Group Signatures: Provable Security, Efficient Constructions and Anonymity from Trapdoor-Holders

Aggelos Kiayias

Computer Science & Engineering
University of Connecticut
Storrs, CT, USA
aggelos@cse.uconn.edu

Moti Yung

Computer Science
Columbia University
New York, NY, USA
moti@cs.columbia.edu

Abstract

To date, a group signature construction which is both efficient and proven secure in a formal model has not been suggested. In this work we give the first such construction. To this end we present a new formal model for group signatures capturing the state-of-the-art requirements in the area. We then construct an efficient scheme and prove its security. Our methods require novel cryptographic constructs and new number-theoretic machinery for arguing security over the group of quadratic residues modulo a composite when its factorization is *known*. Along the way, we unveil properties which go beyond the state-of-the-art-scheme [Ateniese et al.2000] and reveal subtle points regarding the assumptions and requirements that underly efficient group signature schemes.

Contents

1 Introduction.

The notion of *group signature* is a useful anonymous non-repudiable credential primitive that was introduced by Chaum and Van Heyst [CvH91]. This primitive involves a group of users, each holding a membership certificate that allows a user to issue a publicly verifiable signature which hides the identity of the signer within the group. The public-verification procedure employs only the public-key of the group. Furthermore, in a 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 scalable group signature has been a research target for many years since its introduction, see e.g., [CP94, CS97, CM98, CM99, Cam97, KP98, AT99, ACJT00, CL01, KY03]. The current state of the art is the scalable scheme of Ateniese, Camenisch, Joye and Tsudik [ACJT00] that provides constant signature size and resistance to attacks by coalitions of users. This remarkable scheme was based on a novel use of the DDH assumption combined with the Strong-RSA assumption over groups of unknown order. Recently, Bellare, Micciancio and Warinschi [BMW03], noticing that [ACJT00] only prove a collection of individual intuitive security properties[1], advocated the need for a formal model for arguing the security of group signature. This important observation is in line with the development of solid security notions in modern cryptography. They also offered a model of a relaxed group signature primitive and a generic construction in that model. Generic constructions are inefficient and many times are simpler than efficient constructions (that are based on specific number theoretic problems). This is due to the fact that generic constructions can employ (as a black box) the available heavy and powerful machinery of general zero-knowledge protocols and general secure multi-party computations. Thus, generic constructions typically serve only as plausibility results for the existence of a cryptographic primitive [Gol97].

The above state of affairs [ACJT00, BMW03] indicates that there exists a gap in the long avenue of research efforts regarding the group signature primitive. This gap is typical in cryptography and is formed by a difference between prohibitively expensive constructions secure in a formal sense and efficient more ad-hoc constructions. In many cases, as indicated above, it is easier to come up with provably secure generic inefficient constructions or to design efficient ad-hoc constructions. It is often much harder to construct an efficient implementation that is proven secure within a formal model. To summarize the above, it is apparent that the following question remained open till today:

> Design an **efficient** group signature which is **provably secure** within a formal model.

One of our contributions is solving the above open question by, both, proposing a new formal model for group signature which follows the [GMR84] paradigm, as well as providing an efficient provably secure construction. Our construction is motivated by the [ACJT00]-scheme.

This contribution reveals numerous subtleties regarding what assumptions are actually necessary for achieving the security properties. For example, the security property called coalition resistance in [ACJT00] was dealt with by assuming the Strong-RSA assumption. In contrast, we find that actually this property (derived formally in our model) can be completely disassociated from factoring related assumptions. Our investigation also reveals delicate issues regarding the proper formal modeling of the group signature primitive with regards to the work of [BMW03]. For example, the need of formalizing security against an attack by an external entity that is not part of the group. Lack of such treatment, while proper for the relaxed notion of group signature of [BMW03], is insufficient for proving the security of efficient state-of-the-art schemes that follow the line of work of [ACJT00].

1.1 Our Contributions

Below, we outline what this work achieves in more details.

[1] These properties include unforgeability, anonymity, coalition-resistance (the fact that coalitions of group members cannot produce another membership certificate), exculpability (the fact that an adversarial group manager cannot produce a signature that opens to a non-adversarial controlled user).

1. MODELING. We present a new formal model that defines properties of schemes which share the nature of the state-of-the-art efficient group signatures schemes [ACJT00]. The model captures all their security requirements and functions (e.g., interactively joining group members). In particular, our model has three types of attacks that involve the GM and the users. All the attacks are modeled as games between the adversaries and a party called the interface. The interface represents the system in a real environment and simulates the behavior of the system (a probabilistic polynomial time simulator) in the security proof. The attacker gets oracle query capabilities to probe the state of the system and is also challenged with an attack task; this follows the approach of [GMR84] for modeling security of digital signatures. We provide a natural classification of attacks that capture all possible adversarial behaviors against a group signature; we call these attacks the insider-attack, the outsider-attack and the deanonymizer-attack.

2. ADVERSARIAL OPENING IN EFFICIENT SCHEMES. Our formal model extends the security requirements given by the list of security properties of [ACJT00] by allowing the adversary to request that the system opens signatures of its choice. In [ACJT00] opening of signatures was implicitly assumed to be an internal operation of the GM. We note that such stronger adversarial capability was considered in the formal model of [BMW03]. For achieving an efficient scheme with adversarial opening we needed to develop novel cryptographic constructs.

3. STRONGER ANONYMITY PROPERTY. In the scheme of [ACJT00] anonymity is argued against an adversary that is not allowed to corrupt the GM. This is a natural choice since in their scheme the GM holds the trapdoor which provides the opening capability, namely an ElGamal key. The GM also holds the trapdoor that is required to enroll users to the group, namely the factorization of an RSA-modulus. However, pragmatically, there is no need to combine the GM function that manages group members (which in real life can be run by e.g., a commercial company) with the opening authority function (which in real life can be run by a government entity). To manage members the GM still needs to know the factorization. The opening authority, on the other hand, must know the ElGamal key. This split of functions is not a relaxation of group signatures but rather a constraining of the primitive. In fact, now we should allow the deanonymizer adversary to *corrupt the GM as well*. For proving security in the above stronger adversarial scenario, we had to develop a novel machinery for arguing security.

4. NUMBER-THEORETIC MACHINERY AND CRYPTOGRAPHIC PRIMITIVES. The two contributions above required building cryptographic primitives over the set of quadratic residues modulo $n = pq$ that remain secure when the factorization (into two strong primes) p, q is known to the adversary.

To this end, we investigate the Decisional Diffie Hellman Assumption over the quadratic residues modulo n and we prove that it remains hard even if the adversary knows the factorization. In fact, we prove that any adversary that knows the factorization p, q and solves the DDH problem over quadratic residues modulo a composite $n = pq$, can be turned into a DDH-distinguisher for quadratic-residues modulo a prime number.

This result is of independent interest since it suggests that the DDH over $QR(n)$ does not depend to the factorization problem at all.

Also, the present work requires a CCA2 encryption mechanism that operates over the quadratic residues modulo n so that (i) encryption should not use the factorization of n, (i.e., the factorization need not be a part of the public-key), but on the other hand (ii) the factorization is *known* to the attacker. In this work we derive such a primitive in the form of an ElGamal variant following the general approach of twin encryption [NY90, DDN91, FP01] which is CCA2 secure under the DDH assumption in the Random Oracle model (note that our efficient group signature requires the random oracle anyway since it is derived from the Fiat-Shamir transform [FS86, AABN02]).

5. EFFICIENT CONSTRUCTION. We provide an efficient construction of a group signature that is proven secure in our model. While, we would like to note that our scheme is strongly influenced by [ACJT00] (and originally we tried to rely on it as much as possible), our scheme, nevertheless, possesses certain subtle and important differences. These differences enable the proof of security of our scheme whereas the scheme in [ACJT00] cannot be proven secure in our model: There are many reasons for this, e.g., the [ACJT00]-scheme lacks an appropriate CCA2 secure identity embedding mechanism. Moreover, our efficient construction can support (if

so desired), the separation of group management and opening capability as mentioned above. Finally we note that a syntactically degenerated version of our construction (that retains its efficiency) can be proven secure in the model of [BMW03] (and is, in fact, a relaxed group signature scheme of the type they have suggested).

