Group Signatures with Efficient Concurrent Join

Aggelos Kiayias* Moti Yung†

Abstract

A group signature is a basic privacy mechanism. The group joining operation is a critical component of such a scheme. To date all secure group signature schemes either employed a trusted-party aided join operation or a complex joining protocol requiring many interactions between the prospective user and the Group Manager (GM). In addition no efficient scheme employed a join protocol proven secure against adversaries that have the capability to dynamically initiate multiple concurrent join sessions during an attack.

This work presents the first efficient group signature scheme with a simple Joining protocol that is based on a "single message and signature response" interaction between the prospective user and the GM. This single-message and signature-response registration paradigm where no other actions are taken, is the most efficient possible join interaction and was originally alluded to in 1997 by Camenisch and Stadler, but its efficient instantiation remained open till now.

The fact that joining has two short communication flows and does not require secure channels is highly advantageous: for example, it allows users to easily join by a proxy (i.e., a security officer of a company can send a file with all registration requests in his company and get back their certificates for distribution back to members of the company). It further allows an easy and non-interactive global system re-keying operation as well as straightforward treatment of multi-group signatures. We present a strong security model for group signatures (the first explicitly taking into account concurrent join attacks) and an efficient scheme with a single-message and signature-response join protocol.

The present manuscript is a full version containing proofs, minor corrections as well as a more flexible and efficient protocol construction compared to the proceedings version [28].

*Computer Science and Engineering Dept., University of Connecticut, Storrs, CT, USA, aggelos@cse.uconn.edu. Research partly supported by NSF CAREER Award CNS-0447808.

†RSA Laboratories, Bedford, MA, USA and Computer Science Dept., Columbia University, NY, USA moti@cs.columbia.edu

Contents

1 Introduction

Group signatures is a useful anonymous non-repudiable multi-use credential primitive that was introduced by Chaum and Van Heyst [18]. It involves a group of users, each holding a membership certificate that allows a user to issue a publicly verifiable signature while hiding the identity of the actual signer within the group. The public-verification procedure employs only the public-key of the group. Furthermore, in the case of any dispute or abuse, it is possible for the group manager (GM) to "open" an individual signature and reveal the identity of its originator.

Constructing an efficient group signature has been a research target for many years, see e.g., [19, 17, 14, 15, 10, 29, 4, 2, 12, 26, 9, 3, 11, 13]. A scalable scheme that provides constant signature size and has resistance to attacks by coalitions of users was given in [2]. Earlier constructions were designed without a formal model and definition of security of such schemes, and thus with partial security proofs at the best case (while many were actually broken).

A central issue in group signatures has been the way by which users join the group. Recently, [6] gave the first formal model of a somewhat "relaxed" group signature primitive where a trusted party generates and hands out all users' keys. They also produced a generic solution thus demonstrating the polynomial-time plausibility of their notion of trusted-party aided join group signatures. This is in contrast with users who dynamically join the system and get their individual keys by interacting with the group manager (as in the protocol of [2]). Dynamic joins that allow users to register sequentially were studied formally in [25, 27] where efficient constructions were given and in [6, 7] where a generic plausibility proof was provided.

The most efficient and conceptually simple joining procedure for a group signature scheme (what we will call the "single-message and signature-response paradigm") was illustrated by Camenisch and Stadler [17] who sketched a generic solution (which was followed in careful details in [6, 7]). In this type of joining protocol, the prospective user has an appropriately distributed secret x' and it computes a one way function f on it to obtain $x = f(x')$. The user sends x to the GM who, in turn, signs x and returns the signature σ to the user using an appropriate signing algorithm. This completes the interaction of the join protocol. The possession of the signature σ on $x = f(x')$ enables a user to sign anonymously a message m by simply encrypting x probabilistically into ψ (under the GM's public key or whatever entity is supposed to execute the opening algorithm) and by providing a zero-knowledge proof of (i) the fact that the ψ is an encryption of some x known to the prover, (ii) the fact that the prover knows x' a preimage of that x under f, (iii) the fact that the prover knows a signature issued by the GM on that x.

While the Camenisch-Stadler approach is elegant and advantageous (as we argue below), its instantiation by an efficient scheme turned out to be elusive, since the many schemes that have been suggested in the last eight years approximated it but none really employed it. In fact, all the efficient schemes in the non-trusted-party-aided joining setting that were not broken used additional communications during the join protocol usually to assure that certain constraints and certain knowledge of the joining user is present, i.e., the prospective user had to engage in an interactive zero-knowledge proof with the GM. It was not at all apparent whether the single-message and signature-response join would actually be instantiable in an *efficient* manner in a *provably secure scheme*. Moreover the employment of such proofs of knowledge has the usual shortcomings with respect to adversaries operating in the concurrent setting (namely, rewinding cannot be employed and a "straight-line" approach needs to be followed that makes the joining protocols even more involved).

To conclude the motivation for our result, we summarize the advantages of a group signature employing a single-message and signature-response joining protocol:

1. *Concurrency:* Joining of users can be done concurrently where a batch of users join at the same

time. This enables group managers over the Internet (where servers are multi-thread machines).

2. *Proxy Join:* Users can be joined by a proxy collecting all their requests and then collecting the responses from the group manager; this is a very effective way to enroll companies and organizations by delegating collection and distribution to security officers. It is highly effective in enrolling to an identity escrow scheme without the need for random oracle proofs.

3. Multi-Group Scenario: There may be a number of groups; since single-message and signature-response joins require essentially no interaction between the GM and the prospective user, users may accumulate many GM membership signatures on the same x value non-interactively thus easily becoming members of multiple-groups.

1.1 Our result

In this work we implement the first group signature scheme with a single-message and signature-response join protocol to be exploited for concurrent joins and other advantages as above, thus implementing efficiently the Camenisch-Stadler approach for the first time[1].

We start by presenting the first model of "group signature with concurrent joins" which builds on the recent formal models and consists of a set of attacks. We note that in a privacy primitive interacting users may be conducting simultaneous attacks against each other and these need to be captured formally. We call our attacks: misidentification attack, framing attack and anonymity attack and is an extension of our sequential-join formal model for group signatures in [27]. We then implement a scheme based on specific assumptions and prove its security. The scheme allows adversarial opening of signatures and its signature size is only about twice the size of the scheme of [2] (that did not allow for adversarial opening or concurrent join attacks).

From a technical viewpoint we employ a number of complex primitives including the digital signature scheme of Boneh and Boyen [8] (hence referred to as the BB signature) as well as verifiable encryption for discrete-logarithms that are based on the Paillier encryption function [31, 21, 16, 24].

A novelty of our technical approach (and perhaps a partial explanation why we manage to achieve an efficient single-message and signature-response join) is that we deviate from most of recent group signature literature by instantiating the one-way function employed by the prospective user during the join with multiplication instead of exponentiation. Our general design approach is outlined in figure 1: users sample an RSA modulus and then obtain a short chain of BB certificates on it (numbering from one to five signatures). This modest interaction (which is simply a PKI registration in a domain employing RSA moduli with a BB signature for certification) allows users to sign as group members.

Our security proofs follow a modular approach: in a nutshell, a misidentification adversary is turned into a BB-forger, a framing adversary is turned into a factoring algorithm and an anonymity attacker is turned into a CCA2 adversary against the encryption algorithm we employ. The group signature itself is based on the Fiat-Shamir paradigm, by essentially turning an identity escrow (anonymous identification) system into a signature and employing a random oracle. We note that the interactive version of our group signature yields an identity escrow scheme in a straightforward manner that can also have concurrent group signing by employing general transformation techniques for Σ-protocols, e.g. [23].

[1] In some recent schemes of group signatures and related primitives based on dynamic accumulators [33, 20], a simple two message join was implemented; nevertheless this was to be followed by local modifications of keys of *all existing users*; we do not consider such a protocol efficient. In our solution, keys of other users are unaffected when new members are introduced to the group.

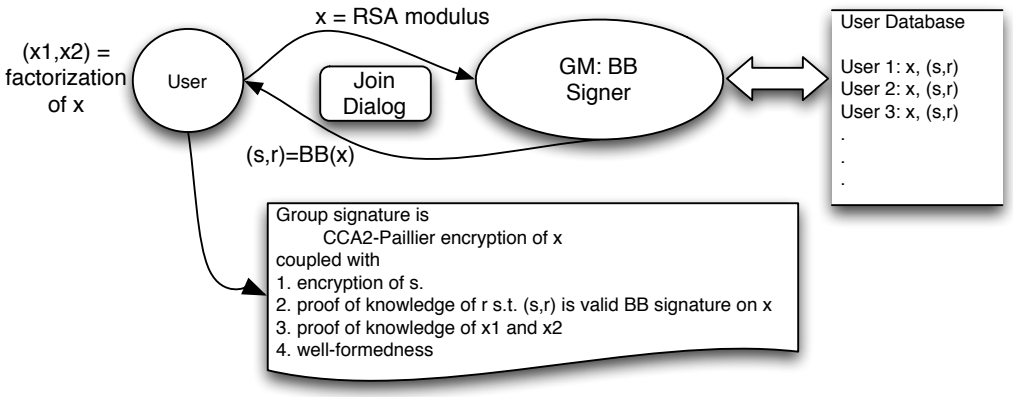

Figure 1: Overview of our general group signature design. The BB signature can be substituted by potentially other signatures that are suitable for algebraic encryption with efficient validity proof.

The present manuscript is a full version containing proofs, minor corrections as well as a more flexible and efficient protocol construction compared to the proceedings version [28].

2 Preliminaries

Interactive Turing Machines and Concurrent Executions. A two-party protocol is a pair of probabilistic polynomial-time bounded Interactive Turing machines $\langle \mathsf{A}, \mathsf{B} \rangle$. Each of A, B has a private input tape, work-tapes, a (joint) communication tape and a private output tape. An execution of a protocol $\langle \mathsf{A}, \mathsf{B} \rangle$ on inputs x, y for the two players will be denoted by $[\mathsf{A}(x), \mathsf{B}(y)]$. For an execution of a protocol we will consider the following random variables: (i) $\mathsf{Trans}[\mathsf{A}(x), \mathsf{B}(y)]$ is the contents of the communication tape after the two parties terminate. (ii) $\mathsf{Out}_{\mathsf{A}}[\mathsf{A}(x), \mathsf{B}(y)]$ is the contents of the private output tape of player A after termination. (iii) $\mathsf{Out}_{\mathsf{B}}[\mathsf{A}(x), \mathsf{B}(y)]$ is the contents of the private output tape of player B after termination.

Now suppose that $\mathcal{P} = \langle \mathsf{A}, \mathsf{B} \rangle$ is a protocol. An "interface oracle" for concurrent simulation of player B, denoted by $\mathcal{I}[\mathcal{P}_{\leftrightarrow \mathsf{B}(y)}]$, is an oracle that accepts the following queries:

Q1. $\texttt{Start} - \texttt{Session}$: The interface oracle $\mathcal{I}[\mathcal{P}_{\leftrightarrow \mathsf{B}(y)}]$ initiates a session for the protocol $\mathcal{P}$: it selects a session identifier s and if B is the player that goes first in the protocol $\mathcal{P}$, the interface simulates the first move of B on input y; the interface returns as answer to the $\texttt{Start} - \texttt{Session}$ query the session identifier s and the output of the simulation of player B's first move (if any). The interface keeps a database with the state of player B for the session identifier s; the state includes all coin tosses of B, and the contents of all tapes including the communication tape.

Q2. $\texttt{Advance} - \texttt{Session}(s, msg)$ The interface oracle looks up the table of sessions and recovers the state of player B for the session with identifier s (if there is no such session the interface returns $\perp$ as answer to the oracle query). If session s exists the interface appends msg to the communication tape of the session and continues the simulation of player B (as if msg is the message that is written to the communication tape of player B by player A).

We will use the notation $M^{\mathcal{I}[\mathcal{P}_{\leftrightarrow \mathsf{B}(\cdot)}]}$ to denote any probabilistic Turing machine M that has access to

an interface oracle as defined above. Note that the interface oracle $\mathcal{I}[\mathcal{P}_{\leftrightarrow \mathsf{A}(x)}]$ (for concurrent executions of player A in the protocol $\mathcal{P}$) can be defined in the same fashion as above. Frequently protocol executions are stateful, e.g. there is a database, or state St in general that an instantiation of the protocol $\mathcal{P}$ may consult. This state St will be maintained by the interface oracle $\mathcal{I}$. In this case we will write $\mathcal{I}_{St}[\mathcal{P}_{\leftrightarrow \mathsf{B}(\cdot)}]$. In the case that a TM M has access to a stateful interface oracle $\mathcal{I}$ we will write $M^{\mathcal{I}_{St}[\mathcal{P}_{\leftrightarrow \mathsf{B}(\cdot)}]}$. Depending on the case, $\mathcal{I}$ may modify the state St or even allow read and write access to St by M. In such cases we will write $M^{\mathcal{I}_{St}[\mathcal{P}_{\leftrightarrow \mathsf{B}(\cdot)}, \mathsf{READ}_{St}, \mathsf{MODIFY}_{St}]}$, where READ_{St} is an oracle interfaced to M through $\mathcal{I}$ that given a fixed input returns the contents of the St and similarly MODIFY_{St} is an oracle that allows M to modify the contents of St using some standard encoding and addressing scheme.

Bilinear Maps. Let $\mathbb{G}_1, \mathbb{G}_2$ two groups of prime order p so that (i) $\mathbb{G}_1 = \langle g_1 \rangle$ and $\mathbb{G}_2 = \langle g_2 \rangle$; (ii) $\tau : \mathbb{G}_2 \rightarrow \mathbb{G}_1$ is an isomorphism with $\tau(g_2) = g_1$ and (iii) $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ is a bilinear map. We remark that in many cases it can be that $\mathbb{G}_1 = \mathbb{G}_2$ (and in this case ψ would be the identity mapping). Let $\mathbb{G}_1 = \langle g_1 \rangle, \mathbb{G}_2 = \langle g_2 \rangle$ groups as above with $|\mathbb{G}_1| = |\mathbb{G}_2| = p$; a bilinear map is a map e s.t. for all $(u, v) \in \mathbb{G}_1 \times \mathbb{G}_2$ it holds that $e(u^x, v^y) = e(u, v)^{xy}$ and $e(g_1, g_2) \neq 1$.

Intractability Assumptions. We will employ the following intractability assumptions:

The *Strong Diffie Hellman Assumption* (SDH) was put forth by Boneh and Boyen [8]. The q-SDH problem over two groups $\mathbb{G}_1, \mathbb{G}_2$ is defined as follows: given a $(q+2)$-tuple $\langle g_1, g_2, g_2^{\gamma}, \ldots, g_2^{(\gamma)^q} \rangle$ as input, output a pair $(g_1^{\frac{1}{\gamma + x}}, x)$ where $x \in \mathbb{Z}_p^*$. The q-SDH assumption suggests that any probabilistic polynomial-time (PPT) algorithm solving the q-SDH problem has negligible success probability. When q is any polynomial-time function on the security parameter we will write simply SDH.

The *Strong-RSA* problem [5] is as follows: given $n, z \in QR(n)$, where $QR(n)$ is the group of quadratic residues of $\mathbb{Z}_n^*$ asks for two integers $u, e > 1$ so that $u^e \equiv_n z$. The Strong-RSA assumption suggests that any PPT algorithm solving the Strong-RSA problem has negligible success probability.

The *Decisional Composite Residuosity* (DCR) assumption [31] is defined as follows: it is computationally hard to distinguish between the distributions of tuples of the form $(N, u^N \bmod N^2)$ where N is an RSA safe composite modulus and $u \leftarrow_R \mathbb{Z}_{N^2}^*$ and the distribution of tuples of the form (N, v) where N is an RSA safe composite modulus and $v \leftarrow_R \mathbb{Z}_N^*$.

