurocrypt 2004*, Springer-Verlag LNCS 3027, 609–626, 2004.
17. C. Dwork, M. Naor, O. Reingold and L. Stockmayer. Magic Functions. *Journal of the ACM* , **50**, 852–921, 2003.
18. J. Furukawa and H. Imai. An efficient group signature scheme from bilinear maps. In *ACISP 2005*, Springer-Verlag LNCS 3574, 455–467, 2005.
19. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Advances in Cryptology – Eurocrypt 1990*, Springer-Verlag LNCS 473, 481–486, 1990.
20. J. Groth. Fully anonymous group signatures without random oracles. In *Advances in Cryptology – Asiacrypt 2007*, Springer-Verlag LNCS 4833, 164–180, 2007.
21. S. Goldwasser and Y. Tauman. On the (in)security of the Fiat-Shamir paradigm. In *IEEE Symposium on Foundations of Computer Science – FOCS 2003*, 102–115, 2003.
22. S. Han, J. Wang and W. Liu. An efficient identity-based group signature scheme over elliptic curves. In *Universal Multiservice Networks*, Springer-Verlag LNCS 3262, 417-429, 2007.
23. J. Kilian and E. Petrank. Identity escrow. In *Advances in Cryptology – Crypto 1998*, Springer-Verlag LNCS 1642, 169–185, 1998.
24. A. Kiayias and M. Yung. Group signatures with efficient concurrent join. In *Advances in Cryptology – Eurocrypt 2005*, Springer-Verlag LNCS 3494, 198–214, 2005.
25. X. Liang, Z. Cao, J. Shao and H. Lin. Short group signature without random oracles. In *Information and Communications Security – ICS 2007*, Springer-Verlag LNCS 4861, 69–82, 2007.
26. A. Miyaji and K. Umeda. A fully-functional group signature scheme over only known-order group. In *Applied Cryptography and Network Security – ACNS 2004*, Springer-Verlag LNCS 3089, 164–179, 2004.
27. T. Nakanishi and N. Funabiki. A short verifier-local revocation group signature scheme with backward unlinkability. In *Advances in Information and Computer Security*, Springer-Verlag LNCS 4266, 17–32, 2006.

28. T. Nakanishi, H. Fujii, Y. Hira and N. Funabiki. Revocable group signature schemes with constant costs for signing and verifying. In *Public Key Cryptography - PKC 2009*, Springer-Verlag LNCS 5443, 463–480, 2009.
29. T. Nakanishi, T. Fujiwara and H. Watanabe. A Linkable Group Signature and Its Application to Secret Voting. *Information Processing Society of Japan* , **40**, 3085–3096, 1999.
30. T. Nakanishi, F. Kubooka, N. Hamada and N. Funabiki. Group signature schemes with membership revocation for large groups. In *ACISP 2005*, Springer-Verlag LNCS 3574, 443–454, 2005.
31. T. Nakanishi and Y. Sugiyama. A group signature scheme with efficient membership revocation for reasonable groups. In *ACISP 2004*, Springer-Verlag LNCS 3108, 336–347, 2004.
32. K. Ohta and T. Okamoto. On concrete security treatment of signatures derived from identification. In *Advances in Cryptology – Crypto 1998*, Springer-Verlag LNCS 1462, 223–242, 1998.
33. D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. In *Journal of Cryptology*, **13**, 361–396, 2000.
34. V. K. Wei, T. H. Yuen and F. Zhang. Group signature where group manager, members and open authority are identity-based. In *Information Security and Privacy*, Springer-Verlag LNCS 3574, 468–480, 2005.
35. J. Zhang, J. Zou and Y. Wang. An improved group signature scheme. In *Trust, Privacy and Security in Digital Business*, Springer-Verlag LNCS 3592, 185–194, 2005.