6. UNDERPINNING PRINCIPLES. Some seemingly rather surprising results come from our investigation when compared with the related design and arguments in [ACJT00] (at least we were surprised by the findings). These new findings demonstrate that some ingredients that appeared to be crucial in the [ACJT00] design are in fact redundant or unnecessary. Let us review them:
(i) one of the major contributions of [ACJT00] was their argument that showed coalition resistance based on the Strong-RSA assumption. In our setting, coalition-resistance is subsumed by our "insider-attack" (the system is conspiring against innocent users). We prove that *any* kind of factoring related assumption is not needed for security against insider attacks and we can base them solely on the discrete-logarithm assumption.
(ii) The Join protocol is a rather complex and sophisticated mechanism in [ACJT00] which requires various stages and proofs of knowledge with a total of eight rounds of interaction (which can be compressed yet remain complex). This sophistication seemed to be necessary and crucial for the proof of security. We discover that this is not so and in fact the Join protocol in our construction is only 2 simple rounds of communication without requiring any proofs of knowledge.
(iii) Anonymity was argued in [ACJT00] to be based on the decisional Diffie-Hellman Assumption over Quadratic Residues modulo a composite and given that the GM was assumed to be uncorrupted, the key-issuing trapdoor (the factorization of the modulus) was not meant to be known to the adversary. As argued above, we prove that anonymity *still holds* when the adversary is given the factorization (even when there is no separation of group management and opening authority). Thus, we totally disassociate anonymity from the factoring problem.

1.2 Organization

In section 2 we present some background useful tools and the intractability assumptions. In section 3 we investigate the behavior of the DDH assumption over the quadratic residues modulo a composite when the factorization is known to the distinguisher. In section 4 we discuss the kind of CCA2 security that will be required in our setting (over $QR(n)$ but with known factorization) and we present an efficient and provably secure construction based on the ElGamal twin-encryption paradigm. In section 5 we present our security model and definitions and in section 6 we give our construction and its proofs of correctness and security. In section 7 we present group signatures with separated authorities (i.e., the GM and the opening authority – OA). In this setting, we demonstrate how our construction can still be proven secure when assuming a stronger deanonymizer adversary that is allowed to corrupt the GM in addition to users. Finally, in section 8 we discuss the interactive version of our scheme as an identity escrow scheme.

2 Preliminaries

NOTATIONS. We will write PPT for probabilistic polynomial-time. If $\mathcal{D}_1$ and $\mathcal{D}_2$ are two probability distributions defined over the same support set that is parameterized by k we will write $\mathrm{dist}_{\mathcal{A}}(\mathcal{D}_1, \mathcal{D}_2)$ to denote the distance $|\mathbf{Prob}_{x \leftarrow \mathcal{D}_1}[\mathcal{A}(x) = 1] - \mathbf{Prob}_{x \leftarrow \mathcal{D}_2}[\mathcal{A}(x) = 1]|$. Note that typically $\mathrm{dist}_{\mathcal{A}}$ will be expressed as a function of k. If n is an integer, we will denote by $[n]$ the set $\{1, \ldots, n\}$. If we write $a \equiv_n b$ for two integers a, b we mean that n divides $a - b$ or equivalently that a, b are the same element within $\mathbf{Z}_n$. A function $f : \mathbb{N} \rightarrow \mathbb{R}$ will be called negligible if for all $c > 0$ there exists a k_c such that for all $k \geq k_c$, $f(k) < k^{-c}$. In this case we will write $f(k) = \mathsf{negl}(k)$. If $\ell, \mu \in \mathbb{N}$ we will write $S(2^\ell, 2^\mu)$ for the set $\{2^\ell - 2^\mu + 1, \ldots, 2^\ell + 2^\mu - 1\}$. PPT will stand for "probabilistic polynomial time."

Throughout the paper (unless noted otherwise) we will work over the group of quadratic residues modulo n, denoted by $QR(n)$, where $n = pq$ and $p = 2p' + 1$ and $q = 2q' + 1$ and p, q, p', q' prime numbers. All

operations are to be interpreted as modulo n (unless noted otherwise). We will employ various related security parameters (as introduced in the sequel); with respect to an entity we will use ν as the security parameter to denote a quantity proportional to the logarithm of the size of the entity. Next we define the Cryptographic Intractability Assumptions that will be relevant in proving the security properties of our constructions.

The first assumption is the Strong-RSA assumption. It is similar in nature to the assumption of the difficulty of finding e-th roots of arbitrary elements in $\mathbf{Z}_n^*$ with the difference that the exponent e is not fixed (i.e., it is not part of the instance).

Definition 1 Strong-RSA. *Given a composite n (as described above), and $z \in QR(n)$, it is infeasible to find $u \in \mathbf{Z}_n^*$ and $e > 1$ such that $u^e = z (\mathrm{mod} n)$, in time polynomial in ν.*

Note that the variant we employ above restricts the input z to be a quadratic residue. This variant of Strong-RSA has been discussed before [CS00], and by restricting the exponent solutions to be only odd numbers we have that (i) it cannot be easier than the standard unrestricted Strong-RSA problem, but also (ii) it enjoys a random-self reducibility property (see [CS00]).

The second assumption that we employ is the Decisional Diffie-Hellman Assumption (see e.g., [Bon98] for a survey). We state it below for a general group G and later on in definition 11 we will specialize this definition to two specific groups.

Definition 2 Decisional Diffie-Hellman *Given a description of a cyclic group G that includes a a generator g, a DDH distinguisher $\mathcal{A}$ is a polynomial in ν time* PPT *that distinguishes the family of triples of the form $\langle g^x, g^y, g^z \rangle$ from the family of triples of the form $\langle g^x, g^y, g^{xy} \rangle$, where $x, y, z \in_R \#G$. The* DDH *assumption suggests that this advantage is a negligible function in ν.*

Finally, we will employ the discrete-logarithm assumption over the quadratic residues modulo n with known factorization (note that the discrete-logarithm problem is assumed to be hard even when the factorization is known, assuming of course that the factors of n are large primes p, q and where $p-1$ and $q-1$ are non-smooth).

Definition 3 Discrete-Logarithm. *Given two values a, b that belong to the set of quadratic residues modulo n with known factorization, so that $x \in [p'q'] : a^x = b$, find in time polynomial in ν the integer x so that $a^x = b$.*

Conventions. (i) our proofs of knowledge will only be proven to work properly in the honest-verifier setting. On the one hand, the honest-verifier setting is sufficient for producing signatures. On the other hand, even in the general interactive setting the honest-verifier scenario can be enforced by assuming the existence, e.g., of a beacon, or some other mechanism that can produce trusted randomness; alternatively the participants may execute a coin flipping algorithm and use methods that transform the honest verifier proofs to a regular proofs. (ii) the public parameters employed in our various protocol designs (e.g., the composite modulus n) will be be assumed to be selected honestly.

2.1 Auxiliary Lemmas

We prove below two auxiliary lemmas that will be useful later on. The first lemma is an extension of a well-known lemma. This known lemma is case (i) in the proof, and is attributed to [Sha83]. Several variants and extensions of this lemma have been used before (e.g., [CS00, CL02]).