3 Group Signatures with Concurrent Join : Modeling

In this section we give the formal definition of group signatures with concurrent join. First we start with the syntax of the signature. The parties that are involved in the scheme include the Group Manager (GM), the Users and the Verifiers.

3.1 Syntax

Definition 3.1 *A group signature scheme with concurrent joins is a digital signature scheme that is comprised of the following five procedures:*

`SETUP`*: it is a probabilistic algorithm that on input a security parameter 1^{ν}, it outputs the group public key $\mathcal{Y}$ (including all system parameters) and the secret key $\mathcal{S}$ for the GM.* `SETUP` *initializes a public state string $St = (St_{users}, St_{join-trans})$ with two components $St_{users} = \epsilon$ and $St_{join-trans} = \epsilon$. The public state string St will hold the user identity database and the database of the Join protocol transcripts. This information will be publicly available and will grow as more users are introduced into the system.*

`JOIN`*: A* protocol *between the GM and a user that results in the user becoming a new group member. The user's output is a* membership certificate *and a* membership secret*. We will denote the i-th user's membership certificate by* cert_i *and the corresponding membership secret by* sec_i*.*

Since `JOIN` *is a two-party protocol, its specification includes the description of two interactive Turing Machines (ITM)* $\mathsf{J}_{\mathsf{user}}, \mathsf{J}_{\mathsf{GM}}$*. Only* $\mathsf{J}_{\mathsf{user}}$ *will have a private output.*

According to the notations of section 2 an execution of the protocol is denoted as $[\mathsf{J}_{\mathsf{user}}(St, \mathcal{Y}) \leftrightarrow \mathsf{J}_{\mathsf{GM}}(St, \mathcal{Y}, \mathcal{S})]$ *and has two "output" components:*

1. *the user private output,* $\langle i, \mathsf{cert}_i, \mathsf{sec}_i \rangle \leftarrow \mathsf{User}[\mathsf{J}_{\mathsf{user}}(St, \mathcal{Y}) \leftrightarrow \mathsf{J}_{\mathsf{GM}}(St, \mathcal{Y}, \mathcal{S})]$*, and*

2. *the public transcript,* $\mathsf{transcript}_i \leftarrow \mathsf{Trans}[\ \mathsf{J}_{\mathsf{user}}(St, \mathcal{Y}) \leftrightarrow \mathsf{J}_{\mathsf{GM}}(St, \mathcal{Y}, \mathcal{S})]$.

After a successful execution of `JOIN` *the following updates are made:* $St_{users} = St_{users} || \langle i \rangle$ *and* $St_{join-trans} = St_{join-trans} || \ \langle i, \mathsf{transcript}_i \rangle$*. The identity-tag i will be selected from a space of possible identity tags denoted by* ID*.*

`SIGN`*: A probabilistic algorithm that given the group's public-key, a membership certificate, a membership secret and a message m, it outputs a group signature for the message m. We write* $\texttt{SIGN}(\mathcal{Y}, \mathsf{cert}_i, \mathsf{sec}_i, m)$ *to denote the application of the signing algorithm on the message m.*

`VERIFY`*: An algorithm for establishing the validity of an alleged group signature on a message with respect to a group public-key. If σ is a signature on a message m, then we have* $\texttt{VERIFY}(\mathcal{Y}, m, \sigma) \in \{\top, \bot\}$.

`OPEN`*: An algorithm that, given a message, a valid group signature on it, a group public-key, the GM's secret-key and the public-state it determines the identity of the signer. In particular* $\texttt{OPEN}(m, \sigma, St, \mathcal{Y}, \mathcal{S}) \in St_{users} \cup \{\bot\}$.

Notation. Below we will introduce a helpful notation for describing the relationship between transcripts and membership certificates and secrets. Given $\langle \mathcal{Y}, \mathcal{S} \rangle \leftarrow \texttt{SETUP}(1^\nu)$ we define the following relations over strings based on $\mathcal{Y}$ and some public state St,

$\langle i, \mathsf{cert}, \mathsf{sec} \rangle \leftrightharpoons_{(\mathcal{Y}, St)} \mathsf{transcript}$ if there exist coin tosses ρ for J_{GM} and $\mathsf{J}_{\mathsf{User}}$ so that

$$\langle i, \mathsf{cert}, \mathsf{sec} \rangle = \mathsf{User}[\mathsf{J}_{\mathsf{User}}(St, \mathcal{Y}) \leftrightarrow \mathsf{J}_{\mathsf{GM}}(St, \mathcal{Y}, \mathcal{S})](\rho)$$

and

$$\mathsf{transcript} = \mathsf{Trans}[\mathsf{J}_{\mathsf{User}}(St, \mathcal{Y}) \leftrightarrow \mathsf{J}_{\mathsf{GM}}(St, \mathcal{Y}, \mathcal{S})](\rho)$$

Similarly we will define $\mathsf{cert} \leftrightharpoons_{\mathcal{Y}} \mathsf{sec}$, if there exist coin tosses ρ for J_{GM} and $\mathsf{J}_{\mathsf{User}}$ and a state St so that

$$\langle i, \mathsf{cert}, \mathsf{sec} \rangle = \mathsf{User}[\mathsf{J}_{\mathsf{User}}(St, \mathcal{Y}) \leftrightarrow \mathsf{J}_{\mathsf{GM}}(St, \mathcal{Y}, \mathcal{S})](\rho)$$

Finally we define the set of all valid public states Valid as follows: $St_0 \in \mathsf{Valid}$ if there exists a PPT Turing machine M and $\langle \mathcal{Y}, \mathcal{S} \rangle \leftarrow \texttt{SETUP}(1^\nu)$ so that when $M^{\mathcal{I}_{St}[\texttt{JOIN} \leftrightarrow GM(St, \mathcal{Y}, \mathcal{S}), \mathsf{READ}_{St}]}$ terminates it holds that $St = St_0$ and the interface oracle $\mathcal{I}$ given to M initializes $St = (\epsilon, \epsilon)$ and allows M to have read access to St through `READ` queries. If $\mathcal{I}_{St}$ initializes St to some $St_0 \in \mathsf{Valid}$ that is not (ϵ, ϵ) then this defines the set of all *valid extensions* of the public-state St_0 that will be denoted by Valid_{St_0}. Obviously $\mathsf{Valid} = \mathsf{Valid}_{(\epsilon, \epsilon)}$.

3.2 Correctness

Below we define the correctness of a group signature scheme that satisfies the above syntax. Note that a group signature is a tuple $\langle \mathtt{SETUP}, \mathtt{JOIN}, \mathtt{SIGN}, \mathtt{VERIFY}, \mathtt{OPEN}\rangle$ with $\mathtt{JOIN} = \langle \mathsf{J}_{\mathsf{User}}, \mathsf{J}_{\mathsf{GM}}\rangle$.

Definition 3.2 *A group signature with concurrent join is correct if the following are true:*

C1. *(users are assigned unique names) For any $St \in \mathsf{Valid}$ it holds that St_{users} contains no multiply defined identity-tags, i.e., if $St_{users} = \langle i_1\rangle||\ldots||\langle i_K\rangle$ it holds that $j \neq j' \Rightarrow i_j \neq i_{j'}$.*

C2. *(signing is correct) For any $\langle \mathcal{Y}, \mathcal{S}\rangle \leftarrow \mathtt{SETUP}(1^\nu)$, any strings $\mathsf{cert} \leftrightharpoons_{\mathcal{Y}} \mathsf{sec}$ and any $m \in \{0,1\}^*$, it holds that $\mathtt{VERIFY}(\mathcal{Y}, m, \mathtt{SIGN}(\mathcal{Y}, \mathsf{cert}, \mathsf{sec}, m)) = \top$.*

C3. *(open is correct) For any $\langle \mathcal{Y}, \mathcal{S}\rangle \leftarrow \mathtt{SETUP}(1^\nu)$, any $St \in \mathsf{Valid}$, any $m \in \{0,1\}^*$, and any $\langle i, \mathsf{cert}, \mathsf{sec}\rangle \leftrightharpoons_{(\mathcal{Y},St)} \mathsf{transcript}$ it holds that $\mathtt{OPEN}(m, \mathtt{SIGN}(\mathcal{Y}, \mathsf{cert}, \mathsf{sec}, m), St'', \mathcal{Y}, \mathcal{S}) = i$, where $St'' \in \mathsf{Valid}_{St'}$ and St' is defined as follows: $St'_{users} = St_{users}||\langle i\rangle$ and $St'_{join-trans} = St_{join-trans}||\langle i, \mathsf{transcript}\rangle$.*

Property *C1* requires that the JOIN protocol assigns a different identity tag to all users. Property *C2* ensures the correctness of the underlying signing and verification for any valid signing key (that includes a membership secret and a membership certificate). Finally, property *C3* ensures that the OPEN algorithm correctly identifies all signers: in particular it says that if a user is introduced at some moment in the system's operation and the public-state St is updated with the user's identity tag resulting to state St' then it holds that whenever this user issues a group signature the user will be correctly identified for every public state St'' that succeeds the public-state St' of the system. We note that it may be viable to collapse *C1* and *C3* but, given the intuitiveness of the formulation, we keep them as separate properties.

3.3 Security

Security against group signatures with concurrent join, will be broken into three basic properties following the model designs of [25, 27]. The properties are formalized as games between the adversary and an entity called the interface, denoted by $\mathcal{I}$ that represents the *uncorrupted aspect of the system* in each attack.

Misidentification. In a misidentification attack, the adversary joins the system through possibly many concurrent sessions of the JOIN protocol and it attempts to produce a signature that cannot be opened to any of the users that are adversarially controlled. We note that without loss of generality we will assume that *all* users introduced in the system are adversarially controlled; this means that the goal of the adversary is to simply make the OPEN algorithm to fail. We remark that adversaries that make the OPEN algorithm to point to an innocent user will be handled in the framing attack (next paragraph).

Below, $\mathcal{I}_{St}[\mathtt{JOIN}_{\leftrightarrow GM}]$ will denote the interface oracle for concurrent simulation of the GM party in the protocol JOIN (refer to section 2 for the definition). Note that the interface $\mathcal{I}$ has access to the public state string St and it updates it accordingly whenever a new user (the adversary that is) successfully completes the JOIN dialog. Also, an oracle $\mathtt{READ}_{St}$ is provided to the adversary that allows him to read the contents of the public state database that contains the identification transcripts and user identity tags. Finally, an oracle OPEN is provided to the adversary that allows him to submit signatures and obtain the output of the opening algorithm.

The Misidentification-Attack Game $G^{\mathcal{A}}_{\mathsf{mis}}$ (denoted by $G^{\mathcal{A}}_{\mathsf{mis}}(1^\nu)$):

1. $\langle \mathcal{Y}, \mathcal{S}\rangle \leftarrow \mathtt{SETUP}(1^\nu)$; $St = (St_{users}, St_{join-trans}) = (\epsilon, \epsilon)$;
2. $\langle m, \sigma\rangle \leftarrow \mathcal{A}^{\mathcal{I}_{St}[\mathtt{JOIN}_{\leftrightarrow GM(St,\mathcal{Y},\mathcal{S})}, \mathtt{READ}_{St}, \mathtt{OPEN}]}(\mathcal{Y})$;
3. If $(\mathtt{VERIFY}(\mathcal{Y}, m, \sigma) = \top) \wedge (\mathtt{OPEN}(m, \sigma, \mathcal{Y}, \mathcal{S}, St) = \bot)$ then return $\top$ else return $\bot$;

We will say that a group signature is secure against misidentification attacks with concurrent join provided that for all $\mathcal{A}$ it holds that $\mathbf{Prob}[G^{\mathcal{A}}_{\mathsf{mis}}(1^\nu) = \top] = 1 - \mathsf{negl}(\nu)$.

Framing. In a framing attack the adversary plays the role of the system where the interface represents a handful of innocent users. A framing attack is meant to capture any adversarial behavior that allows the adversary to make the open algorithm point to an innocent user. We remark that this captures the notion of exculpability as well as any other adversarial behavior that frames an innocent user. In the concurrent setting, we allow the adversary to initiate many concurrent executions of the JOIN dialog playing the role of a malicious GM. The goal of the adversary now is to produce a signature that opens to one of the innocent users.

Naturally in modeling such an attack we cannot allow to the adversary to do all the bookkeeping for the user database himself (otherwise an OPEN operation would be without meaning). Every time the adversary successfully terminates a JOIN dialog with an innocent user that is controlled by the interface $\mathcal{I}$, the interface will add the user identity into the St_{users} and will append the whole communication transcript to $St_{join-trans}$. Moreover it will keep a private database containing the secrets of the innocent users that will have the format $\langle i, \mathsf{sec}_i \rangle$ (these will not be accessible to the adversary). In addition to the above, we will allow the adversary to submit queries to a SIGN oracle that will be handled by the interface $\mathcal{I}$ and will accept the identity of one of the innocent users and a message and will return a signature of this message with the signing mechanism of the named user.

We allow the adversary to have appropriately restricted modify access to the public-state St; this access will be handled by $\mathcal{I}$ in the form of the $\texttt{MODIFY}_{St}$ oracle query. As mentioned already we will not give to the adversary full write capability to the public state St since if he is allowed to this, opening any signature correctly would be meaningless (e.g., if the adversary erases the database of JOIN transcripts it is straightforward that the opening capability is cancelled). The restrictions are as follows: $\texttt{MODIFY}_{St}$ will not permit the adversary to insert a join transcript that reuses an identity tag (this restriction is essential to maintain the semantics of the OPEN unambiguous) and will not allow the adversary to modify any of the identity tags or join transcripts of the innocent users (to these the adversary will have read-only access). Any other modification of the public-state will be allowed by $\mathcal{I}$ (in particular the adversary is allowed to introduce users to the public-state as well as erase them — for this reason there is no need for a "corrupt" oracle).

We will use the notation $St^{\mathcal{I}}_{users}$ to denote all innocent users in the system that are introduced by the execution of the concurrent JOIN oracle and are managed by the interface oracle $\mathcal{I}$.

The Framing-Attack Game $G^{\mathcal{A}}_{\mathsf{fra}}$ (denoted by $G^{\mathcal{A}}_{\mathsf{fra}}(1^\nu)$):

1. $\langle \mathcal{Y}, \mathcal{S} \rangle \leftarrow \texttt{SETUP}(1^\nu)$; $St = (St_{users}, St_{join-trans}) = (\epsilon, \epsilon)$;
2. $\langle m, \sigma \rangle \leftarrow \mathcal{A}^{\mathcal{I}[\texttt{JOIN}_{\leftrightarrow User(\mathcal{Y})}, \texttt{SIGN}, \texttt{READ}_{St}, \texttt{MODIFY}_{St}]}(\mathcal{Y}, \mathcal{S})$
3. $i = \texttt{OPEN}(m, \sigma, St, \mathcal{Y}, \mathcal{S})$;
4. If $(\texttt{VERIFY}(\mathcal{Y}, m, \sigma) = \top) \wedge (i \in St^{\mathcal{I}}_{users})$ then return $\top$ else return $\bot$;

We say that a group signature satisfies security against framing attacks with concurrent join provided that for all $\mathcal{A}$ it holds that $\mathbf{Prob}[G^{\mathcal{A}}_{\mathsf{fra}}(1^\nu) = \top] = 1 - \mathsf{negl}(\nu)$.

Anonymity. Finally, anonymity is modeled as a sort of CCA2 attack against the identity encryption embedding mechanism of the group signature.