Lemma 4 *Let $n = pq$ with $p = 2p' + 1$ and $q = 2q' + 1$ with p, q, p', q' all prime numbers. Suppose we know $y \in \mathbf{Z}_n^*$, $z \in QR(n)$ and $t, m \in \mathbf{Z}$ such that $y^t \equiv_n z^m$ with $\gcd(t, m) < t$ and $t > 1$ is an odd number. Then we can find $e > 1$ and $u \in \mathbf{Z}_n^*$ such that $z \equiv_n u^e$, or we can factor n.*

Proof. Case (i): $\gcd(t, m) = 1$ we can compute $\alpha, \beta \in \mathbf{Z}$ 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 $\gcd(t, m) = \delta > 1$. 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 since t is an odd number and $z^m \in QR(n)$ it must be the case that y also belongs to $QR(n)$. Under the assumption $\gcd(\delta, p'q') = 1$ we know that the exponentiation-map over $QR(n)$ defined as $f_{\delta}(a) = a^{\delta} (\mathrm{mod} n)$ is bijective (since $p'q'$ is the order of $QR(n)$), from which we obtain that $y^{t'} \equiv_n z^{m'}$ with $\gcd(t', m') = 1$; moreover $t' > 1$ since $\gcd(t, m) = \delta < t$. Thus, we reduced case (ii) to case (i).

Now 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, $w \in \mathbf{Z}_n^*$ 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 $\pi \in \mathbf{Z}$. 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 $r \in \mathbf{Z}$ such that $U - 1 = rp$. Observe that it has to be that $r < q$ since $U < n$. From this we obtain that $\gcd(U - 1, n) = p$. □

The second lemma below is a probabilistic indistinguishability result that will be useful in the proof of security of our construction.

Lemma 5 *Let n be a ν-bit composite $n = (2p' + 1)(2q' + 1)$ and $a, a_0 \in QR(n)$ with a a generator. Consider the following distributions over $QR(n)$:*

1. *$\mathcal{D}_1$ with $A \leftarrow a^x$ where $x \leftarrow_R [\lfloor n/4 \rfloor]$.*
2. *$\mathcal{D}_2$ with $A \leftarrow a^x$ where $x \leftarrow_R [n^2]$.*
3. *$\mathcal{D}_3$ with $A \leftarrow (a_0 a^x)^{1/e} (\mathrm{mod} n)$ where $x \leftarrow_R [\lfloor n/4 \rfloor]$ and e a prime with $e \leftarrow_R S(2^{\ell}, 2^{\mu}) - \{p', q'\}$ where $\ell, \mu \in \mathbb{N}$ with $S(2^{\ell}, 2^{\mu}) \subseteq [p'q']$.*
4. *$\mathcal{R}$ with $A \leftarrow_R QR(n)$.*

It holds that $\mathrm{dist}(\mathcal{D}_i, \mathcal{R}) = \mathsf{negl}(\nu)$ for all $i = 1, 2, 3$.

Proof. For the case $i = 1$, recall that the order of $QR(n)$ in $\mathbf{Z}_n^*$ is $p'q'$ and that $n = (2p' + 1)(2q' + 1) = 4p'q' + 2(p' + q') + 1$. It follows that $\lfloor n/4 \rfloor = p'q' + \frac{p'-1}{2} + \frac{q'-1}{2} + 1$. The dominant term in the statistical distance of the two distributions $\mathcal{D}_1$ and $\mathcal{R}$ is the fraction $\frac{\lfloor n/4 \rfloor - p'q'}{\lfloor n/4 \rfloor} = \frac{p'+q'}{\lfloor n/4 \rfloor}$ which is clearly negligible in the parameter ν.

Regarding the case $i = 2$, let us consider the general case for two integers $A > B$, of the distribution of the random variable $X \bmod B$ when $X \leftarrow_R [A]$. It follows that in this distribution $A \bmod B$ elements of $\mathbf{Z}_B$ are assigned probability $(\lfloor A/B \rfloor + 1)/A$ and $B - A \bmod B$ elements of $\mathbf{Z}_B$ are assigned probability $\lfloor A/B \rfloor / A$. The dominant term in the statistical distance between this distribution and the uniform over $\mathbf{Z}_B$ is $\frac{A \bmod B}{A}$. For the case $A = n^2$ and $B = p'q'$ we immediately have that the statistical distance is negligible.

Finally, regarding $i = 3$, we know from item 1, that a^x with $x \leftarrow_R [\lfloor n/4 \rfloor]$ is indistinguishable from $\mathcal{R}$. It follows that $a_0 a^x$ is indistinguishable from $\mathcal{R}$, and assuming that $\gcd(e, p'q') = 1$ (ensured by the statement of the theorem) it follows that the mapping in $QR(n)$ defined as $f_e(a) = a^e (\mathrm{mod} n)$ is bijective, thus the distribution $(a_0 a^x)^{1/e} (\mathrm{mod} n)$ is statistically indistinguishable from $QR(n)$. This will be the case for *any* fixed choice of e in the set $(S(2^{\ell}, 2^{\mu}) - \{p', q'\}) \cap \{p \mid p \text{ is a prime}\}$. Now if e is selected at random, the random variable $(a_0 a^x)^{1/e} (\mathrm{mod} n)$ will also be indistinguishable from $\mathcal{R}$. □

2.2 The Forking Lemma

Below we mention a general-purpose lemma that is instrumental in proving the security of signature schemes in the random oracle setting that has been formulated by Pointcheval and Stern [PS00] as the "forking-lemma":

Lemma 6 *(General Forking-Lemma). Consider be a probabilistic* PPT $\mathcal{P}$*, a* PPT *predicate Q and hash-function $\mathcal{H}$ with range $\{0,1\}^k$ thought of as a random oracle. The predicate Q satisfies the property $Q(x) = \top \Longrightarrow (x = \langle \rho_1, c, \rho_2 \rangle) \wedge (c = \mathcal{H}(\rho_1))$. $\mathcal{R}$ is a process that given $\langle t, c \rangle$ reprograms $\mathcal{H}$ so that $\mathcal{H}(t) = c$. $\mathcal{P}$ is allowed to ask queries on $\mathcal{H}$ and on $\mathcal{R}$. Moreover, it is assumed that $\mathcal{P}$ behaves in such a way so that queries $\langle t, c \rangle$ submitted by $\mathcal{P}$ to $\mathcal{R}$ adhere to the following conditions:*

- *The component c is uniformly distributed over $\{0,1\}^k$.*
- *The component t follows a probability distribution so that the probability of the occurrence of a specific t_0 is bounded by $2/2^k$.*

Assume now that $\mathcal{P}^{\mathcal{H},\mathcal{R}}(\mathsf{param})$ returns output x such that $Q(x) = \top$ with non-negligible probability $\epsilon \geq 10(s+1)(s+q)/2^k$, where q is the number of $\mathcal{H}$-queries performed by $\mathcal{P}$, and s is the number of $\mathcal{R}$ queries. Then, there exists a PPT *$\mathcal{P}'$ so that if $y \leftarrow \mathcal{P}'(\mathsf{param})$ it holds with probability $1/9$ (i) $y = \langle \rho_1, c, \rho_2, c', \rho_2' \rangle$ (ii) $Q(\langle \rho_1, c, \rho_2 \rangle) = \top$, (iii) $Q(\langle \rho_1, c', \rho_2 \rangle) = \top$, (iv) $c \neq c'$. The probabilities are taken over the choices for $\mathcal{H}$, the random coin tosses of $\mathcal{P}$ and the random choice of the public-parameters* param.

The above fundamental lemma (slightly differently formulated) was investigated and proven in [PS00]. The original lemma (as presented in [PS00]) allowed random oracle and signing queries. Nevertheless, it is apparent in the arguments presented in [PS00] that signing queries submitted to a simulator machine can be abstracted as random oracle "reprogramming" queries (i.e., the adversary submits signing queries to the simulator, who in turn reprograms the random oracle). Note that the two conditions stated in our formulation above are essential due to the way that the simulator treats signing queries in the proof of the original forking-lemma of [PS00] (as well as the properties of the underlying signing scheme) enforce these conditions as well.

2.3 Discrete-Log Relation Sets and Signatures

Discrete-log relation sets were introduced in [KTY04] as a basic tool to plan complex proofs of zero-knowledge over the set of quadratic residues modulo n and will be useful in our designs. This type of proofs was motivated and in fact unifies previous works on such proof systems, cf. [CM98, ACJT00].

Unlike [KTY04] here we use the tool of discrete-log relation sets in two alternate settings: when the factorization is unknown (as originally employed in [KTY04]), and also when the factorization is known. Interestingly, we need both cases and in fact we require a uniform protocol that would work smoothly under both scenarios. To the best of our knowledge this is the first such proof system that is required to be operational in both domains simultaneously. Let us first recall the definition of a discrete-log relation set:

Definition 7 *A* discrete-log relation set *R with z relations over r variables and m objects is a set of relations defined over the objects $A_1, \ldots, A_m \in QR(n)$ and the free variables $\alpha_1, \ldots, \alpha_r$ with the following specifications: (1) The i-th relation in the set R is specified by a tuple $\langle a_1^i, \ldots, a_m^i \rangle$ so that each a_j^i is selected to be one of the free variables $\{\alpha_1, \ldots, \alpha_r\}$ or an element of $\mathbf{Z}$. The $z \times m$ matrix $[a_j^i]_{i,j}$ will be denoted by D_R or simply D. The relation is to be interpreted as $\prod_{j=1}^m A_j^{a_j^i} = 1$. (2) Every free variable α_j is assumed to take values in a finite integer range $S(2^{\ell_j}, 2^{\mu_j})$ where $\ell_j, \mu_j \geq 0$.*

We will write $R(\alpha_1, \ldots, \alpha_r)$ to denote the conjunction of all relations $\prod_{j=1}^m A_j^{a_j^i} = 1$ that are included in R.

A discrete-log relation set R is said to be triangular, if for each relation i containing the free variables $\alpha_w, \alpha_{w_1}, \ldots, \alpha_{w_b}$ it holds that the free-variables $\alpha_{w_1}, \ldots, \alpha_{w_b}$ were contained in relations $1, \ldots, i-1$. In [KTY04] a 3-move honest verifier zero-knowledge proof (see e.g. [CDS94]) was designed that allows to a prover that knows witnesses $x_1, \ldots, x_r$ such that $R(x_1, \ldots, x_r) = 1$ to prove knowledge of these values.

Theorem 8 *For any discrete-log relation set R there exists a 3-move protocol (figure 1) that satisfies (i) completeness and (ii) honest-verifier zero-knowledge. (iii) assuming that the factorization of $n = pq$ is unknown, given two accepting conversations with the same first move, it is possible to extract a witness for a triangular D_R or any prover generating accepting conversations can be turned to a factorization algorithm;*

Alternatively, (iii′) assuming the factorization of $n = pq$ is known, given two accepting conversations with the same first move, it is possible to extract a witness for any discrete-log relation set assuming that $2^k < \min\{p', q'\}$ (where k is the length of the challenge).

Proof. Items (i), (ii), (iii) were proven in [KTY04].

We note that (i) and (ii) are unconditional. This is immediate for completeness, where regarding honest-verifier zero-knowledge, in [KTY04] this property is proven in the statistical sense. As a result, knowledge of the factorization of n does not alter the argumentation for (i) and (ii).

Regarding property (iii′) we remark that in the soundness argument (iii), the factorization is used only as a tool to ensure that if a prover generates accepting conversations, some integers formed by the prover's answers divide each other (and if not, a factorization can be found). This way of arguing about soundness was put forth in [FO98] to deal with the fact that the underlying group is of unknown order. In the known factorization setting the order of $QR(n)$ is known and thus one can show soundness more easily (e.g., [CDS94]); the only thing that needs to be ensured is that challenges $c \in \{0,1\}^*$ and differences of challenges (as integers) are invertible modulo $p'q'$, something guaranteed by the condition $2^k < \min\{p', q'\}$. □

A signature based on a proof of knowledge based on a discrete-log relation set can be obtained by applying the Fiat-Shamir transform [FS86]. We detail this transformation below. Let $\mathcal{G}$ be a hash function with range $\{0,1\}^k$ and D the matrix of some discrete-log relation set R over the base elements $A_1, \ldots, A_m \in QR(n)$. The proof of knowledge of figure 1 can be made into a signature as follows: given a message M, the verifier's challenge in the proof of knowledge will be computed as $c \leftarrow \mathcal{G}(M, A_1, \ldots, A_m, B_1, \ldots, B_z)$ using the hash $\mathcal{G}$. The signature on M will be denoted as $\mathtt{sgn}^D_{\mathcal{G}}(M)$ and computed as $\langle c, s_1, \ldots, s_r \rangle$ where $s_1, \ldots, s_r$ are computed as in the figure 1 and c is the hash $\mathcal{G}(M, A_1, \ldots, A_m, B_1, \ldots, B_z)$.

The verification algorithm $\mathtt{ver}^D_{\mathcal{G}}$ on a signature $\langle c, s_1, \ldots, s_r \rangle$ for a message M is implemented by the following check: $c \stackrel{?}{=} \mathcal{G}(M, A_1, \ldots, A_m, B_1, \ldots, B_z)$, where each B_i is computed by the verifier as $B_i = (\prod_{j:a^i_j \in \mathbf{Z}} A_j^{a^i_j} \prod_{j:\exists w, a^i_j = \alpha_w} A_j^{2^{\ell_w}})^{-c} \prod_{j:\exists w, a^i_j = \alpha_w} A_j^{s_w}$, for $i = 1, \ldots, z$.

The security of the Fiat-Shamir signature construction [FS86] was investigated by [PS00] as was noted above.

Note that the proof of knowledge of figure 1 also enforces interval constraints on the witnesses. In particular if proving knowledge of a witness $x \in S(2^\ell, 2^\mu)$ the proof ensures that the witness belongs to the range $S(2^\ell, 2^{\epsilon(\mu+k)+2})$. This constraint comes "for free" in the soundness proof. If tighter integer ranges are needed they can also be achieved at the cost of making the proof slightly longer by employing [Bou00]. The tightness achieved by the proof for discrete-log relation sets itself will be sufficient for our designs.

3 Decisional Diffie Hellman over $QR(n)$ with known Factorization

Our constructions will require the investigation of the number-theoretic machinery presented in this section.

Let n be a composite, $n = pq$ with $p = 2p' + 1$ and $q = 2q' + 1$ (p, q, p', q' primes). Recall that elements of $\mathbf{Z}^*_n$ are in a 1-1 correspondence with the set $\mathbf{Z}^*_p \times \mathbf{Z}^*_q$. Indeed, given $\langle b, c \rangle \in \mathbf{Z}^*_p \times \mathbf{Z}^*_q$, consider the system of equations

Proof of knowledge for a Discrete-Log Relation Set R

objects $A_1, \ldots, A_m$, r free-variables $\alpha_1, \ldots, \alpha_r$, parameters: $\epsilon > 1, k \in \mathbb{N}$,

Each variable α_j takes values in the range $S(2^{\ell_j}, 2^{\mu_j})$

$\mathcal{P}$ proves knowledge of the witnesses $x_j \in S(2^{\ell_j}, 2^{\epsilon(\mu_j+k)+2})$ s.t. $R(x_1, \ldots, x_r) = 1$