The Anonymity-attack Game $G^{\mathcal{A}}_{\mathsf{anon}}$ (denoted by $G^{\mathcal{A}}_{\mathsf{anon}}(1^\nu)$):

1. $\langle \mathcal{Y}, \mathcal{S} \rangle \leftarrow \mathtt{SETUP}(1^\nu)$; $St = (St_{users}, St_{join-trans}) = (\epsilon, \epsilon)$;
2. $\langle aux, m, \mathsf{cert}_1, \mathsf{sec}_1, \mathsf{cert}_2, \mathsf{sec}_2, \rangle \leftarrow \mathcal{A}^{\mathcal{I}[\mathtt{JOIN}_{\leftrightarrow GM(St,\mathcal{Y},\mathcal{S})}, \mathtt{READ}_{St}, \mathtt{OPEN}]}(\mathsf{play}, \mathcal{Y})$
3. if $\neg((\mathsf{cert}_1 \leftrightharpoons_{\mathcal{Y}} \mathsf{sec}_1) \wedge (\mathsf{cert}_2 \leftrightharpoons_{\mathcal{Y}} \mathsf{sec}_2))$ or $\mathsf{cert}_1 = \mathsf{cert}_2$ then terminate; return $\perp$;
4. Choose $b \leftarrow_R \{1, 2\}$;
5. $\psi \leftarrow \mathtt{SIGN}(\mathcal{Y}, \mathsf{cert}_b, \mathsf{sec}_b, m)$;
6. $b^* \leftarrow \mathcal{A}^{\mathcal{I}[\mathtt{JOIN}_{\leftrightarrow GM(St,\mathcal{Y},\mathcal{S})}, \mathtt{READ}_{St}, \mathtt{OPEN}^{\neg\psi}]}(\mathsf{guess}, aux)$;
7. if $b = b^*$ return $\top$ else return $\perp$;

We note that the $\mathtt{OPEN}^{\neg\psi}$ oracle operates as the $\mathtt{OPEN}$ oracle with the usual restriction that it should return $\perp$ if the adversary submits ψ as the signature to be opened.

A group signature is said to be secure against anonymity attacks with concurrent join provided that for all $\mathcal{A}$ it holds that $2\mathbf{Prob}[G^{\mathcal{A}}_{\mathsf{anon}}(1^\nu) = \top] - 1 = \mathsf{negl}(\nu)$.

Based on all the above we will say that a group signature with concurrent join is *secure* provided that it is secure against misidentification, framing and anonymity attacks.

4 Group Signatures with Efficient Concurrent Join : Construction

In this section we describe our efficient group signature construction. Our construction below is an optimized and more flexible variant of the scheme presented in the proceedings version of this work, [28]. A number of primitives proved to be instrumental in our construction including: Integer commitments [22], BB signatures [8] and a CCA2 variant of Pailier encryption [31, 24, 16]. We first begin by describing the public-parameters our system will employ.

Public-parameters. The public parameters of the scheme are as follows:

- p1 A (small) integer parameter v ($1 \leq \mathrm{v} \leq 5$) and v pairs of groups of order p_t where p_t is a ℓ_p-bit prime, $p_t > 2^{\ell_p - 1}$, denoted by $\mathbb{G}_{1,t} = \langle g_{1,t} \rangle$ and $\mathbb{G}_{2,t} = \langle g_{2,t} \rangle$, so that there is a mapping e_t and a group $\mathbb{G}_t$ and $e_t : \mathbb{G}_{1,t} \times \mathbb{G}_{2,t} \rightarrow \mathbb{G}_t$ is a bilinear map. Note that t ranges in $\{1, \ldots, \mathrm{v}\}$. It is assumed that all $p_1, \ldots, p_{\mathrm{v}}$ are distinct. These groups will be assumed to satisfy the Strong Diffie Hellman assumption.

- p2 an RSA-modulus n, of ℓ_n bits; n is selected so that Strong-RSA will be infeasible over $QR(n)$.

- p3 Four integer ranges S, S', S_f, S'_f over the parameters $\ell, \ell_p, \mathrm{v}, k, k'$; note that ℓ is an integer parameter suitable for the selection of RSA moduli. Intuitively S' will be a range of integers from which RSA moduli will be chosen and S'_f will be a range of integers from which the factors of such moduli will be chosen; the ranges S, S_f will be extensions of these two ranges that will be used in the construction.

  The integer ranger S is defined as $S =_{\mathsf{df}} S(2^{\ell-1}, 2^{\mathrm{v}(\ell_p-1)-1})$; we recall that the notation $S(a, b) =_{\mathsf{df}} \{a - b, \ldots, a + b\}$ (we call this an integer sphere centered at a). Observe that if $x, y \in S$ and $x \equiv_{p_{\mathrm{t}}} y$ for all $t = 1, \ldots, \mathrm{v}$ then it holds that $x = y$; indeed, the given condition implies that $x \equiv_{p_1 \ldots p_{\mathrm{v}}} y$ (cf. Chinese remaindering). But since $p_1 \ldots p_{\mathrm{v}} > 2^{\mathrm{v}(\ell_p-1)} = 2 \cdot 2^{\mathrm{v}(\ell_p-1)-1}$, which is the length of the sphere S, it follows that $x = y$.

  Now for two parameters $k, k' \in \mathbb{Z}$ we select the range S' as follows: $S' =_{\mathsf{df}} S(2^{\ell'}, 2^{\mu'}) = S(2^{\ell-1}, 2^{\mathrm{v}(\ell_p-1)-1-k-k'})$. Moreover we select the range $S_f =_{\mathsf{df}} S(2^{\ell_f}, 2^{\mu_f}) = S(2^{\ell/2-1}, 2^{\ell/2-1+k+k'})$ and $S'_f = S(2^{\ell'_f}, 2^{\mu'_f}) = S(2^{\ell/2-1}, 2^{\ell/2-1})$.

Note that all ranges S, S', S_f, S'_f are assumed subsets of $\{1, \ldots, \phi(n)\}$. Moreover note that $S'_f = S(2^{\ell/2-1}, 2^{\ell/2-1}) = \{0, \ldots, 2^{\ell/2}\}$ contains all $\ell/2$-bit numbers; as a result, any ℓ-bit RSA modulus that belongs to the range S and splits to two equal size factors of $\ell/2$ bits will have its factors inside S'_f.

p4 a safe RSA-modulus N of ℓ_N bits with $N = PQ$ and $P = 2P' + 1, Q = 2Q' + 1$, so that in the group $\mathbb{Z}^*_{N^2}$ it holds that the DCR assumption is hard, and the value $G = (G_0)^{2N} (\mathrm{mod} N^2)$ is selected with $G_0 \leftarrow_R \mathbb{Z}^*_{N^2}$. Note that with overwhelming probability $\langle G \rangle$ is the subgroup of quadratic residues modulo N^2 that are simultaneously N-th residues; note that $\#\langle G \rangle = P'Q'$. It will be assumed that $\mathbb{Z}_N$ contains the range S.

`SETUP`. The procedure first generates the public-parameters p1 and p2 and p3 as described above. Then, it executes the following steps:

It selects v pairs of values $\gamma_t, \delta_t \leftarrow_R \mathbb{Z}_p$ and sets $w_t = g_{2,t}^{\gamma_t}$ and $v_t = g_{2,t}^{\delta_t}$ for $t = 1, \ldots, \mathrm{v}$; this is the setup for BB signatures, cf. [8].

It selects $g, f_1, f_2, f_3, f_4 \leftarrow_R QR(n)$. These values will be used for integer commitments.

(Opening functionality) the public parameters N, G according to p4 are selected as well as $H_1, H_2, H_3 \in \langle G \rangle$ with $H_i = G^{a_i}$, $a_i \leftarrow_R \mathbb{Z}_{\lfloor N/4 \rfloor}$ for $i = 1, 2, 3$ and a hash-key hk for a universal one-way hash function family UOHF. We remark that this step can be entirely separated from the GM's setup phase and executed by an opening authority. Nevertheless for convenience and simplification of the presentation we do not make further distinction in the present version of the paper.

The public-key $\mathcal{Y}$ is set to

$$\langle g_{1,t}, g_{2,t}, w_t, v_t, \mathrm{desc}(\mathbb{G}_{1,t} || \mathbb{G}_{2,t} || \mathbb{G}_t) || e_t \rangle_{t=1}^{\mathrm{v}} || \langle \mathsf{UOHF}, g, f_1, f_2, f_3, f_4, n, N, G, H_1, H_2, H_3, \mathsf{hk} \rangle$$

and the secret key $\mathcal{S}$ is set to to $\langle \gamma_t, \delta_t \rangle_{t=1}^{\mathrm{v}} || \langle a_1, a_2, a_3 \rangle$. Note that the factorization of n is not needed and thus it can be discarded.

`JOIN`. In the join protocol execution, the user will obtain a v-long chain of BB signatures on an RSA modulus that he selects. A user's membership certificate is the signature-chain together with the RSA modulus; a user's membership secret on the other hand is the factorization of the modulus. The join procedure between a prospective user and the GM is described in detail below:

- (User→GM) The user initiates the procedure and selects $x \in S'$ to be an ℓ-bit RSA modulus with x_1, x_2 its two $\ell/2$-bit long prime divisors, so that $x \notin S_f$ and $x_1, x_2 \in S'_f$. The User transmits x to the GM.

- (GM→User) The GM checks whether $x \in S' - \{0, 1\}$ and whether x is unique in the system (i.e., it was not submitted by another user in a previous or concurrent `JOIN` instantiation); if either check fails the GM terminates the `JOIN` protocol; otherwise (i) it reads the public-state St, selects $i \in \mathsf{ID}$ so that $i \notin St_{users}$ and in such a manner that i is distinct from any other concurrent executions that he may be involved simultaneously and writes to its communication tape the values $\langle i, \sigma_1, \ldots \sigma_{\mathrm{v}}, r \rangle$ where $r \leftarrow_R \mathbb{Z}_{p_1 \ldots p_{\mathrm{v}}}$ and $\sigma_t = g_{1,t}^{1/(\gamma_t + x + \delta_t r)}$ for $t = 1, \ldots, \mathrm{v}$; finally it updates $St_{join-trans}$ by appending to it the tuple $\langle i, \sigma_1, \ldots, \sigma_{\mathrm{v}}, r \rangle$ and sets $St_{users} = St_{users} || \langle i \rangle$.

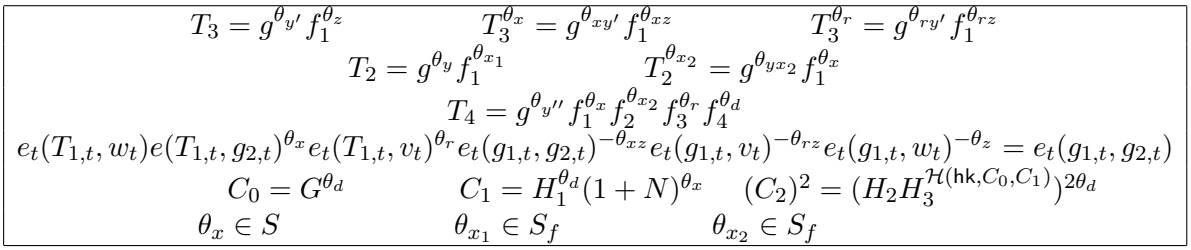

Figure 2: The relations defining the signature of knowledge. Note that the parameter t is running from $1, \ldots \mathrm{v}$.

- The user verifies that $e_t(\sigma_t, w_t g_{2,t}^x v_t^r) = e_t(g_{1,t}, g_{2,t})$ for $t = 1, \ldots, \mathrm{v}$ and that $i \notin St_{users}$; if a test fails the user fails the `JOIN` dialog. Otherwise, it terminates successfully by setting his membership certificate to $\mathsf{cert} = \langle x, \sigma_1, \ldots, \sigma_{\mathrm{v}}, r \rangle_{t=1}^{\mathrm{v}}$ and his membership secret to $\mathsf{sec} = \langle x_1, x_2 \rangle$.

Observe that the user *does not prove* that x was selected appropriately; Perhaps surprisingly, we show that this is still sufficient for security in the concurrent setting. Naturally if the user chooses x inappropriately two things may happen: (i) the user may not be able to issue group signatures, e.g., this may happen when x is a prime; this naturally is of no concern to the GM, (ii) the user selects x as an integer that is easy to factor; while this is of concern there is nothing that can be done about it: this case is conceptually the same as the case that the user just leaks its secret-key; while this possibility is annoying there is little that can be done to prevent this in any group signature scheme (and it is beyond the scope of the present paper at any rate).

`SIGN`. We present the signing algorithm. The user possesses the following: a membership certificate $\langle x, \sigma_1, \ldots, \sigma_{\mathrm{v}}, r \rangle$ and the corresponding membership secret x_1, x_2. The signing algorithm will be obtained by applying the Fiat-Shamir Heuristic on an appropriately designed proof of knowledge. First, the signer computes the following values:

$T_{1,t} = g_{1,t}^z \sigma_t, \ \ t = 1, \ldots, \mathrm{v}$	$z \leftarrow_R \mathbb{Z}_{p_1 \ldots p_{\mathrm{v}}}$	in $\mathbb{G}_{1,1}, \ldots, \mathbb{G}_{1,\mathrm{v}}$
$T_2 = g^y f_1^{x_1}$	$y \leftarrow_R S(1, 2^{\ell_n - 2})$	in $QR(n)$
$T_3 = g^{y'} f_1^z$	$y' \leftarrow_R S(1, 2^{\ell_n - 2})$	in $QR(n)$
$T_4 = g^{y''} f_1^x f_2^{x_2} f_3^r f_4^d$	$y'' \leftarrow_R S(1, 2^{\ell_n - 2})$	in $QR(n)$
$C_0 = G^d$	$d \leftarrow_R S(1, 2^{\ell_N - 2})$	in $\mathbb{Z}_{N^2}^*$
$C_1 = H_1^d (1+N)^x$		in $\mathbb{Z}_{N^2}^*$
$C_2 = \|\|(H_2 H_3^{\mathcal{H}(\mathsf{hk}, C_0, C_1)})^d\|\|$		in $\mathbb{Z}_{N^2}^*$

Note that $||x|| = x$ if $x \leq N^2/2$ and $||x|| = N^2 - x$ otherwise. Also recall that $S(a, b) =_{\mathsf{df}} \{a - b, \ldots, a + b\}$. Subsequently the signer will construct the signature "of knowledge" on the given message m by providing a proof of knowledge for the relations given in figure 2 that involve the fourteen witnesses $\theta_z, \theta_x, \theta_{xz}, \theta_r, \theta_{rz}, \theta_{x_1}, \theta_{x_2}, \theta_y, \theta_{y'}, \theta_{y''}, \theta_{xy''}, \theta_{ry'}, \theta_{yx_2}, \theta_d$.

Given the coin tosses of the signer for the selection of $T_{1,1}, \ldots, T_{1,\mathrm{v}}, T_2, T_3, T_4, C_0, C_1$, the witnesses needed in figure 2 are selected as follows: $\theta_z = z, \theta_x = x, \theta_{xz} = x \cdot z, \theta_r = r, \theta_{rz} = r \cdot z, \theta_{x_1} = x_1, \theta_{x_2} = x_2, \theta_y = y, \theta_{y'} = y', \theta_{y''} = y'', \theta_{xy'} = x \cdot y', \theta_{ry'} = r \cdot y', \theta_{yx_2} = y \cdot x_2, \theta_d = d$. Now, given a message m, the signature will be constructed as follows:

1. (choose blindings) the values $\rho_z \leftarrow_R \pm\{0,1\}^{\nu\ell_p+k+k'}$, $\rho_x \leftarrow_R \pm\{0,1\}^{\mu'+k+k'}$, $\rho_{xz} \leftarrow_R \pm\{0,1\}^{\nu\ell_p+\mu'+k+k'}$, $\rho_r \leftarrow_R \pm\{0,1\}^{\nu\ell_p+k+k'}$, $\rho_{rz} \leftarrow_R \pm\{0,1\}^{2\nu\ell_p+k+k'}$ $\rho_{x_1} \leftarrow_R \pm\{0,1\}^{\mu'_f+k+k'}$, $\rho_{x_2} \leftarrow_R \pm\{0,1\}^{\mu'_f+k+k'}$, $\rho_y, \rho_{y'}, \rho_{y''} \leftarrow_R \pm\{0,1\}^{\ell_n-2+k+k'}$, $\rho_{xy'} \leftarrow_R \pm\{0,1\}^{\ell_n-2+\mu'+k+k'}$ $\rho_{ry'} \leftarrow_R \pm\{0,1\}^{\ell_n-2+\nu\ell_p+k+k'}$ $\rho_{yx_2} \leftarrow_R \pm\{0,1\}^{\ell_n-2+\mu'_f+k+k'}$ $\rho_t \leftarrow_R \pm\{0,1\}^{\ell_N-2+k+k'}$ are selected. Using these values the following are computed (where t runs $1,\ldots,\nu$):

$$R_1 = g^{\rho_{y'}} f_1^{\rho_z} \qquad R_2 = T_3^{-\rho_x} g^{\rho_{xy'}} f_1^{\rho_{xz}} \qquad R_3 = T_3^{-\rho_r} g^{\rho_{ry'}} f_1^{\rho_{rz}}$$

$$R_4 = g^{\rho_y} f_1^{\rho_{x_1}} \qquad R_5 = T_2^{-\rho_{x_2}} g^{\rho_{yx_2}} f_1^{\rho_x}$$

$$R_6 = g^{\rho_{y''}} f_1^{\rho_x} f_2^{\rho_{x_2}} f_3^{\rho_r} f_4^{\rho_d}$$

$$R_{7,t} = e_t(T_{1,t}, g_{2,t})^{\rho_x} e_t(T_{1,t}, v_t)^{\rho_r} e_t(g_{1,t}, g_{2,t})^{-\rho_{xz}} e_t(g_{1,t}, v_t)^{-\rho_{rz}} e_t(g_{1,t}, w_t)^{-\rho_z}$$

$$R_8 = G^{\rho_d} \qquad R_9 = H_1^{\rho_d}(1+N)^{\rho_x} \qquad R_{10} = (H_2 H_3^{\mathcal{H}(\mathsf{hk},C_0,C_1)})^{2\rho_d}$$

2. (calculate challenge) using a hash function denoted by HASH the value

$$c \leftarrow \mathtt{HASH}(m||T_1||\ldots||T_4||R_1||\ldots||R_9,R_{10})$$

is computed. The range of HASH is considered to be $\{0,1\}^k$.

3. (calculate response) Subsequently the following values are computed:

$s_z = \rho_z - cz$	in $\mathbb{Z}$	$s_x = \rho_x - c(x - 2^{\ell'})$	in $\mathbb{Z}$
$s_{xz} = \rho_{xz} - cxz$	in $\mathbb{Z}$	$s_r = \rho_r - cr$	in $\mathbb{Z}$
$s_{rz} = \rho_{rz} - crz$	in $\mathbb{Z}$	$s_{x_1} = \rho_{x_1} - c(x_1 - 2^{\ell''})$	in $\mathbb{Z}$
$s_{x_2} = \rho_{x_2} - c(x_2 - 2^{\ell/2-1})$	in $\mathbb{Z}$	$s_y = \rho_y - cy$	in $\mathbb{Z}$
$s_{y'} = \rho_{y'} - cy'$	in $\mathbb{Z}$	$s_{y''} = \rho_{y''} - cy''$	in $\mathbb{Z}$
$s_{xy'} = \rho xy' - cx \cdot y'$	in $\mathbb{Z}$	$s_{ry'} = \rho ry' - cr \cdot y'$	in $\mathbb{Z}$
$s_{yx_2} = \rho_{yx_2} - cy \cdot x_2$	in $\mathbb{Z}$	$s_d = \rho_d - cd$	in $\mathbb{Z}$

The output of the signing algorithm is the tuple: $\langle T_{1,1},\ldots,T_{1,\nu},T_2,T_3,T_4,C_0,C_1,C_2,c,s_z,s_x,s_{xz},s_r,s_{rz},s_{x_1},s_{x_2},s_y,s_{y'},s_{y''},s_{xy'},s_{ry'},s_{yx_2},s_d\rangle$.

VERIFY. Signature verification is achieved by the following tests:

$$s_x \stackrel{?}{\in} \pm\{0,1\}^{\mu'+k+k'+1} \ \wedge\ s_{x_1}, s_{x_1} \stackrel{?}{\in} \pm\{0,1\}^{\mu'_f+k+k'+1} \ \wedge\ C_0, C_1, C_2 \stackrel{?}{\in} \mathbb{Z}^*_{N^2} \ \wedge\ C_2 \stackrel{?}{\leq} N^2/2$$

$$\begin{aligned} c \stackrel{?}{=} \mathsf{HASH}\ \Big(& m\,||\,T_{1,1}\,||\,\ldots||\,T_{1,\nu}\,||\,T_2\,||\,T_3\,||\,T_4\,|| \\ & ||\,g^{s_{y'}} f_1^{s_z} T_3^c || T_3^{-s_x+c2^{\ell'}} g^{s_{xy'}} f_1^{s_{xz}} || T_3^{-s_r} g^{s_{ry'}} f_1^{s_{rz}} \\ & ||g^{s_y} f_1^{s_{x_1}-c2^{\ell''}} T_2^c || T_2^{-s_{x_2}+c2^{\ell/2-1}} g^{s_{yx_2}} f_1^{s_x-c2^{\ell'}} \\ & ||g^{s_{y''}} f_1^{s_x-c2^{\ell'}} f_2^{s_{x_2}-c2^{\ell/2-1}} f_3^{s_r} f_4^{s_d} T_4^c \\ & ||_{t=1,\ldots,\nu}\ e_t(T_{1,t},g_{2,t})^{s_x-c2^{\ell'}} e_t(T_{1,t},v_t)^{s_r} e_t(g_{1,t},g_{2,t})^{-s_{xz}} \\ & \quad e_t(g_{1,t},v_t)^{-s_{rz}} e_t(g_{1,t},w_t)^{-s_z} e_t(g_{1,t},g_{2,t})^c e_t(T_{1,t},w_t)^{-c} \\ & ||C_0^c G^{s_d} || C_1^c H_1^{s_d}(1+N)^{s_x-c2^{\ell'}} || C_2^c (H_2^2 H_3^{2\mathcal{H}(\mathsf{hk},C_0,C_1)})^{s_d}\Big) \end{aligned}$$

OPEN. Given a signature as described above: first the signature is verified as well as the relation $(C_2)^2 = C_0^{2(a_2+\theta\mathcal{H}(\mathsf{hk},C_0,C_1)}$ is checked. If any check fails OPEN returns $\perp$. Otherwise, OPEN computes $\tilde{m} = C_1^2 C_0^{-2a_1}$; due to the properties enforced by the proof of knowledge (cf. figure 2) it holds

that $x = (\tilde{m}2^{-1} \bmod N - 1)/N$. Then, the OPEN algorithm searches $St_{join-trans}$ for transcripts of the form $\langle j, x_j, \sigma_{j,1}, \ldots, \sigma_{j,\mathrm{v}}, r_j \rangle$ with $x_j = x$; the identity j of the signer is thus recovered. If no such x_j is found, OPEN returns $\perp$.

Length of the signature. It is easy to verify based on the above description that the length of the signature (in bits) is approximately $11 \cdot \mathrm{v} \cdot \ell_p + 9 \cdot \ell_n + 6 \cdot \ell_N + 12k + 11k' + \ell$. We note that several optimizations can be applied potentially to decrease the above size Still at this present form, the protocol, is clearly within practical limits (cf. next paragraph as well).

Parameter Selection. The parameters employed in the construction above are as follows: (i) v, the length of the BB signature chain, (ii) ℓ_p : the size of the elliptic curve groups $\mathbb{G}_{1,t}$ that are employed in the BB signatures, (iii) ℓ_n : an RSA modulus size that is used for commitments. (iv) ℓ : the RSA modulus size that is selected by users. (v) k' and k': parameters affecting the soundness and zero-knowlege properties of the non-interactive proof of knowledge that is employed in the scheme. (vi) ℓ_N : an RSA modulus suitable for the Paillier-like encryption employed in the construction.

A possible choice of the parameters will then be as follows: $\ell_p = 236$, $\ell_n = 1024$, $\ell = 1000$, $k = k' = 80$, $\ell_N = 1024$ and $\mathrm{v} = 5$. With this selection, a chain of 5 BB signatures will be employed over elliptic curve groups of 201-bit prime order and each user will be selecting RSA moduli from $S' = S(2^{999}, 2^{999})$ (i.e., arbitrary 1000-bit RSA moduli) so that the two factors should belong to $S'_f = S(2^{499}, 2^{499})$. Based on these parameters the total signature length is about 32 Kb. In fact one can choose much more efficient parameters if RSA moduli of some specific form are selected; for example, a much more tight parameter choice would be as follows: $\ell_p = 195$, $\ell_n = 1024$, $\ell = 1000$, $k = k' = 50$, $\ell_N = 1024$, $\mathrm{v} = 3$. In this case, there will be only three BB signatures over elliptic curves of size 181 bits and users will select RSA moduli from $S' = S(2^{999}, 2^{481})$. Note that this will result to RSA moduli of the special form $2^{999} \pm t$, something that according to [30] does not appear to give an advantage to known factoring methods (nevertheless such tighter parameter selections should be used with caution). In this setting the total signature size drops to 24 Kb.

We note that the the parameter selection of ℓ_p in [28] (who used $\mathrm{v} = 1$) was incorrect due to an oversight that resulted in weak keys. A possible parameter selection for $\mathrm{v} = 1$ (that employs special RSA keys as above) is $\ell_p = 483$, $\ell_n = 1024$, $\ell = 1000$, $k = k' = 50$ and $\ell_N = 1024$; nevertheless, increasing v as suggested here, allows one to use smaller curves something that in turn allows for faster operations.

5 Proof of Security

The proof of security is described here, it relies on the random oracle model (we prove the group signature rather than the interactive identity escrow variant of the scheme).

Proposition 5.1 *Suppose that one possesses the witnesses $\theta_z, \theta_x, \theta_{xz}, \theta_r, \theta_{rz}, \theta_{x_1}, \theta_{x_2}, \theta_y, \theta_{y'}, \theta_{y''}, \theta_{xy'}$, $\theta_{ry'}, \theta_{yx_2}, \theta_d$ for the relations of figure 2 regarding the values $T_{1,1}, \ldots, T_{1,\mathrm{v}}, T_2, T_3, T_4, C_0, C_1, C_2$ as well as it holds $C_2 \leq N^2/2$. Then the following hold true: (i) Each of the* v *pairs $(T_{1,t} g_{1,t}^{-\theta_z}, \theta_r \bmod p_t)$ for $t = 1, \ldots, \mathrm{v}$ is a BB signature on θ_x under the public-key $g_{1,t}, g_{2,t}, w_t, v_t, e_t : \mathbb{G}_{1,t} \times \mathbb{G}_{2,t} \rightarrow \mathbb{G}_t$. (ii) $\langle C_0, C_1, C_2 \rangle$ is a valid ciphertext encrypting θ_x.*

Proof. For notational simplicity set $\theta_\xi = \xi$ for all possible ξ and we drop the subscripts t for $t = 1, \ldots, \mathrm{v}$. Let us consider now the relation that enforces the signature validity from figure 2. We have that

$$e(T_1, w) e(T_1, g_2)^x e(T_1, v)^r e(g_1, g_2)^{-xz} e(g_1, v)^{-rz} e(g_1, w)^{-z} =$$

$$= e(T_1, wg_2^x v^r)e(g_1^{-z}, wg_2^x v^r) = e(T_1 g_1^{-z}, wg_2^x v^r)$$

by employing the properties of the bilinear map. Then we have that if we call $\sigma = T_1 g_1^{-z}$ it holds that $e(\sigma, wg_2^x v^r) = e(g_1, g_2)$, i.e., $\sigma, r$is a valid BB signature on x.

Finally, observe that $(C_0)^2 = (G^2)^t \bmod N^2, (C_1)^2 = (H_1^2)^t(1+N)^{2x}$, so clearly it holds that $C_1^2 C_0^{-2a_1} = (1+N)^{2x}$ and as a result the decryption will be either x or $\perp$. Moreover observe that $(C_2)^2 = (H_2 H_3^{\mathcal{H}(\mathsf{hk}, C_0, C_1)})^{2d}$. The ciphertext C_0, C_1, C_2 would be a valid ciphertext provided that $C_2 = ||C_2||$ as well as $C_0^2 = C_0^{2(a_2 + a_3 \mathcal{H}(\mathsf{hk}, C_0, C_1))}$. Now it holds that $C_2^2 = (H_2 H_3^{\mathcal{H}(\mathsf{hk}, C_0, C_1)})^{2t} = G^{2da_2 + 2da_3 \mathcal{H}(\mathsf{hk}, C_0, C_1)} = C_0^{2(a_2 + a_3 \mathcal{H}(\mathsf{hk}, C_0, C_1))}$. ■

Theorem 5.2 *The signature of knowledge that specifies the* SIGN *algorithm satisfies: completeness, special soundness under the Strong-RSA assumption and statistical honest verifier zero-knowledge.*

*Proof.*Consider two accepting conversations as folows:

$$\langle T_{1,1}, \ldots, T_{1,\mathrm{v}}, T_2, T_3, T_4, C_0, C_1, C_2, c, s_z, s_x, s_{xz}, s_r, s_{rz}, s_{x_1}, s_{x_2}, s_y, s_{y'}, s_{y''}, s_{xy'}, s_{ry'}, s_{yx_2}, s_d \rangle$$

and

$$\langle T_{1,1}, \ldots, T_{1,\mathrm{v}}, T_2, T_3, T_4, C_0, C_1, C_2, c^*, s_z^*, s_x^*, s_{xz}^*, s_r^*, s_{rz}^*, s_{x_1}^*, s_{x_2}^*, s_y^*, s_{y'}^*, s_{y''}^*, s_{xy'}^*, s_{ry'}^*, s_{yx_2}^*, s_d^* \rangle$$

We will show how to recover the corresponding witnesses for the relation of figure 2. First observe that,

$$g^{s_{y'}} f_1^{s_z} T_3^c = g^{s_{y'}^*} f_1^{s_z^*} T_3^{c^*} \tag{1}$$

from which we obtain that

$$T_3^{c-c^*} = g^{s_{y'}^* - s_{y'}} f_1^{s_z^* - s_z} \tag{2}$$

Using lemma 5.9 we obtain that based on the Strong-RSA assumption, $c - c^*$ should divide both $s_{y'}^* - s_{y'}$ and $s_z^* - s_z$. As a result $T_3^{c-c^*} = (g^{\frac{s_{y'}^* - s_{y'}}{c-c^*}} f_1^{\frac{s_z^* - s_z}{c^* - c}})^{c-c^*}$; now given that (i) the order of T_3 divides $\lambda(n)$, (ii) n is a safe RSA modulus, (iii) c, c^* is a 2^k numbers where k is smaller than $\lfloor \log_2 n \rfloor$, we have that $T_3^2 = (g^{\frac{s_{y'}^* - s_{y'}}{c-c^*}} f_1^{\frac{s_z^* - s_z}{c-c^*}})^2$ which is easily seen to imply that

$$T_3 = \pm g^{\frac{s_{y'}^* - s_{y'}}{c-c^*}} f_1^{\frac{s_z^* - s_z}{c-c^*}} \tag{3}$$

(otherwise we can turn the prover into a factorization algorithm). Using the above we set $\theta_z = \frac{s_z^* - s_z}{c-c^*}$ and $\theta_{y'} = \frac{s_{y'}^* - s_{y'}}{c-c^*}$.