$\mathcal{P}$		$\mathcal{V}$
for $w \in \{1, \ldots, r\}$ select $t_w \in_R \pm\{0,1\}^{\epsilon(\mu_w+k)}$		
for $i \in \{1, \ldots, z\}$ set $B_i = \prod_{j:\exists w, a_j^i = \alpha_w} A_j^{t_w}$	$\xrightarrow{B_1, \ldots, B_z}$	$c \in_R \{0,1\}^k$
	$\xleftarrow{c}$	
for $w \in \{1, \ldots, r\}$ set $s_w = t_w - c \cdot (x_w - 2^{\ell_w})$	$\xrightarrow{s_1, \ldots, s_r}$	Verify:
		for $w \in \{1, \ldots, r\}$
		$s_w \in_? \pm\{0,1\}^{\epsilon(\mu_w+k)+1}$
		for $i \in \{1, \ldots, z\}$
		$\prod_{j:\exists w, a_j^i = \alpha_w} A_j^{s_w} \stackrel{?}{=} B_i (\prod_{j:a_j^i \in \mathbf{Z}} A_j^{a_j^i} \prod_{j:\exists w, a_j^i = \alpha_w} A_j^{2^{\ell_w}})^c$

Figure 1: *Proof of Knowledge for a Discrete-Log relation set R (from [KTY04]). Regarding the length of the proof we note that the proof requires the first communication flow from the prover to the verifier to be of size z $QR(n)$ elements (where z is the number of relations in R) and the second communication flow from the prover to the verifier to be of total bit-length $\sum_{w=1}^{r}(\epsilon(\mu_w + k) + 1)$.*

$x \equiv b (\mathrm{mod} p)$ and $x \equiv c (\mathrm{mod} q)$. Using Chinese remaindering we can construct a solution of the above system since $\gcd(p, q) = 1$ and the solution will be unique inside $\mathbf{Z}_n^*$. Alternatively for any $a \in \mathbf{Z}_n^*$ we can find the corresponding pair $\langle b, c \rangle$ in $\mathbf{Z}_p^* \times \mathbf{Z}_q^*$ by computing $b = a (\mathrm{mod} p)$ and $c = a (\mathrm{mod} q)$ (note that $\gcd(a, n) = 1$ implies that $b \not\equiv 0 (\mathrm{mod} p)$ and $c \not\equiv 0 (\mathrm{mod} q)$. The mapping ρ from $\mathbf{Z}_p^* \times \mathbf{Z}_q^*$ to $\mathbf{Z}_n^*$ is called the Chinese remaindering mapping. Observe that ρ preserves quadratic residuosity: indeed $\rho(QR(p) \times QR(q)) = QR(n)$.

The following two lemmas will be useful in the sequel as they show how the Chinese remaindering mapping behaves when given inputs expressed as powers inside the two groups $QR(p)$ and $QR(q)$. In essence it shows that there is a simple way to define the discrete-logarithm of the result of the mapping if we know the discrete-logarithms of the two inputs to the mapping.

Lemma 9 *Let g_1, g_2 be generators of the groups $QR(p)$ and $QR(q)$ respectively, where the groups are defined as above. Then, if $\beta = \rho(g_1^{x_1}, g_2^{x_2})$, where ρ is the Chinese remaindering mapping, it holds that $\beta = \alpha^{q' x_1 + p' x_2} (\mathrm{mod} n)$ where $\alpha = \rho(g_1^{(q')^{-1}}, g_2^{(p')^{-1}})$ is a generator of $QR(n)$.*

Proof. First we show that α is a generator of $QR(n)$. Assume without loss of generality that $p' > q'$. Then it holds that $q' \in \mathbf{Z}_{p'}^*$ and as a result q' is an invertible element of $\mathbf{Z}_{p'}^*$. It follows that $g_1' = g_1^{(q')^{-1}}$ is well defined and is a generator of $QR(p)$ (since g_1 is a generator of $QR(p)$). Furthermore $p' (\mathrm{mod} q') \in \mathbf{Z}_{q'}^*$ since it cannot be the case that $p' \equiv_{q'} 0$ as this would mean that either $p' = q'$ or p' is not prime. It follows that p' has an inverse modulo q' and as a result $g_2' = g_2^{(p')^{-1}}$ is well defined and is a generator of $QR(q)$ (since g_2 is a generator of $QR(q)$). Finally we remark that if g_1, g_2 are randomly selected generators of $QR(p), QR(q)$ respectively, it holds that g_1', g_2' are uniformly distributed over all generators.

Since $\alpha = \rho(g_1', g_2')$, it follows that $\alpha \equiv_p g_1'(p)$ and $\alpha \equiv_q g_2'(q)$. It is easy to see that α must be a generator unless the order of α inside $\mathbf{Z}_n^*$ is divisible by either p' or q'; but this can only happen if $\alpha \equiv_p 1$ or $\alpha \equiv_q 1$ something not possible unless either $g_1' \equiv_p 1$ or $g_2' \equiv_q 1$. This case is excluded given that g_1', g_2' are generators of their respective groups $QR(p)$ and $QR(q)$. This completes the argumentation that α is a generator of $QR(n)$.

Now, since $\beta = \rho(g_1^{x_1}, g_2^{x_2})$ it follows that $\beta \equiv g_1^{x_1}(p)$ and $\beta \equiv g_2^{x_2}(q)$; Using this fact together with the properties of α we have:

$$\alpha^{q'x_1+p'x_2} \equiv_p \alpha^{q'x_1} \equiv_p (g_1^{(q')^{-1}})^{q'x_1} \equiv_p g_1^{x_1}$$

$$\alpha^{q'x_1+p'x_2} \equiv_q \alpha^{p'x_2} \equiv_p (g_2^{(p')^{-1}})^{p'x_2} \equiv_p g_2^{x_2}$$

Due to the uniqueness of the Chinese remaindering solution inside $\mathbf{Z}_n^*$ it follows that $\beta = \alpha^{q'x_1+p'x_2} (\mathrm{mod} n)$ is the solution of the system. □

Lemma 10 *Fix a generator α of $QR(n)$ and an integer $t \in \mathbb{N}$. The mapping $\tau_\alpha : \mathbf{Z}_{p'} \times \mathbf{Z}_{q'} \to QR(n)$, with $\tau_\alpha(x_1, x_2) = \alpha^{(q')^t x_1 + (p')^t x_2}$ is a bijection. The inverse mapping τ_α^{-1} is defined as $\tau_\alpha^{-1}(\alpha^x) = \langle (q')^{-t} x \mod p', (p')^{-t} x \mod q' \rangle$.*

Proof. Let $\langle x_1, x_2 \rangle, \langle x_1', x_2' \rangle \in \mathbf{Z}_{p'} \times \mathbf{Z}_{q'}$ be two tuples with $\tau(x_1, x_2) = \tau(x_1', x_2')$. It follows that $(q')^t x_1 + (p')^t x_2 \equiv_{\mathrm{order}(\alpha)} (q')^t x_1' + (p')^t x_2'$; since α is a generator, $p'q' \mid (q')^t(x_1 - x_1') + (p')^t(x_2 - x_2')$, from which we have $p' \mid (q')^t(x_1 - x_1')$ which implies $p' \mid x_1 - x_1'$, i.e., $x_1 = x_1'$. In a similar fashion we show that $x_2 = x_2'$. The onto property follows immediately from the number of elements of the domain and the range.

Regarding the inverse, define q^*, p^* to be integers in $\mathbf{Z}_{p'}, \mathbf{Z}_{q'}$ respectively, so that $q^*(q')^t \equiv_{p'} 1$ and $p^*(p')^t \equiv_{q'} 1$. Moreover let $y_1 = q^* x \mod p'$ and $y_2 = p^* x \mod q'$. We can find integers π_1, π_2 so that $q^* x = \pi_1 p' + y_1$ and $p^* x = \pi_2 q' + y_2$. We will show that $(q')^t y_1 + (p')^t y_2 \equiv_{p'q'} x$ which will complete the proof.