Next we consider the following equation:

$$g^{s_{y''}} f_1^{s_x - c2^{\ell'}} f_2^{s_{x_2} - c2^{\ell/2-1}} f_3^{s_r} f_4^{s_d} T_4^c = g^{s_{y''}^*} f_1^{s_x^* - c2^{\ell'}} f_2^{s_{x_2}^* - c2^{\ell/2-1}} f_3^{s_r^*} f_4^{s_d^*} T_4^{c^*}$$

which implies that,

$$T_4^{c^*-c} = g^{s_{y''}^* - s_{y''}} f_1^{s_x^* - s_x + (c-c^*)2^{\ell'}} f_2^{s_{x_2}^* - s_{x_2} + (c-c^*)2^{\ell/2-1}} f_3^{s_r^* - s_r} f_4^{s_d^* - s_d}$$

Using similar reasoning as in equation 2 we obtain that

$$T_4 = \pm g^{\frac{s^*_{y''}-s_{y''}}{c-c^*}} f_1^{\frac{s^*_x-s_x}{c-c^*}+2^{\ell'}} f_2^{\frac{s^*_{x_2}-s_{x_2}}{c-c^*}+2^{\ell/2-1}} f_3^{\frac{s^*_r-s_r}{c-c^*}} f_4^{\frac{s^*_d-s_d}{c-c^*}}$$

We set $\theta_{y''} = \frac{s^*_{y''}-s_{y''}}{c-c^*}$, $\theta_x = \frac{s^*_x-s_x}{c-c^*} + 2^{\ell'}$, $\theta_{x_2} = \frac{s^*_{x_2}-s_{x_2}}{c-c^*} + 2^{\ell/2-1}$, $\theta_r = \frac{s^*_r-s_r}{c-c^*}$, $\theta_d = \frac{s^*_d-s_d}{c-c^*}$. Observe that based on the fact that $s^*_x, s_x \in \pm\{0,1\}^{\mu'+k+k'+1}$ it will hold that with overwhelming probability, $\theta_x \in S(2^{\ell'}, 2^{\mu'+k+k'}) = S$. In a similar fashion it can be shown that $\theta_{x_2} \in S_f$.

Next we have that,

$$T_3^{-s_x+c2^{\ell'}} g^{s_{xy'}} f_1^{s_{xz}} = T_3^{-s^*_x+c^*2^{\ell'}} g^{s^*_{xy'}} f_1^{s^*_{xz}}$$

which implies that,

$$T_3^{s^*_x-s_x+(c-c^*)2^{\ell'}} = g^{s^*_{xy'}-s_{xy'}} f_1^{s^*_{xz}-s_{xz}}$$

and conditioning on the steps we have executed so far, it holds that

$$(T_3^{\theta_x})^{c-c^*} = g^{s^*_{xy'}-s_{xy'}} f_1^{s^*_{xz}-s_{xz}}$$

Using again lemma 5.9 we obtain that $c-c^*$ divides $s^*_{xy'} - s_{xy'}$ and $s^*_{xz} - s_{xz}$ simultaneously and as a result,

$$T_3^{\theta_x} = \pm g^{\frac{s^*_{xy'}-s_{xy'}}{c-c^*}} f_1^{\frac{s^*_{xz}-s_{xz}}{c-c^*}} \tag{4}$$

We set $\theta_{xy'} = \frac{s^*_{xy'}-s_{xy'}}{c-c^*}$ and $\theta_{xz} = \frac{s^*_{xz}-s_{xz}}{c-c^*}$ and based on the fact that T_3 is already specified, using eqations 3 and 4 we can obtain the following relation :

$$(g^{\theta_{y'}} f_1^{\theta_z})^{\theta_x} = g^{\theta_{xy'}} f_1^{\theta_{xz}} \Longrightarrow g^{\theta_x\theta_{y'}} f_1^{\theta_x\theta_z} = g^{\theta_{xy'}} f_1^{\theta_{xz}}$$

something that can be easily seen to imply that $\theta_{xy'} = \theta_x\theta_{y'}$ and $\theta_{xz} = \theta_x\theta_z$.

We proceed now to the next relation,

$$T_3^{-s_r} g^{s_{ry'}} f_1^{s_{rz}} = T_3^{-s^*_r} g^{s^*_{ry'}} f_1^{s^*_{rz}}$$

proceeding in identical fashion as above we obtain that $\theta_{ry'} = \theta_r\theta_{y'}$ and $\theta_{rz} = \theta_r\theta_z$.

We proceed now to the next relation; we have that:

$$g^{s_y} f_1^{s_{x_1}-c2^{\ell''}} T_2^c = g^{s^*_y} f_1^{s^*_{x_1}-c2^{\ell''}} T_2^{c^*} \tag{5}$$

which implies that

$$T_2^{c-c^*} = g^{s^*_y-s_y} f_1^{s^*_{x_1}-s_{x_1}+(c-c^*)2^{\ell''}} \tag{6}$$

Using similar reasoning as in equation 2 we obtain that

$$T_2 = \pm g^{\frac{s^*_y-s_y}{c-c^*}} f_1^{\frac{s^*_{x_1}-s_{x_1}}{c-c^*}+2^{\ell''}}$$

We set the witnesses $\theta_y = \frac{s^*_y-s_y}{c-c^*}$ and $\theta_{x_1} = \frac{s^*_{x_1}-s_{x_1}}{c-c^*} + 2^{\ell''}$. Observe that due to the condition $s_{x_1}, s^*_{x_1} \in \pm\{0,1\}^{\mu'_f+k+k'+1}$ it follows that with overwhelming probability $\theta_{x_1} \in S_f$.

We proceed to the next relation:

$$T_2^{-s_{x_2}+c2^{\ell/2-1}} g^{s_{yx_2}} f_1^{s_x-c2^{\ell'}} = T_2^{-s^*_{x_2}+c^*2^{\ell/2-1}} g^{s^*_{yx_2}} f_1^{s^*_x-c2^{\ell'}} =$$

This implies that

$$T_2^{s^*_{x_2}-s_{x_2}+(c-c^*)2^{\ell/2-1}} = g^{s^*_{yx_2}-s_{yx_2}} f_1^{s^*_x-s_x+(c-c^*)2^{\ell'}}$$

conditioning on the already reconstructed values we have that the above is written as :

$$T_2^{\theta_{x_2}(c-c^*)} = g^{s^*_{yx_2}-s_{yx_2}} f_1^{\theta_x(c-c^*)}$$

and using similar arguments as in equation 2

$$T_2^{\theta_{x_2}} = \pm g^{\theta_{yx_2}} f_1^{\theta_x}$$

Using the fact that $T_2 = \pm g^{\theta_y} f_1^{\theta_{x_1}}$ (as shown above) we can easily derive that it must be the case that $\theta_{yx_2} = \theta_y \theta_{x_2}$ and $\theta_x = \theta_{x_1}\theta_{x_2}$.

Next we consider the v relations over the elliptic curves. Observe that all witnesses for these relations have been reconstructed already over the integers and there are no witnesses dependent on the parameter $t = 1, \ldots, \mathrm{v}$. For convenience and readability we drop the subscripts t and we have that

$$e(T_1, g_2)^{s_x - c2^{\ell'}} e(T_1, v)^{s_r} e(g_1, g_2)^{-s_{xz}} e(g_1, v)^{-s_{rz}} e(g_1, w)^{-s_z} e(g_1, g_2)^{c} e(T_1, w)^{-c}$$
$$= e(T_1, g_2)^{s^*_x - c^*2^{\ell'}} e(T_1, v)^{s^*_r} e(g_1, g_2)^{-s^*_{xz}} e(g_1, v)^{-s^*_{rz}} e(g_1, w)^{-s^*_z} e(g_1, g_2)^{c^*} e(T_1, w)^{-c^*}$$

which implies:

$$e(T_1, g_2)^{s^*_x - s_x + (c-c^*)2^{\ell'}} e(T_1, v)^{s^*_r - s_r} e(T_1, w)^{c-c^*}$$
$$= e(g_1, g_2)^{c-c^*} e(g_1, g_2)^{s^*_{xz} - s_{xz}} e(g_1, v)^{s^*_{rz} - s_{rz}} e(g_1, w)^{s^*_z - s_z}$$

conditional on all the above witness reconstructions we have that:

$$e(T_1, g_2)^{\theta_x} e(T_1, v)^{\theta_r} e(T_1, w) = e(g_1, g_2) e(g_1, g_2)^{\theta_x\theta_z} e(g_1, v)^{\theta_r\theta_z} e(g_1, w)^{\theta_z}$$

which implies that

$$e(T_1, w g_2^{\theta_x} v^{\theta_r}) e(g_1, g_2) = e(g_1^{\theta_z}, w g_2^{\theta_x} v^{\theta_r}) \Longrightarrow e(T_1 g_1^{-\theta_z}, w g_2^{\theta_x} v^{\theta_r}) = e(g_1, g_2)$$

This implies that $T_1 g_1^{-\theta_z \bmod p}, r(\bmod p)$ would be a valid signature for the underlying BB-signature scheme.

Finally, we turn to the relations regarding the public-key encryption. First we have that

$$C_0^{2c} G^{2s_d} = C_0^{2c^*} G^{2s^*_d}$$

which implies,

$$(G^2)^{s^*_d - s_d} = (C_0^2)^{c-c^*}$$

Since we have already reconstructed $\theta_d = \frac{s^*_d - s_d}{c-c^*}$ we can rewrite the above as:

$$(G^2)^{\theta_d(c-c^*)} = (C_0^2)^{c-c^*}$$

Note that G generates the subgroup of squares within $\mathbb{Z}^*_{N^2}$ that are simultaneously N-th residues. Moreover C_0^2 is of course a square inside $\mathbb{Z}^*_{N^2}$. Given that $P, Q, P', Q' > 2^k$ we have that $c - c^*$ cancels in the above equation and thus it holds that $C_0^2 = G^{2\theta_d} \bmod N^2$.

In a similar fashion we have that

$$C_1^{2c} H_1^{2s_d} (1+N)^{2(s_x - c2^{\ell'})} = C_1^{2c^*} H_1^{2s_d^*} (1+N)^{2(s_x^* - c^* 2^{\ell'})}$$

From which we obtain that,

$$(C_1^2)^{c-c^*} = (H_1^2)^{s_d^* - s_d} ((1+N)^2)^{s_x^* - s_x + 2^{\ell'}(c-c^*)}$$

and using the already reconstructed witnesses we obtain that,

$$(C_1^2)^{c-c^*} = (H_1^2)^{(c-c^*)\theta_d} ((1+N)^2)^{\theta_x (c-c^*)}$$

Similarly as before we obtain that

$$C_1^2 = H_1^{2\theta_d} (1+N)^{2\theta_x}$$

Finally, we have that

$$C_2^{2c} (H_2^2 H_3^{2\mathcal{H}(\mathsf{hk}, C_0, C_1)})^{s_d} = C_2^{2c^*} (H_2^2 H_3^{2\mathcal{H}(\mathsf{hk}, C_0, C_1)})^{s_d^*}$$

which implies that,

$$(C_2^2)^{c-c^*} = (H_2^2 H_3^{2\mathcal{H}(\mathsf{hk}, C_0, C_1)})^{s_d^* - s_d}$$

given that we have that $s_d^* - s_d = (c - c^*)\theta_d$ we rewrite the above equation as,

$$(C_2^2)^{c-c^*} = (H_2^2 H_3^{2\mathcal{H}(\mathsf{hk}, C_0, C_1)})^{\theta_d (c-c^*)}$$

from which we obtain that

$$(C_2)^2 = (H_2^2 H_3^{2\mathcal{H}(\mathsf{hk}, C_0, C_1)})^{\theta_d}$$

It is easy to see that the ciphertext (C_0, C_1, C_2) with the above structure in conjunction to the relation $C_2 \leq N^2/2$ will decrypt to the value θ_x. This completes the soundness proof.

Regarding the statistical honest verifier zero-knowledge consider the following simulator for any randomly selected $T_{1,1}, \ldots, T_{1,\mathrm{v}}, T_2, T_3, T_4, C_0, C_1, C_2$ from the respective groups and then form an accepting conversation as follows:

$$\langle T_{1,1}, \ldots, T_{1,\mathrm{v}}, T_2, T_3, T_4, C_0, C_1, C_2, c, s_z, s_x, s_{xz}, s_r, s_{rz}, s_{x_1}, s_{x_2}, s_y, s_{y'}, s_{y''}, s_{xy'}, s_{ry'}, s_{yx_2}, s_d \rangle$$

where $c \leftarrow_R \{0,1\}^k$, and $s_z \leftarrow_R \pm\{0,1\}^{\mathrm{v}\ell_p + k + k'}$, $s_x \leftarrow_R \pm\{0,1\}^{\mu' + k + k'}$, $s_{xz} \leftarrow_R \pm\{0,1\}^{\mathrm{v}\ell_p \mu' + k + k'}$, $s_r \leftarrow_R \pm\{0,1\}^{\mathrm{v}\ell_p + k + k'}$, $s_{rz} \leftarrow_R \pm\{0,1\}^{2\mathrm{v}\ell_p + k + k'}$ $s_{x_1} \leftarrow_R \pm\{0,1\}^{\mu'_f + k + k'}$, $s_{x_2} \leftarrow_R \pm\{0,1\}^{\ell/2 - 1 + k + k'}$, $s_y, \rho_{y'}, \rho_{y''} \leftarrow_R \pm\{0,1\}^{\ell_n - 2 + k + k'}$, $s_{xy'} \leftarrow_R \pm\{0,1\}^{\ell_n - 2 + \mu' + k + k'}$ $s_{ry'} \leftarrow_R \pm\{0,1\}^{\ell_n - 2 + \mathrm{v}\ell_p + k + k'}$ $s_{yx_2} \leftarrow_R \pm\{0,1\}^{\ell_n - 2 + \ell/2 - 1 + k + k'}$ $s_t \leftarrow_R \pm\{0,1\}^{\ell_N - 2 + k + k'}$ are selected.

and the values $R_1, \ldots, R_{10}$ will be defined as they are determined inside the HASH application in the verification algorithm of the group signature. Applying lemma 5.10 on the s-values above we obtain the fact that the simulator is statistically close to real conversations. ■

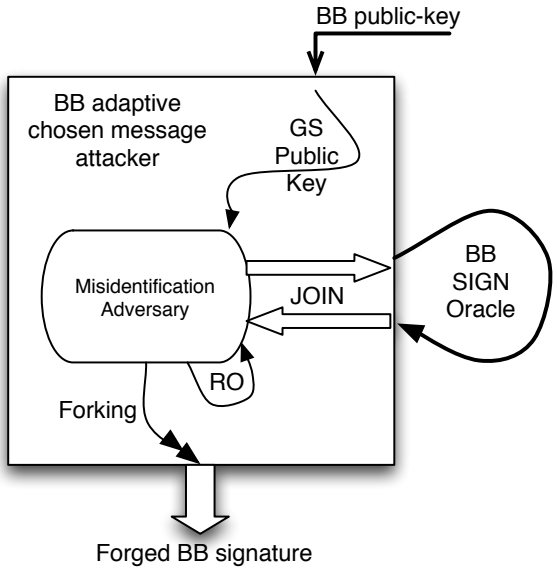

Figure 3: Overview of the reduction in the proof of security for the misidentification attack.

5.1 Misidentification

Theorem 5.3 *Any misidentification attacker in the random oracle model against our group signature can be transformed to an adaptive chosen message attacker in the standard model against the BB signature assuming the Strong-RSA assumption.*

Proof. — see figure 3 for an overview of the transformation performed in this proof. In this proof we will assume that $\mathrm{v} = 1$; the case where $\mathrm{v} > 1$ is identical (in this case, the adversary would break at least one of the underlying BB-signature instantiations).

Let $\mathcal{A}$ be a misidentification adversary as specified in the misidentification attack game that also has access to a random oracle $\mathcal{H}$. We will reduce this adversary to an adaptive chosen message adversary for the BB digital signature.