In order for $p'q'$ to divide $(q')^t y_1 + (p')^t y_2 - x$ it should hold that both p', q' divide $(q')^t y_1 + (p')^t y_2 - x$. Indeed, p' divides $(q')^t y_1 + (p')^t y_2 - x$ since $(q')^t y_1 + (p')^t y_2 - x = (q')^t(q^* x - \pi_1 p') + p' y_2 - x \equiv_{p'} (q')^t q^* x - x \equiv_{p'} 0$. In a similar fashion we show that q' divides $(q')^t y_1 + (p')^t y_2 - x$. From these two facts it follows immediately that $\tau(\tau^{-1}(\alpha^x)) = \tau(\langle y_1, y_2 \rangle) = \alpha^x$. □

Let $\mathrm{desc}(1^\nu)$ be a PPT algorithm, called a group descriptor, that on input 1^ν it outputs a description of a cyclic group G denoted by $\tilde{d}_G$. Depending on the group, $\tilde{d}_G$ may have many entries; in our setting it will include a generator of G, denoted by $\tilde{d}_G.\mathsf{gen}$ and the order of G denoted by $\tilde{d}_G.\mathsf{ord}$. We require that $2^{\nu-1} \leq \tilde{d}_G.\mathsf{ord} < 2^\nu$, i.e., the order of G is a ν-bit number with the first bit set. Additionally $\tilde{d}_G$ contains the necessary information that is required to implement multiplication over G. We will be interested in the following two group descriptors:

- desc_p: Given 1^ν find a ν-bit prime $p' > 2^{\nu-1}$ for which it holds that $p = 2p' + 1$ and p is also prime. Let g be any quadratic residue modulo p. We set $QR(p)$ to be the group of quadratic residues modulo p (which in this case is of order p' and is generated by g). The descriptor desc_p returns $\langle g, p, p' \rangle$ and it holds that if $\tilde{d} \leftarrow \mathrm{desc}_\mathsf{p}(1^\nu)$, $\tilde{d}.\mathsf{ord} = p'$ and $\tilde{d}.\mathsf{gen} = g$.
- desc_c: Given ν find two distinct primes p', q' of bit-length $\nu/2$ so that $p'q'$ is a ν-bit number that is greater than $2^{\nu-1}$ and so that there exist primes p, q such that $p = 2p' + 1$ and $q = 2q' + 1$. Let g be any quadratic residue modulo n that is a generator of the group of $QR(n)$ (such element can be found easily). The descriptor desc_c returns $\langle \alpha, n, p, q, p', q' \rangle$ and it holds that if $\tilde{d} \leftarrow \mathrm{desc}_\mathsf{c}(1^\nu)$, $\tilde{d}.\mathsf{ord} = p'q'$ and $\tilde{d}.\mathsf{gen} = \alpha$. The implementation of desc_c that we will consider is the following: execute desc_p twice, to obtain $\tilde{d}_1 = \langle g_1, p, p' \rangle$ and $\tilde{d}_2 = \langle g_2, q, q' \rangle$ with $p \neq q$, and set $\tilde{d} = \langle g, n = pq, p, q, p', q' \rangle$ where $\alpha = \rho(g_1^{(q')^{-1}}, g_2^{(p')^{-1}})$. For such a description $\tilde{d}$ we will call the descriptions $\tilde{d}_1$ and $\tilde{d}_2$, the prime coordinates of $\tilde{d}$.

Now we proceed to define the Decisional Diffie Hellman Problem.

Definition 11 *A Decisional Diffie Hellman (DDH) distinguisher for a group descriptor desc is a PPT algorithm $\mathcal{A}$ with range the set $\{0, 1\}$; the advantage of the distinguisher is defined as follows:*

$$\mathsf{Adv}^{DDH}_{\mathrm{desc},\mathcal{A}}(\nu) = \mathrm{dist}_\mathcal{A}(\mathcal{D}_\nu^{\mathrm{desc}}, \mathcal{R}_\nu^{\mathrm{desc}})$$

where $\mathcal{D}_{\nu}^{\mathrm{desc}}$ *contains elements of the form* $\langle \tilde{d}, g^x, g^y, g^{x\cdot y}\rangle$ *where* $\tilde{d} \leftarrow \mathrm{desc}(1^{\nu})$, $g = \tilde{d}.\mathsf{gen}$ *and* $x, y \leftarrow_R [\tilde{d}.\mathsf{ord}]$, *and* $\mathcal{R}_{\nu}^{\mathrm{desc}}$ *contains elements of the form* $\langle \tilde{d}, g^x, g^y, g^z\rangle$ *where* $\tilde{d} \leftarrow \mathrm{desc}(1^{\nu})$, $g = \tilde{d}.\mathsf{gen}$ *and* $x, y, z \leftarrow_R [\tilde{d}.\mathsf{ord}]$. *Finally we define the overall advantage quantified over all distinguishers as follows:*

$$\mathsf{Adv}_{\mathrm{desc}}^{DDH}(\nu) = \max_{\text{PPT } \mathcal{A}} \mathsf{Adv}_{\mathrm{desc},\mathcal{A}}^{DDH}(\nu)$$

The main result of this section is the theorem below that shows that the DDH over $QR(n)$ *with known factorization* is essentially no easier than the DDH over the prime coordinates of $QR(n)$. The proof of the theorem is based on the construction of a mapping of DDH triples drawn from the two prime coordinate groups of $QR(n)$ into DDH triples of $QR(n)$ that is shown in the following lemma:

Lemma 12 *Let* $\tilde{d} \leftarrow \mathrm{desc}_{\mathsf{c}}(1^{\nu})$ *with* $\tilde{d}_1, \tilde{d}_2 \leftarrow \mathrm{desc}_{\mathsf{p}}(1^{\nu/2})$, *its two prime coordinates, such that* $\tilde{d}_1 = \langle g_1, p, p'\rangle$ *and* $\tilde{d}_2 = \langle g_2, q, q'\rangle$. *The mapping* ρ^* *as follows:*

$$\rho^*(\langle \tilde{d}_1, A_1, B_1, C_1\rangle, \langle \tilde{d}_2, A_2, B_2, C_2\rangle) =_{\mathsf{df}} \langle \tilde{d}, \rho(A_1, A_2), \rho(B_1, B_2), \rho((C_1)^{q'}, (C_2)^{p'})\rangle$$

satisfies the properties (i) $\rho^*(\mathcal{D}_{\nu/2}^{\mathrm{desc}_{\mathsf{p}}}, \mathcal{D}_{\nu/2}^{\mathrm{desc}_{\mathsf{p}}}) \cong \mathcal{D}_{\nu}^{\mathrm{desc}_{\mathsf{c}}}$ *and (ii)* $\rho^*(\mathcal{R}_{\nu/2}^{\mathrm{desc}_{\mathsf{p}}}, \mathcal{R}_{\nu/2}^{\mathrm{desc}_{\mathsf{p}}}) \cong \mathcal{R}_{\nu}^{\mathrm{desc}_{\mathsf{c}}}$, *where* $\cong$ *stands for statistically indistinguishable.*

The mapping ρ^* *will return* $\perp$ *in case* $\tilde{d}_1.\mathsf{ord} = \tilde{d}_2.\mathsf{ord}$. *This is a negligible probability event when selecting* $\tilde{d}_1, \tilde{d}_2$ *at random from* $\mathrm{desc}_{\mathsf{p}}$ *and is the event that contributes the negligible statistical difference in properties (i) and (ii).*

Proof. Observe that if $A_1 = g_1^{x_1}, B_1 = g_1^{y_1}, C_1 = g_1^{x_1 y_1}$ and $A_2 = g_2^{x_2}, B_2 = g_2^{y_2}, C_2 = g_1^{x_2 y_2}$, based on the properties of the mapping ρ shown in lemma 9 it follows that

$$\rho(A_1, A_2) = \alpha^{q' x_1 + p' x_2} \text{ and } \rho(B_1, B_2) = \alpha^{q' y_1 + p' y_2}$$

$$\rho((C_1)^{q'}, (C_2)^{p'}) = \alpha^{(q')^2 x_1 y_1 + (p')^2 x_2 y_2}$$

Now we show that if $\langle A_1, B_1, C_1\rangle$ is a DDH triple from $\tilde{d}_1$, and $\langle A_2, B_2, C_2\rangle$ is a DDH triple from $\tilde{d}_2$ then $\langle A, B, C\rangle$ is a DDH triple from $\tilde{d}$ that has $\tilde{d}_1$ and $\tilde{d}_2$ as its two prime coordinates:

$$\alpha^{\log_\alpha A \log_\alpha B} = \alpha^{(q' x_1 + p' x_2)(q' y_1 + p' y_2)} = \alpha^{(q')^2 x_1 y_1 + (p')^2 x_2 y_2 + p' q' (x_1 y_2 + x_2 y_1)} \equiv_n \alpha^{(q')^2 x_1 y_1 + (p')^2 x_2 y_2} = C$$

From the above and lemma 10 we can deduce easily that $\rho^*(\mathcal{D}_{\nu/2}^{\mathrm{desc}_{\mathsf{p}}}, \mathcal{D}_{\nu/2}^{\mathrm{desc}_{\mathsf{p}}}) = \mathcal{D}_{\nu}^{\mathrm{desc}_{\mathsf{c}}}$. i.e., the distribution defined by ρ^* when applied to two distributions of DDH triples from $\mathcal{D}_{\nu/2}^{\mathrm{desc}_{\mathsf{p}}}$ over the respective groups is statistically close to the distribution $\mathcal{D}_{\nu}^{\mathrm{desc}_{\mathsf{c}}}$. This completes the proof for property (i) of the lemma. Regarding property (ii), observe that if $A_1 = g_1^{x_1}, B_1 = g_1^{y_1}, C_1 = g_1^{z_1}$ and $A_2 = g_2^{x_2}, B_2 = g_2^{y_2}, C_2 = g_1^{z_2}$, based on the properties of the mapping ρ shown in lemma 9 it follows that

$$\rho(A_1, A_2) = \alpha^{q' x_1 + p' x_2} \text{ and } \rho(B_1, B_2) = \alpha^{q' y_1 + p' y_2}$$

$$\rho((C_1)^{q'}, (C_2)^{p'}) = \alpha^{(q')^2 z_1 + (p')^2 z_2}$$

and thus $\rho^*(\mathcal{R}_{\nu/2}^{\mathrm{desc}_{\mathsf{p}}}, \mathcal{R}_{\nu/2}^{\mathrm{desc}_{\mathsf{p}}}) = \mathcal{R}_{\nu}^{\mathrm{desc}_{\mathsf{c}}}$ follows easily from lemma 10. □

Theorem 13 $\mathsf{Adv}_{\mathrm{desc}_{\mathsf{c}}}^{DDH}(\nu) \leq 2\mathsf{Adv}_{\mathrm{desc}_{\mathsf{p}}}^{DDH}(\nu/2)$.

Proof. Let $\mathcal{A}$ be any DDH-distinguisher for $\mathrm{desc_c}$. Consider the following PPT $\mathcal{A}_1$: $\mathcal{A}_1$ takes as input a description $\tilde{d}_1 \leftarrow \mathrm{desc_p}(1^{\nu/2})$, with $\tilde{d}_1 = \langle g_1, p, p' \rangle$ and a triple τ_1 of $QR(p)$; $\mathcal{A}_1$ operates as follows: then samples a quadruple $\langle \tilde{d}_2, \tau_2 \rangle$ of $\mathcal{D}_{\nu/2}^{\mathrm{desc_p}}$ and then simulates $\mathcal{A}$ on input $\rho^*(\langle \tilde{d}_1, \tau_1 \rangle, \langle \tilde{d}_2, \tau_2 \rangle)$, where ρ^* is the mapping defined in lemma 12. Using property (i) of lemma 12, we have that $\mathcal{D}_{\nu}^{\mathrm{desc_c}} \cong \rho^*(\mathcal{D}_{\nu/2}^{\mathrm{desc_p}}, \mathcal{D}_{\nu/2}^{\mathrm{desc_p}})$ and thus,

$$\text{(Fact 1)} \qquad \mathrm{dist}_{\mathcal{A}}(\mathcal{D}_{\nu}^{\mathrm{desc_c}}, \rho^*(\mathcal{R}_{\nu/2}^{\mathrm{desc_p}}, \mathcal{D}_{\nu/2}^{\mathrm{desc_p}})) = \mathsf{Adv}^{DDH}_{\mathrm{desc_p},\mathcal{A}_1}(\nu/2) \leq \mathsf{Adv}^{DDH}_{\mathrm{desc_p}}(\nu/2)$$

Consider now the PPT $\mathcal{A}_2$ that takes as input a description $\tilde{d}_2 \leftarrow \mathrm{desc_p}(1^{\nu/2})$ with $\tilde{d}_2 = \langle g_2, q, q' \rangle$ and a triple τ_2 over $QR(q)$. $\mathcal{A}_2$ samples a quadruple $\langle \tilde{d}_1, \tau_1 \rangle$ of $\mathcal{R}_{\nu/2}^{\mathrm{desc_p}}$ and simulates $\mathcal{A}$ on input $\rho^*(\langle \tilde{d}_1, \tau_1 \rangle, \langle \tilde{d}_2, \tau_2 \rangle)$. Using property (ii) of lemma 12 we have that $\mathcal{R}_{\nu}^{\mathrm{desc_c}} \cong \rho^*(\mathcal{R}_{\nu/2}^{\mathrm{desc_p}}, \mathcal{R}_{\nu/2}^{\mathrm{desc_p}})$ and thus,

$$\text{(Fact 2)} \qquad \mathrm{dist}_{\mathcal{A}}(\rho^*(\mathcal{R}_{\nu/2}^{\mathrm{desc_p}}, \mathcal{D}_{\nu/2}^{\mathrm{desc_p}}), \mathcal{R}_{\nu}^{\mathrm{desc_c}}) = \mathsf{Adv}^{DDH}_{\mathrm{desc_p},\mathcal{A}_2}(\nu/2) \leq \mathsf{Adv}^{DDH}_{\mathrm{desc_p}}(\nu/2)$$

Finally by applying the triangle inequality to facts 1 and 2 above, we obtain:

$$\mathsf{Adv}^{DDH}_{\mathcal{A},\mathrm{desc_c}}(\nu) = \mathrm{dist}_{\mathcal{A}}(\mathcal{D}_{\nu}^{\mathrm{desc_c}}, \mathcal{R}_{\nu}^{\mathrm{desc_c}}) \leq 2\mathsf{Adv}^{DDH}_{\mathrm{desc_p}}(\nu/2)$$

Since the above holds for an arbitrary choice of $\mathcal{A}$ the theorem follows. □

Then we proceed to state explicitly the two variants of the assumption:

Definition 14 *The following are two Decisional Diffie Hellman Assumptions:* • *The DDH assumption over quadratic residues for groups of prime order (*DDH-Prime*) asserts that:*

$$\mathsf{Adv}^{DDH}_{\mathrm{desc_p}}(\nu) = \mathsf{negl}(\nu)$$

• *The DDH assumption over quadratic residues for groups of composite order with known Factorization (*DDH-Comp-KF*) asserts that:*

$$\mathsf{Adv}^{DDH}_{\mathrm{desc_c}}(\nu) = \mathsf{negl}(\nu)$$

Theorem 15 DDH-Prime $\Longrightarrow$ DDH-Comp-KF.