Let us describe now an adversary $\mathcal{B}$ for the Boneh-Boyen digital signature. First the adversary is given the public-key of the signature which includes the description of the groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ as well as the description of the bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ and the values g_1, g_2 and $w, v \in \mathbb{G}_2$. The forger $\mathcal{B}$ using this information samples all remaining public parameters for the group signature including the parameters for the CCA2 encryption $\mathsf{p4}$, i.e., the values N, G. $\mathcal{B}$ now forms the public-key of the group signature scheme:

$$\mathcal{Y} = \langle g_1, g_2, w, v, \mathrm{desc}(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}||e)||\langle \mathsf{UOHF}, g, f, n, N, G, H_1, H_2, H_3, \mathsf{hk}\rangle$$

note that $\mathcal{B}$ knows the secret-key of the GM in the group signature with the exception of the signing key for the BB signature, i.e., $\mathcal{B}$ does not know $\gamma = \log_{g_2} w$ and $\delta = \log_{g_2} v$.

Prior to beginning the simulation of $\mathcal{A}$ the $\mathcal{B}$ initializes a table $\mathcal{H}$ for the simulation of the random oracle that is employed by $\mathcal{A}$. The public-state St is initialized as described in the attack game (empty).

Now, $\mathcal{B}$ starts the simulation of the adversary $\mathcal{A}$; $\mathcal{B}$ has the following queries to answer:

Random Oracle queries: these are simulated using the table $\mathcal{H}$ as usual: if the query is already on the table $\mathcal{B}$ returns the corresponding value; if not, $\mathcal{B}$ samples a random element of $\{0,1\}^k$, it places it to the table and returns it.

$\texttt{READ}_{St}$: $\mathcal{B}$ returns the whole contents of the St strings.

$\texttt{JOIN}_{\leftrightarrow GM}$; $\texttt{Start}-\texttt{Session}$ queries: $\mathcal{B}$ receives the start session command from the adversary. As the GM does not goes first in the JOIN protocol, $\mathcal{B}$ simply selects a session id and returns is to the adversary.

$\texttt{JOIN}_{\leftrightarrow GM}$; $\texttt{Advance}-\texttt{Session}$ queries: $\mathcal{B}$ takes a message $\langle s, x\rangle$ from the adversary. Recall that the JOIN protocol has only two communication flows. $\mathcal{B}$ returns fail in the following cases: (i) s is not a valid session identifier, (ii) s is a valid session identifier but x is not an integer in the range S'; in this case the session s is marked by $\mathcal{B}$ as invalid and is terminated. Now if x is an integer in the range S', $\mathcal{B}$ submits x to his signing oracle for the BB signature, and obtains through it a signature (σ, r) such that $e(\sigma, wg_2^x v^r) = e(g_1, g_2)$. $\mathcal{B}$ returns to the adversary $\langle i, \sigma, r\rangle$ where $i = \max St_{users} + 1$ and updates St_{users} and $St_{join-trans}$ accordingly.

OPEN: $\mathcal{B}$ knows the secret-keys for the encryption thus opening queries are also straightforward to answer.

Continuing as above the adversary terminates by producing a message m and a valid group signature $\langle m, \sigma\rangle$ where, σ is equal to

$$\langle T_{1,1}, T_2, T_3, T_4, C_0, C_1, C_2, c, s_z, s_x, s_{xz}, s_r, s_{rz}, s_{x_1}, s_{x_2}, s_y, s_{y'}, s_{y''}, s_{xy'}, s_{ry'}, s_{yx_2}, s_d\rangle$$

Assume now that the above passes the verification test and moreover it satisfies $\texttt{OPEN}(m, \sigma, \mathcal{Y}, \mathcal{S}, St) = \perp$ (i.e., the adversary wins the misidentification attack game); note that OPEN employs only the encryption secret-key and as a result it can be applied by $\mathcal{B}$. Now using rewinding and the forking lemma of Pointcheval and Stern [32], we can obtain with non-negligible probability a σ^* on the same message m, such that σ^* is equal to

$$\langle T_{1,1}, T_2, T_3, T_4, C_0, C_1, C_2, c^*, s_z^*, s_x^*, s_{xz}^*, s_r^*, s_{rz}^*, s_{x_1}^*, s_{x_2}^*, s_y^*, s_{y'}^*, s_{y''}^*, s_{xy'}^*, s_{ry'}^*, s_{yx_2}^*, s_d^*\rangle$$

so that the above is also a valid signature of knowledge. Now based on theorem 5.2 we can reconstruct the witnesses (this is where the Strong-RSA assumption is needed) and based on proposition 5.1 we have that $\langle T_{1,1} g_{1,1}^{-\theta_z}, \theta_r\rangle$ is a valid BB signature on θ_x. Now observe that $\texttt{OPEN}(m, \sigma, \mathcal{Y}, \mathcal{S}, St) = \perp$; due to proposition 5.1, the ciphertext C_0, C_1, C_2 must be valid and decrypts to the value θ_x; since $\texttt{OPEN}(m, \sigma, \mathcal{Y}, \mathcal{S}, St) = \perp$ this means that θ_x is not among the x-values that were submitted by the adversary in the concurrent protocol JOIN dialogs with the interface. As a result $\mathcal{B}$ never queried the value θ_x to its signing oracle. $\mathcal{B}$ terminates by returning $\langle \theta_x; T_{1,1} g_{1,1}^{-\theta_z}, \theta_r\rangle$ as a forgery against the BB signature scheme. Note that due to the fact that the integer range S is selected appropriately (see the parameter selection p3) it cannot be the case that the adversary submitted some value n_0 to the signing oracle for which it holds $\theta_x \equiv_{p_1} n_0$ but $\theta_x \neq n_0$ as integers (with Chinese remaindering this carries to the case $\mathrm{v} > 1$.

From the above it follows that $\mathcal{B}$ outputs a forged signature and thus it is an adaptive chosen message attacker against the BB signature. ■

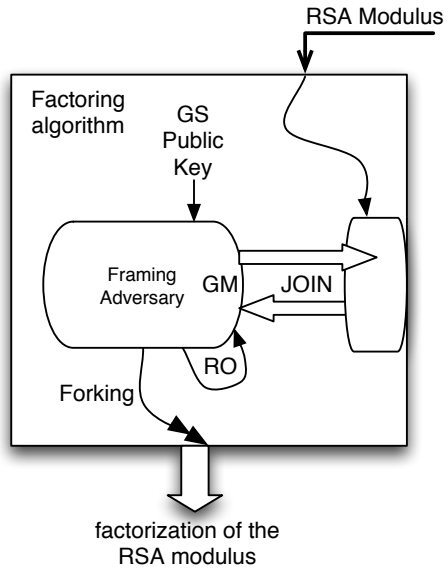

Figure 4: Overview of the reduction in the proof of security for the framing attack.

5.2 Framing

Theorem 5.4 *Any framing attacker in the random oracle model against our group signature can be transformed to a factoring algorithm in the standard model assuming the Strong-RSA assumption.*

Proof. — consult figure 4 for an overview of the transformation.

Let $\mathcal{A}$ be a framing adversary as specified in the framing attack game that also has access to a random oracle $\mathcal{H}$. We will transform this adversary to a factoring algorithm $\mathcal{B}$ (that will work for RSA moduli according to the specifications of the selection of x in step 1 of the JOIN dialog, i.e., $x \in S' - S_f$ such that $x_1, x_2 \in S'_f$). Again without loss of generality we assume that $\mathrm{v} = 1$.

We describe below how $\mathcal{B}$ operates given an integer $n_0 \in S'$. $\mathcal{B}$ will sample the public-key $\mathcal{Y}$ and the secret-key $\mathcal{S}$ of the system exactly as it is specified in the SETUP algorithm.

$$\mathcal{Y} = \langle g_1, g_2, w, v, \mathrm{desc}(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}||e)||\langle \mathsf{UOHF}, g, f, n, N, G, H_1, H_2, H_3, \mathsf{hk}\rangle$$

Prior to beginning the simulation of $\mathcal{A}$, $\mathcal{B}$ will initialize a table $\mathcal{H}$ for the simulation of the random oracle that is employed by $\mathcal{A}$. The public-state St is initialized as described in the attack game (empty).

Now, $\mathcal{B}$ starts the simulation of the adversary $\mathcal{A}$; suppose that the adversary $\mathcal{A}$ makes q_0 JOIN dialog initiations. $\mathcal{B}$ selects $j_0 \leftarrow_R \{1, \ldots, q_0\}$. Then, in the simulation of $\mathcal{A}$, $\mathcal{B}$ has the following queries to answer:

Random Oracle queries: these are simulated using the table $\mathcal{H}$ as usual: if the query is already on the table $\mathcal{B}$ returns the corresponding value; if not, $\mathcal{B}$ samples a random element of $\{0,1\}^k$, it places it to the table and returns it.

$\mathtt{READ}_{St}$: $\mathcal{B}$ returns the whole contents of the St strings.

$\mathtt{MODIFY}_{St}$: $\mathcal{B}$ allows $\mathcal{A}$ to modify join transcripts and identity tags to the St_{users} and $St_{join-trans}$ according to the stated restrictions. Note that $\mathcal{A}$ is not allowed to reuse existing identity tags or modify the tags or transcripts of any of the innocent users.

$\texttt{JOIN}_{\leftrightarrow User}$; $\texttt{Start}-\texttt{Session}$ queries: $\mathcal{B}$ receives the start session command from the adversary. Suppose this is the j-th time that the adversary issued a $\texttt{Start}-\texttt{Session}$ query. As the user does go first in JOIN protocol, $\mathcal{B}$ selects x_j an RSA modulus in S', unless it is the case that $j = j_0$ where $\mathcal{B}$ selects $x_j = n_0$ (the challenge RSA modulus). $\mathcal{B}$ also selects a session id and returns the id and x_j to the adversary.

$\texttt{JOIN}_{\leftrightarrow User}$; $\texttt{Advance}-\texttt{Session}$ queries: $\mathcal{B}$ takes a message $\langle s, msg\rangle$ from the adversary. Recall that the JOIN protocol has only two communication flows. $\mathcal{B}$ returns fail in the following cases: (i) s is not a valid session identifier, (ii) s is a valid session identifier but msg is *not* a tuple of the form $\langle i, \sigma, r\rangle$ with $e(\sigma, wg_2^{x_j}v^r) = e(g_1, g_2)$ or it holds that $i \in St_{users}$; in this case the session s is marked by $\mathcal{B}$ as invalid and is terminated. Otherwise $\mathcal{B}$ writes in the public-state St the user i, i.e., it modifies $St_{users} = St_{users}||\langle i\rangle$ and $St_{\mathsf{join-trans}} = St_{\mathsf{join-trans}}||\langle i, x_j, \sigma, r\rangle$.

Note that we do not require from the adversary to follow the same numbering of users that the GM performs during normal protocol executions.

Continuing as above the adversary terminates by producing a message m and a valid group signature m, σ where σ is equal to:

$$\langle T_{1,1}, T_2, T_3, T_4, C_0, C_1, C_2, c, s_z, s_x, s_{xz}, s_r, s_{rz}, s_{x_1}, s_{x_2}, s_y, s_{y'}, s_{y''}, s_{xy'}, s_{ry'}, s_{yx_2}, s_d\rangle$$

that passes the verification test and moreover it satisfies $\texttt{OPEN}(m, \sigma, \mathcal{Y}, \mathcal{S}, St) \in St^{\mathcal{I}}_{users}$; note that OPEN employs only the public-key encryption secret-key and as a result it can be applied by $\mathcal{B}$. Now using rewinding and the forking lemma of Pointcheval and Stern [32], we can obtain with non-negligible probability a signature σ^* on the same message m, where σ^* is equal to

$$\langle T_{1,1}, T_2, T_3, T_4, C_0, C_1, C_2, c^*, s^*_z, s^*_x, s^*_{xz}, s^*_r, s^*_{rz}, s^*_{x_1}, s^*_{x_2}, s^*_y, s^*_{y'}, s^*_{y''}, s^*_{xy'}, s^*_{ry'}, s^*_{yx_2}, s^*_d\rangle$$

so that the above is also a valid signature of knowledge. Now based on theorem 5.2 we can reconstruct the witnesses (this is where the Strong-RSA assumption is needed); based on the properties of the reconstruction we have that $\theta_{x_1}, \theta_{x_2}$ is a factorization of θ_x so that $\theta_{x_1}, \theta_{x_2} \in S_f$ (this range constraint is enforced by the proof).

Now recall that $\texttt{OPEN}(m, \sigma, \mathcal{Y}, \mathcal{S}, St) \in St^{\mathcal{I}}_{users}$; this means that θ_x is among the x-values that were selected by $\mathcal{B}$ in the concurrent protocol JOIN dialogs and as a result with probability $1/q_0$ it holds that $\theta_x = n_0$. In this case, $\theta_{x_1}, \theta_{x_2}$ is a non-trivial split of n_0 (due to the fact that $n_0 \notin S_f$) and thus $\mathcal{B}$ terminates successfully by returning $\theta_{x_1}, \theta_{x_2}$. ■

5.3 Anonymity

Theorem 5.5 *Any anonymity adversary against our group signature in the random oracle model can be transformed to a CCA2 attacker against the public-key encryption that is employed in our scheme; this is conditional on the validity of (i) the assumption that the digital signature scheme employed (BB-signature) satisfies strong existential unforgeability (which is true assuming the Strong Diffie Hellman Assumption). (ii) the DLOG assumption over the subgroup of $2N$-th residues inside $\mathbb{Z}^*_{N^2}$.*

Proof. — consult figure 5 for an overview of the transformation described in this proof.

Let $\mathcal{A}$ be an anonymity adversary as specified in the anonymity attack game $G^{\mathcal{A}}_{\mathsf{anon}}(1^\nu)$. Note that $\mathcal{A}$ has access to the random oracle $\mathcal{H}$. We will transform $\mathcal{A}$ to a CCA2 adversary $\mathcal{B}$ against the CCA2 public-key encryption that is employed for the OPEN algorithm. Without loss of generality we assume that $\mathrm{v} = 1$. Below we describe the operation of this algorithm $\mathcal{B}$:

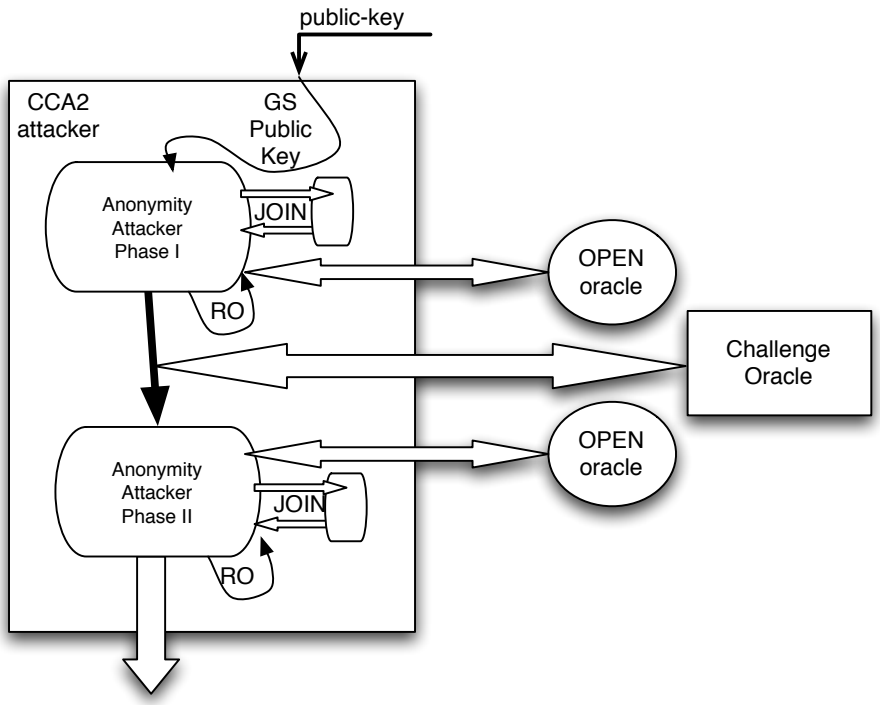

Figure 5: Overview of the reduction in the proof of security for the anonymity attack.

Initially, $\mathcal{B}$ is given the public-key $N, G, H_1, H_2, H_3, \mathrm{desc}(\mathsf{UOHF}), \mathsf{hk}$ for the encryption scheme. $\mathcal{B}$ will use these values to form the public-key of the group signature scheme. In particular, $\mathcal{B}$ will select

$$\mathcal{Y} = \langle g_1, g_2, w, v, \mathrm{desc}(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}||e)||\langle \mathsf{UOHF}, g, f, n, N, G, H_1, H_2, H_3, \mathsf{hk}\rangle$$

as in the SETUP protocol with the exception of the public-key of the encryption. Moreover, $\mathcal{B}$ will initialize St to empty and initializes tables for the random oracle simulation as usual.

The simulation of $\mathcal{A}$ will proceed as follows: The play phase of $\mathcal{A}$ will correspond to the plaintext selection of the CCA2 adversary $\mathcal{B}$. $\mathcal{B}$ has to simulate the following oracle queries of $\mathcal{A}$:

Random Oracle queries: these are simulated using the table $\mathcal{H}$ as usual: if the query is already on the table, $\mathcal{B}$ returns the corresponding value; if not, $\mathcal{B}$ samples a random element of $\{0,1\}^k$, it places it to the table and returns it.

$\mathtt{READ}_{St}$: $\mathcal{B}$ returns the whole contents of the St strings.

$\mathtt{JOIN}_{\leftrightarrow GM}$; $\mathtt{Start}-\mathtt{Session}$ queries: $\mathcal{B}$ receives the start session command from the adversary. As the GM does not go first in the JOIN protocol, $\mathcal{B}$ simply selects a session id and returns is to the adversary.

$\mathtt{JOIN}_{\leftrightarrow GM}$; $\mathtt{Advance}-\mathtt{Session}$ queries: $\mathcal{B}$ takes a message $\langle s, x\rangle$ from the adversary. Recall that the JOIN protocol has only two communication flows. $\mathcal{B}$ returns fail in the following cases: (i) s is not a valid session identifier, (ii) s is a valid session identifier but x is not an integer in the range S'; in this case the session s is marked by $\mathcal{B}$ as invalid and is terminated. Now if x is an integer in the range S', $\mathcal{B}$ computes a BB signature for x ($\mathcal{B}$ is in possession of the signing key); as a result $\mathcal{B}$ obtains a signature (σ, r) such that $e(\sigma, wg_2^x v^r) = e(g_1, g_2)$. $\mathcal{B}$ returns to the adversary $\langle i, \sigma, r\rangle$ where $i = \max St_{users} + 1$ and updates St_{users} and $St_{join-trans}$ accordingly.

OPEN. A signature is given to be opened. First $\mathcal{B}$ verifies the signature; if verification fails, $\mathcal{B}$ returns $\perp$. Then, $\mathcal{B}$ parses the signature for the values C_0, C_1, C_2 and submits $\langle C_0, C_1, C_2\rangle$ to its open oracle. If the open oracle returns $\perp$ then $\mathcal{B}$ returns $\perp$ as well. Otherwise the oracle returns a plaintext x. $\mathcal{B}$ searches into $St_{\mathsf{join-trans}}$ for a join transcript of the form $\langle i, x, \sigma, r\rangle$ and if it is found it returns i; otherwise it returns $\perp$. Observe that this is indistinguishable operation from the point of view of the adversary to the operation of the regular attack game.

The simulation of the play terminates and $\mathcal{B}$ receives $\langle aux, m, \mathsf{cert}_1, \mathsf{sec}_1, \mathsf{cert}_2, \mathsf{sec}_2\rangle$; if $\mathcal{B}$ receives $\mathsf{cert}_i, \mathsf{sec}_i$ that are not valid certificates and secrets it terminates with $\perp$.

Observe the following now: the certificates returned by $\mathcal{A}$ are of the form: $\mathsf{cert}_1 = \langle x_1, \sigma_1, r_1\rangle$ and $\mathsf{cert}_2 = \langle x_2, \sigma_2, r_2\rangle$. Moreover, it must be that $x_1 \neq x_2$. We note that if it is the case $x_1 = x_2$ but $\mathsf{cert}_1 \neq \mathsf{cert}_2$ then this means that the adversary obtained two BB signatures on the same integer $x_1 = x_2$. As the signing oracle would never provide such a signature (it keeps a record of what values, users have submitted before) it must be the case that the adversary came up with two different signatures on the same message. This violates the strong existential unforgeability (see [1]) of the BB signature and thus it can be considered a negligible event.

The CCA2 adversary $\mathcal{B}$ terminates its guess stage by outputing $aux' = aux||\mathsf{cert}_1||\mathsf{sec}_1||\mathsf{cert}_2||\mathsf{sec}_2$ and x_1, x_2 as its choices for the two plaintexts for the CCA2 challenge.

Now $\mathcal{B}$, given the challenge ciphertext C_0, C_1, C_2, must build the group signature ψ_{gs} that will be the challenge for $\mathcal{A}$ and should incorporate C_0, C_1, C_2. The SIGN operation at step 5 is simulated by $\mathcal{B}$

as follows: $\mathcal{B}$ selects $T_{1,1}, T_2, T_3, T_4$ so that $T_{1,1} \leftarrow_R \mathbb{G}_1$, $T_2, T_3, T_4 \leftarrow_R \mathbb{Z}_n^*$. Using these values and by programming the random oracle appropriately, $\mathcal{B}$ forms the group signature ψ_{gs}.

It is easy to see that forming the challenge group signature ψ_{gs} in this fashion it is indistinguishable from actual group signatures of the signer that is selected by the adversary (there will also be some negligible statistical distance due to the commitments T_2, T_3, T_4 and the honest verifier zero-knowledge simulation of the signature). The challenge signature on the message m would be as follows:

$$\psi_{gs} = \langle T_{1,1}, T_2, T_3, T_4, C_0, C_1, C_2, c, s_z, s_x, s_{xz}, s_r, s_{rz}, s_{x_1}, s_{x_2}, s_y, s_{y'}, s_{y''}, s_{xy'}, s_{ry'}, s_{yx_2}, s_d \rangle$$

With this setup $\mathcal{B}$ proceeds with the simulation of the guess stage of the adversary $\mathcal{A}$. The operation proceeds in the same fashion as above with the usual restrictions: the `OPEN` oracle supplied to the adversary is restricted not to accept the same signature as the challenge signature. Suppose that the adversary supplies the following signature on a message m' for opening:

$$\psi' = \langle T'_{1,1}, T'_2, T'_3, T'_4, C'_0, C'_1, C'_2, c', s'_z, s'_x, s'_{xz}, s'_r, s'_{rz}, s'_{x_1}, s'_{x_2}, s'_y, s'_{y'}, s'_{y''}, s'_{xy'}, s'_{ry'}, s'_{yx_2}, s'_d \rangle$$

According to the restriction imposed to the adversary it holds that $(m, \psi_{gs}) \neq (m', \psi')$. Now, in the case that ψ' is a valid signature, with a simulation soundness argument, the restriction that $(m, \psi_{gs}) \neq (m', \psi')$ implies that $\langle C_0, C_1, C_2 \rangle \neq \langle C'_0, C'_1, C'_2 \rangle$ with overwhelming probability. Indeed if $\mathcal{A}$ is capable of producing a pair (m', ψ') so that ψ' is a valid signature on m and ψ' and ψ_{gs} share the same identification ciphertext $\langle C_0, C_1, C_2 \rangle$ this means that we will be able to compute discrete-logarithms over the subgroup of quadratic and N-th residues in $\mathbb{Z}_{N^2}^*$.

Based on the above it holds that the signature ψ' can be opened by $\mathcal{B}$ through his opening oracle. Finally, $\mathcal{B}$ terminates by returning the output of the adversary $\mathcal{A}$. It is easy to verify that $\mathcal{B}$ is a CCA2 adversary for the employed public-key encryption. ■

The above three theorems culminate to the following theorem:

Theorem 5.6 *Our group signature is correct and secure in the random oracle model under the assumptions: SDH, Linear-DDH, Strong-RSA and DCR assumptions.*

Proof. Security follows directly from theorems 5.3, 5.4 and 5.5 as well as the following known results: the public-key encryption employed is CCA2 secure assuming the DCR assumption [16], the BB-digital signature scheme is strongly existential unforgeable under the SDH assumption, [8]. Correctness as in definition 3.2 is easier to show and is omitted. ■

References

[1] J. H. An, Y. Dodis, and T. Rabin. On the security of joint signature and encryption. In L. R. Knudsen, editor, *EUROCRYPT*, volume 2332 of *Lecture Notes in Computer Science*, pages 83–107. Springer, 2002.

[2] G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik. A practical and provably secure coalition-resistant group signature scheme. In M. Bellare, editor, *Advances in Cryptology – CRYPTO ' 2000*, volume 1880 of *Lecture Notes in Computer Science*. International Association for Cryptologic Research, Springer, 2000.

[3] G. Ateniese and B. de Medeiros. Efficient group signatures without trapdoors. In *ASIACRYPT: Advances in Cryptology – ASIACRYPT: International Conference on the Theory and Application of Cryptology*, Lecture Notes in Computer Science. International Association for Cryptologic Research, Springer, 2003.

[4] G. Ateniese and G. Tsudik. Some open issues and new directions in group signatures. In M. Franklin, editor, *Financial cryptography: Third International Conference, FC '99, Anguilla, British West Indies, February 22–25, 1999: proceedings*, volume 1648 of *Lecture Notes in Computer Science*, pages 196–211. Springer-Verlag, 1999.

[5] N. Barić and B. Pfitzmann. Collision-free accumulators and fail-stop signature schemes without trees. In W. Fumy, editor, *Advances in Cryptology – EUROCRYPT 1997*, volume 1233 of *Lecture Notes in Computer Science*, pages 480–494. International Association for Cryptologic Research, Springer-Verlag, Berlin Germany, 1997.

[6] M. Bellare, D. Micciancio, and B. Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In E. Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, Warsaw, Poland, 2003. Springer.

[7] M. Bellare, H. Shi, and C. Zhang. Foundations of group signatures: The case of dynamic groups. Cryptology ePrint Archive, Report 2004/077, 2004. `http://eprint.iacr.org/`.

[8] D. Boneh and X. Boyen. Short signatures without random oracles. In C. Cachin and J. Camenisch, editors, *Advances in Cryptology – EUROCRYPT ' 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 56–73, Interlaken, Switzerland, 2004. Springer.

[9] D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In M. Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 41–55. Springer, 2004.

[10] J. Camenisch. Efficient and generalized group signatures. In W. Fumy, editor, *Advances in Cryptology - EUROCRYPT '97, International Conference on the Theory and Application of Cryptographic Techniques*, Lecture Notes in Computer Science, pages 465–479. International Association for Cryptologic Research, Springer, 1997.

[11] J. Camenisch and J. Groth. Group signatures: Better efficiency and new theoretical aspects. In *Security in Communication Networks - SCN 2004*. Springer, 2003.

[12] J. Camenisch and A. Lysyanskaya. An identity escrow scheme with appointed verifiers. In J. Kilian, editor, *Advances in Cryptology – CRYPTO ' 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 388–407. International Association for Cryptologic Research, Springer-Verlag, Berlin Germany, 2001.

[13] J. Camenisch and A. Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In M. Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*. Springer, 2004.

[14] J. Camenisch and M. Michels. A group signature scheme with improved efficiency. In K. Ohta and D. Pei, editors, *ASIACRYPT: Advances in Cryptology – ASIACRYPT: International Conference on the Theory and Application of Cryptology*, volume 1514 of *Lecture Notes in Computer Science*, pages 160–174. International Association for Cryptologic Research, Springer-Verlag, 1998.

[15] J. Camenisch and M. Michels. Separability and efficiency for generic group signature schemes (extended abstract). In M. j. Wiener, editor, *19th International Advances in Cryptology Conference – CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*, pages 413–430. Springer, 1999.

[16] J. Camenisch and V. Shoup. Practical verifiable encryption and decryption of discrete logarithms. In *CRYPTO 2003*. Springer-Verlag, 2003.

[17] J. Camenisch and M. Stadler. Efficient group signature schemes for large groups. In B. S. K. Jr., editor, *Advances in Cryptology – CRYPTO ' 1997*, volume 1294 of *Lecture Notes in Computer Science*, pages 410–424. International Association for Cryptologic Research, Springer, 1997.

[18] D. Chaum and E. van Heyst. Group signatures. In D. W. Davies, editor, *Advances in Cryptology, EUROCRYPT 1991 (Lecture Notes in Computer Science 547)*, pages 257–265. Springer-Verlag, April 1991. Brighton, U.K.

[19] L. Chen and T. P. Pedersen. New group signature schemes (extended abstract). In A. D. Santis, editor, *Advances in Cryptology—EUROCRYPT 94*, volume 950 of *Lecture Notes in Computer Science*, pages 171–181. Springer-Verlag, 1995, 9–12 May 1994.

[20] Y. Dodis, A. Kiayias, A. Nicolosi, and V. Shoup. Anonymous identification in ad hoc groups. In C. Cachin and J. Camenisch, editors, *Advances in Cryptology – EUROCRYPT ' 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 609–626, Interlaken, Switzerland, 2004. Springer.

[21] P.-A. Fouque and J. Stern. One round threshold discrete-log key generation without private channels. In K. Kim, editor, *Public Key Cryptography — 4th International Workshop on Practice and Theory in Public Key Cryptosystems*, volume 1992 of *Lecture Notes in Computer Science*, pages 300–316, Cheju Island, Korea, 2001. Springer.

[22] E. Fujisaki and E. Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In B. S. Kaliski, Jr., editor, *Advances in Cryptology – CRYPTO ' 97*, volume 1294 of *Lecture Notes in Computer Science*, pages 16–30. International Association for Cryptologic Research, Springer-Verlag, Berlin Germany, 1997.

[23] J. Garay, P. D. MacKenzie, and K. Yang. Strengthening zero-knowledge protocols using signatures. In E. Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 177–194, Warsaw, Poland, 2003. Springer.

[24] R. Gennaro and Y. Lindell. A framework for password-based authenticated key exchange. In E. Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, Warsaw, Poland, 2003. Springer.

[25] A. Kiayias, Y. Tsiounis, and M. Yung. Traceable signatures. In C. Cachin and J. Camenisch, editors, *Advances in Cryptology – EUROCRYPT ' 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 571–589, Interlaken, Switzerland, 2004. Springer.

[26] A. Kiayias and M. Yung. Extracting group signatures from traitor tracing schemes. In E. Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 630–648, Warsaw, Poland, 2003. Springer.

[27] A. Kiayias and M. Yung. Group signatures: Provable security, efficient constructions and anonymity from trapdoor-holders. Cryptology ePrint Archive, Report 2004/076, 2004. `http://eprint.iacr.org/`.

[28] A. Kiayias and M. Yung. Group signatures with efficient concurrent join. In R. Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 198–214, Aarhus, Denmark, 2005. Springer.

[29] J. Kilian and E. Petrank. Identity escrow. In H. Krawczyk, editor, *Advances in Cryptology – CRYPTO 1998*, volume 1462 of *Lecture Notes in Computer Science*, pages 169–185. International Association for Cryptologic Research, Springer, 1998.

[30] A. K. Lenstra. Generating rsa moduli with a predetermined portion. In K. Ohta and D. Pei, editors, *ASIACRYPT*, volume 1514 of *Lecture Notes in Computer Science*, pages 1–10. Springer, 1998.

[31] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in cryptology—EUROCRYPT 1999*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–238, 1999.

[32] D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, Mar. 2000.

[33] G. Tsudik and S. Xu. Accumulating composites and improved group signing. In *ASIACRYPT: Advances in Cryptology – ASIACRYPT: International Conference on the Theory and Application of Cryptology*, Lecture Notes in Computer Science, pages 269–286. International Association for Cryptologic Research, Springer, 2003.

Appendix: Auxiliary Lemmas

Lemma 5.7 *Let $n = pq$ with $p = 2p' + 1$ and $q = 2q' + 1$ with p, q, p', q' all prime numbers. Suppose we know $y \in \mathbb{Z}_n^*$, $z \in QR(n)$ and $t, m \in \mathbb{Z}$ such that $y^t \equiv_n z^m$ with $\gcd(t, m) < t$ and $|t| > 1$. Then we can find $e > 1$ and $u \in \mathbb{Z}_n^*$ such that $z \equiv_n u^e$, or we can factor n.*

Proof. (of lemma 5.7) First assume w.l.o.g. that t is a positive integer (if not, set $y \leftarrow y^{-1} \bmod n$).

We consider three cases according to $\delta =_{\mathsf{df}} \gcd(t, m)$.

Case (i). $\delta = 1$. In this case we can compute $\alpha, \beta \in \mathbb{Z}$ such that $\alpha t + \beta m = 1$. From this, in turn, we obtain:

$$z = z^{\alpha t + \beta m} = (z^\alpha)^t (z^m)^\beta = (z^\alpha y^\beta)^t$$

and thus, we return as the solution to the challenge, the pair $\langle u, e\rangle = \langle z^\alpha y^\beta, t\rangle$.

Case (ii). suppose that $\delta > 1$ and $\gcd(\delta, p'q') = 1$. It follows that $\delta \leq \min\{|t|, |m|\}$ and if $t' = \frac{t}{\delta}$ and $m' = \frac{m}{\delta}$, it holds that $(y^{t'})^\delta \equiv_n (z^{m'})^\delta$.

Now observe that if $2 \not| \delta$ we will have immediately that $y^{t'} \equiv_n z^{m'}$ and because $\gcd(t', m') = 1$ and $t' > 1$ (this is the case by the requirement of the statement of the theorem $\gcd(t, m) < t$) we are reduced to case (i).

Suppose instead that $\delta = 2^\ell v$ with $\ell > 1$. We have that $(y^{t'})^{2^\ell v} \equiv_n (z^{m'})^{2^\ell v}$ by assumption we have that $\gcd(v, p'q') = 1$ and we obtain from this that $(y^{t'})^2 \equiv_n (z^{m'})^2$. From this we have that there exist $b_1, b_2 \in \{0, 1\}$ such that $z^{m'} = (-1)^{b_1} \chi^{b_2} y^{t'}$ where χ is an element of order 2 inside $\mathbb{Z}_n^*$

so that $\chi \not\equiv_n -1$. Now if t' is odd, we have that $(-1)^{b_1}\chi^{b_2} \equiv_n ((-1)^{b_1}\chi^{b_2})^{t'}$ and as a result we have $z^{m'} = ((-1)^{b_1}\chi^{b_2}y)^{t'}$ and we reduce to case (i) as well. Now suppose that t' is even (something that forces m' to be odd). But then, $(y^{t'})^2 \equiv_n (z^{m'})^2$ would imply immediately that $y^{t'} \equiv_n z^{m'}$ since both $y^{t'}, z^{m'} \in QR(n)$ and we are reduced again to case i.
Case (iii). Suppose that $\gcd(\delta, p'q') > 1$. It follows that δ is a multiple of p' (w.l.o.g.). Then we can factor n as follows: choose a random integer w less than n; if $\gcd(w, n) > 1$ then we are done; otherwise, $w \in \mathbb{Z}_n^*$ and will happen that w is a square modulo $p = 2p' + 1$ with high probability, (since approximately half of the positive integers less than n are squares modulo p). It follows that $w^{p'} = (w^{\frac{1}{2}})^{2p'} = (w^{\frac{1}{2}})^{p-1} \equiv_p 1$. Now compute the integer $U = w^{\delta} = w^{\pi p'} (\mathrm{mod} n)$, where $\delta = \pi p'$ for some $\pi \in \mathbb{Z}$. Since $n \mid U - w^{\pi p'}$ it follows that $p \mid U - w^{\pi p'}$ and as a result, $U \equiv_p w^{\pi p'} \equiv_p (w^{p'})^{\pi} \equiv_p 1$. It follows that there exists an $r \in \mathbb{Z}$ such that $U - 1 = rp$. Observe that it has to be that $r < q$ since $U < n$. From this we obtain that $\gcd(U - 1, n) = p$. ■

Lemma 5.8 *Let A, B be two integers with $A > B$ and $A = \pi B + v$ with $0 \leq v < B$ and let X be a random variable with $X \leftarrow_R [A]$. If $Y = X \bmod B$ it holds that Y's statistical distance from the uniform in $\mathbb{Z}_B$ is at most v/A. If $Y' = \lfloor X/B \rfloor$, then the statistical distance of the uniform over $\{0, \ldots, \pi\}$ and Y' is at most $1/(\pi + 1)$.*

Lemma 5.9 *Let $B_1, \ldots, B_m \leftarrow_R QR(n)$ and let $\mathcal{A}$ be a* PPT *algorithm that on input $B_1, \ldots, B_m$ with probability α, it outputs integers $e_1, \ldots, e_m, t$ and $y \in \mathbb{Z}_n^*$ such that $|t| > 1$ and $\prod_{i=1}^m B_i^{e_i} = y^t$ such that $\exists i : t \not| e_i$. Then we can use $\mathcal{A}$ to solve the Strong-RSA problem with probability at least $\alpha/3$.*

Proof. (of lemma 5.9) First observe we may assume without loss of generality that t is positive since we can always set $y = y^{-1}$ as the output of $\mathcal{A}$.

The sample space over which the probability α is taken is identified to the coin tosses of $\mathcal{A}$ and the random choices of $B_1, \ldots, B_m$ from $QR(n)$.

Consider now the following experiment denoted by $\mathcal{E}$. Let g be a fixed generator of $QR(n)$. Select $b_i \leftarrow_R [n^2]$, and simulate $\mathcal{A}$ on input $g^{b_1}, \ldots, g^{b_m}$. From lemma 5.8 it follows that $b_i \bmod p'q'$ is statistically indistinguishable from the uniform $\mathbb{Z}_{p'q'}$ and thus the elements $B_i = g^{b_i}$ for $i = 1, \ldots, m$ are uniformly selected from $QR(n)$. It follows that $y^t = g^{e_1 b_1 + \ldots + e_m b_m}$. The output of the experiment $\mathcal{E}$ is $e_1, \ldots, e_m, y, t$. Let $\delta = \gcd(e_1 b_1 + \ldots + e_m b_m, t)$. The sample space for the event $\mathcal{E}$ corresponds to the choices for $b_1, \ldots, b_m$ as well as the coin tosses for the simulation of $\mathcal{A}$.

We can split the sample space of $\mathcal{E}$ to the following events (i) E_{fail}: the output of $\mathcal{A}$ fails to meet the specifications (either, $t = 1$, or $\sum_{i=1}^m e_i b_i \not\equiv_{p'q'} t$). (ii) E_{div} the output of $\mathcal{A}$ meets the specifications except that it holds that for all $i = 1, \ldots, n$, $t \mid e_i$. (iii) $E_{\delta<\mathsf{t}}$ all the specifications are met and $\delta < t$. (iv) $E_{\delta=\mathsf{t}}$ all the specifications are met and $\delta = t$. Based on the assumption of the lemma we have that $\mathbf{Prob}[E_{\delta<t}] + \mathbf{Prob}[E_{\delta=t}] = \alpha$ (for simplicity we omit the negligible statistical distance that exists between the executions of $\mathcal{A}$ and the experiment $\mathcal{E}$).

Observe that if the event $E_{\delta<t}$ happens the result of the theorem will follow directly from lemma 5.7 (by plugging the Strong-RSA challenge in the place of the generator g above and setting $m = e_1 b_1 + \ldots + e_m b_m$).

Next, let us consider the event $E_{\delta=t}$. The event $E_{\delta=t}$ suggests $t \mid e_1 b_1 + \ldots + e_m b_m$. Observe that we can view the event $E_{\delta=t}$ as containing tuples of the form $(\pi_1, \ldots, \pi_m, \rho)$ where $\pi_j = \lfloor \frac{b_j}{p'q'} \rfloor$ and ρ is a sequence of coin tosses that fixes the randomness of $\mathcal{A}$ as well as the choice of $b_j (\mathrm{mod} p'q')$ for $j = 1, \ldots, m$. Moreover, note that the output of $\mathcal{A}$ depends only on ρ and is independent of the choice of $\pi_1, \ldots, \pi_m$. Consider the subset Ω of $E_{\delta=t}$ for which it holds $(\pi_1, \ldots, \pi_m, \rho) \in \Omega$ iff there exists j such that no tuple of the form $(\pi_1, \ldots, \pi_{j-1}, \pi_j \pm 1, \pi_{j+1}, \ldots, \pi_m, \rho)$ belongs in $E_{\delta=t}$.

Next we show that if $\mathbf{Prob}[\Omega] \geq \epsilon$ then $\mathbf{Prob}[E_{t=\delta}] \leq \alpha - \epsilon$. Observe that the fact $(\pi_1, \ldots, \pi_m, \rho) \in \Omega$ excludes at least one tuple of the form $(\pi_1, \ldots, \pi_{j-1}, \pi_j \pm 1, \pi_{j+1}, \ldots \pi_m, \rho)$ to belong in $E_{\delta=t}$. Nevertheless this tuple has the same ρ component to a tuple that belongs to $E_{\delta=t}$ and thus it cannot belong to either E_{fail} or E_{div}. It follows that the tuple $(\pi_1, \ldots, \pi_{j-1}, \pi_j \pm 1, \pi_{j+1}, \ldots \pi_m, \rho)$ belongs to $E_{\delta<t}$ and thus if $\mathbf{Prob}[\Omega] \geq \epsilon$ it holds that $\mathbf{Prob}[E_{\delta<t}] \geq \epsilon$ and as a result $\mathbf{Prob}[E_{\delta=t}] \leq \alpha - \epsilon$.

Now suppose that the event $E_{\delta=t} - \Omega$ happens, we will show how to factor n. In this case we will obtain $e_1, \ldots, e_m$ and t such that $t \mid e_1 b_1 + \ldots + e_m b_m$ and $\exists i : t \not\mid e_i$. Suppose without loss of generality that $i = 1$. We know that the outcome $e_1, \ldots, e_m, t$ of the execution of $\mathcal{A}$ corresponds to some tuple $(\pi_1, \ldots, \pi_m, \rho)$ of $E_{\delta=t} - \Omega$ and we obtain that $t \mid e_1(\pi_1 p'q' + b_1 \bmod p'q') + e_2 b_2 + \ldots e_m b_m$ with $t \not\mid e_1$. Due to the fact that $(\pi_1, \ldots, \pi_m, \rho) \in E_{\delta=t} - \Omega$ it follows that some tuple of the form $(\pi_1 \pm 1, \pi_2, \ldots, \pi_m, \rho) \in E_{\delta=t}$. Because the behavior of $\mathcal{A}$ only depends on ρ it follows that for the same $t, e_1, \ldots, e_m$ it will hold $t \mid e_1((\pi_1 \pm 1)p'q' + b_1 \bmod p'q') + e_2 b_2 + \ldots e_m b_m$. By combining the above two divisibility relationships we obtain that $t \mid e_1 p'q'$, and since $t \not\mid e_1$ we have that $t \mid p'q'$. This implies that we can factor n as argued in the proof of lemma 5.7.

Suppose that $\mathbf{Prob}[E_{\delta<t}] \geq \alpha/3$. It follows that we can solve the Strong-RSA with probability $\alpha/3$. On the other hand if $\mathbf{Prob}[E_{\delta<t}] < \alpha/3$ we have that $\mathbf{Prob}[E_{\delta=t}] \geq 2\alpha/3$. Moreover, it follows that $\mathbf{Prob}[\Omega] < \alpha/3$ and thus $\mathbf{Prob}[E_{\delta=t} - \Omega] \geq \alpha/3$. Since we can solve the Strong-RSA problem when either of the events $E_{\delta<t}$ or $E_{\delta=t} - \Omega$ happen, it follows that we can solve the strong-RSA problem with probability at least $\alpha/3$. ■

Lemma 5.10 *Consider a fixed $x \in [L, R]$ with $m = R - L$ and the random variables $t \in_R \pm[0, 2^{k+l}m]$, $c \in_R \{0,1\}^k$. The statistical distance of the random variable $\hat{s} = t - c(x - L)$ from the random variable $s \in_R \pm[0, 2^{k+l}m]$ is less than 2^{-l-1}.*

Proof. We will denote by $\mathcal{D}_{\mathsf{a}}$ the distribution of the random variable s and by $\mathcal{D}_{\mathsf{b}}$ the distribution of $\hat{s} = t - c(x - L)$. Assume that the support of the two random variables is $\mathbb{Z}$.

- Regarding $\mathcal{D}_{\mathsf{a}}$ observe that a certain s_0 in $\pm[0, 2^{k+l}m]$ has probability of being selected equal to $\frac{1}{2^{k+l+1}m+1}$ (uniform probability distribution). Any $s_0 \notin \pm[0, 2^{k+l}m]$ has probability 0.
- Regarding $\mathcal{D}_{\mathsf{b}}$ observe that a certain s_0 has the following probabilities of being selected:
  1. For $s_0 \in [-2^{k+l}m, 2^{k+l}m - (2^k - 1)m]$ for each of the 2^k different $c_0 \in \{0,1\}^k$ we can find a t_0 such that $s_0 = t_0 - c_0 x$, as a result the probability of obtaining the given s_0 according to $\mathcal{D}_{\mathsf{b}}$ is $\frac{2^k}{2^k(2^{k+l+1}m+1)} = \frac{1}{2^{k+l+1}m+1}$.
  2. For $s_0 \in [-2^{k+l}m - (2^k - 1)m, -2^{k+l}m - 1]$ or $s_0 \in [2^{k+l}m - (2^k - 1)m + 1, 2^{k+l}m]$ the probability of obtaining s_0 according to $\mathcal{D}_{\mathsf{b}}$ is less than $\frac{1}{2^{k+l+1}m+1}$.
  3. For the remaining $s_0 < -2^{k+l}m - (2^k - 1)m$ and $s_0 > 2^{k+l}m$ the probability of selecting them according to $\mathcal{D}_{\mathsf{b}}$ is equal to 0.

It is clear from the above that the absolute difference between the probability of a certain s_0 according to $\mathcal{D}_{\mathsf{b}}$ and $\mathcal{D}_{\mathsf{a}}$ is 0 for the integer ranges of cases 1 and 3 above. The distributions $\mathcal{D}_{\mathsf{a}}$ and $\mathcal{D}_{\mathsf{b}}$ will accumulate some statistical distance though due to their different behavior for s_0 that belong to the integer range specified in item 2. In this case, for a specific s_0, distribution $\mathcal{D}_{\mathsf{a}}$ assigns probability either 0 or $\frac{1}{2^{k+l+1}m+1}$ whereas distribution $\mathcal{D}_{\mathsf{b}}$ assigns probability that belongs in the real interval $[0, \frac{1}{2^{k+l+1}m+1})$. Clearly, in the worst case the absolute difference will be $\frac{1}{2^{k+l+1}m+1}$ The number of elements s_0 of case 2, are $2 \cdot (2^k - 1)m$ thus it follows that the statistical distance of the distributions $\mathcal{D}_{\mathsf{a}}$ and $\mathcal{D}_{\mathsf{b}}$ cannot be greater than $(2^k - 1)m/(2^{k+l+1}m + 1) < 2^{-l-1}$. This completes the proof. ■