Proof. The DDH assumption over the quadratic residues for prime order suggests that for all $c > 0$ there exists a $\nu_c \in \mathbf{Z}$ such that $\mathsf{Adv}^{DDH}_{\mathrm{desc_p}}(\nu) < \nu^{-c}$ for all $\nu \geq \nu_c$. Now we have to show that for any $c > 0$ we can find some value for ν beyond which it holds that $\mathsf{Adv}^{DDH}_{\mathrm{desc_c}}(\nu) < \nu^{-c}$. Fix some arbitrary $c > 0$. Let ν_{2c} be such that for all $\nu \geq \nu_{2c}$ it holds that $\mathsf{Adv}^{DDH}_{\mathrm{desc_p}}(\nu) < \nu^{-2c}$. Using theorem 13 we have that for all $\nu \geq \nu_{2c}$, $\mathsf{Adv}^{DDH}_{\mathrm{desc_c}}(\nu) \leq 2\mathsf{Adv}^{DDH}_{\mathrm{desc_p}}(\nu/2) < 2(\nu/2)^{-2c} = 2^{2c+1}/\nu^{2c} < \nu^{-c}$, where the last inequality holds if $\nu > \sqrt[c]{2^{2c+1}}$. We conclude that for any c it holds that $\mathsf{Adv}^{DDH}_{\mathrm{desc_c}}(\nu) < \nu^{-c}$, provided that $\nu \geq \max\{\sqrt[c]{2^{2c+1}}, \nu_{2c}\}$. □

4 CCA2 PK-Encryption over $QR(n)$ with known factorization

Our constructions will require an identity embedding mechanism that is CCA2 secure; such a mechanism is presented in this section.

A public-key encryption scheme comprises three procedures $\langle \mathtt{Gen}, \mathtt{Enc}, \mathtt{Dec} \rangle$. The syntax of these procedures is as follows: $\mathtt{Gen}(1^{\nu})$ returns a pair $\langle \mathsf{pk}, \mathsf{sk} \rangle$ that constitutes the public-key and secret-key of the scheme respectively. The probabilistic encryption function $\mathtt{Enc}$ takes as input the parameter 1^{ν}, a public-key pk and a

message m and returns a ciphertext ψ. The decryption function $\mathtt{Dec}$ takes as input a secret-key sk and a ciphertext ψ and returns either the corresponding plaintext m, or the special failure value $\perp$. The soundness of a public-key encryption requires that for any $\langle \mathsf{pk}, \mathsf{sk} \rangle$, $\mathtt{Dec}(\mathsf{sk}, \mathtt{Enc}(1^\nu, \mathsf{pk}, m)) = m$ with very high probability in the security parameter ν (preferably always). There are various notions of security for public-key encryption [GM84, NY90, RS92, DDN00], below we will be interested in the so-called CPA and CCA2 security in the indistinguishability sense. For completeness we define these notions below:

A CCA2 adversary $\mathcal{A}$ against a public-key encryption scheme $\langle \mathtt{Gen}, \mathtt{Enc}, \mathtt{Dec} \rangle$ is a PPT predicate with range in $\{0,1\}$ that is thought to operate in the following game:

The CCA2 Game $G^{\mathcal{A}}_{\mathsf{cca2}}$ for security parameter ν (denoted by $G^{\mathcal{A}}_{\mathsf{cca2}}(1^\nu)$):

1. $\langle \mathsf{pk}, \mathsf{sk} \rangle \leftarrow \mathtt{Gen}(1^\nu)$;
2. $\langle aux, m_0, m_1 \rangle \leftarrow \mathcal{A}^{\mathtt{Dec}(\mathsf{sk},\cdot)}(\mathsf{choose}, 1^\nu, \mathsf{pk})$
3. Choose $b \leftarrow_R \{0,1\}$;
4. Set $\psi^* \leftarrow \mathtt{Enc}(1^\nu, \mathsf{pk}, m_b)$;
5. Set $\mathtt{Dec}^{\neg\psi^*}(\mathsf{sk}, x)$ to be "$\mathtt{if}\ x \neq \psi^*\ \mathtt{then}\ \mathtt{return}\ \mathtt{Dec}(\mathsf{s}, x)\ \mathtt{else}\ \mathtt{return}\ \perp$";
6. $b^* \leftarrow \mathcal{A}^{\mathtt{Dec}^{\neg\psi^*}[\mathsf{sk},\cdot]}(\mathsf{guess}, aux, \psi^*)$;
7. if $b = b^*$ return $\top$ else return $\perp$;

A CPA adversary $\mathcal{A}$ operates as above but is denied access to the Dec oracles in steps 2 and 6 in the above game. The corresponding restricted game is called $G^{\mathcal{A}}_{\mathsf{cpa}}$.

Definition 16 *For* $\mathsf{X} \in \{\mathsf{cca2}, \mathsf{cpa}\}$*, A public-key encryption scheme satisfies* X*-security if for any* PPT *predicate* $\mathcal{A}$ *it holds that* $2\mathbf{Prob}[G^{\mathcal{A}}_{\mathsf{X}}(1^\nu) = \top] - 1 = \mathsf{negl}(\nu)$.

4.1 An ElGamal CCA2 variant over $QR(n)$ with known factorization in the RO Model

Consider the following cryptosystem $\langle \mathtt{Gen}_{qr}, \mathtt{Enc}_{qr}, \mathtt{Dec}_{qr} \rangle$:

- The key-generator $\mathtt{Gen}_{qr}$ on input 1^ν samples the description $\tilde{d} = \langle g, n, p, q, p', q' \rangle \leftarrow \mathrm{desc}_{\mathsf{c}}(1^\nu)$, selects a value $x \leftarrow_R [p'q']$ and outputs $\mathsf{pk} = \langle g, n, p, q, h = g^x \rangle$ and $\mathsf{sk} = x$.
- The encryption function $\mathtt{Enc}_{qr}$ operates as follows: given $M \in QR(n)$, it selects $r \leftarrow_R [\lfloor n/4 \rfloor]$ and returns the pair $\langle g^r, h^r M \rangle$.
- The decryption operation $\mathtt{Dec}_{qr}$ is given $\langle G, H \rangle$ and returns $G^{-x}H (\mathrm{mod} n)$.

Note that this cryptosystem is an ElGamal variant over quadratic residues modulo a composite, so that

(i) the factorization is available to the adversary, but:
(ii) the factorization is not necessary for encryption.

Theorem 17 *The cryptosystem* $\langle \mathtt{Gen}_{qr}, \mathtt{Enc}_{qr}, \mathtt{Dec}_{qr} \rangle$ *described above satisfies* CPA*-security under the assumption* DDH-Compo-KF*, and thus under the assumption* DDH-Prime *(theorem 15).*

Proof. The proof of CPA-security for the ElGamal variant we define is similar to the proof of CPA-security for the proof of semantic security for the regular ElGamal encryption, see [TY98]. □

We remark that ElGamal variants over composite order groups have been considered before, e.g., [McC88]; in the setup that was considered the adversary was denied the factorization and security properties of the cryptosystem were associated with the factoring assumption. Our variant above, on the other hand, shows that the semantic security (in the sense of CPA-security) of the composite modulus ElGamal variant we define still holds under the standard prime-order Decisional Diffie-Hellman assumption DDH-Prime.

Now let us turn our attention to achieving CCA2 security in the above setting. Double encryption has been employed as a tool to obtain chosen-ciphertext security [NY90]. The "twin-conversion" has been formalized in [FP01] and transforms a CPA-secure cryptosystem into a CCA2-cryptosystem $\langle \mathtt{Gen}', \mathtt{Enc}', \mathtt{Dec}' \rangle$ as follows:

- $\mathtt{Gen}'$ performs two independent executions of $\mathtt{Gen}$ to obtain the public-key $\mathsf{pk}' = \langle \mathsf{pk}_1, \mathsf{pk}_2 \rangle$ and the secret-keys $\mathsf{sk}' = \langle \mathsf{sk}_1, \mathsf{sk}_2 \rangle$.
- The encryption algorithm $\mathtt{Enc}'$, given a plaintext m, it outputs

$$\mathtt{Enc}'(\mathsf{pk}', m) = \langle c_1, c_2, \sigma \rangle = \langle \mathtt{Enc}(\mathsf{pk}_1, m), \mathtt{Enc}(\mathsf{pk}_2, m), \sigma \rangle$$

  where σ is a non-interactive proof that shows that the two ciphertext outputs of $\mathtt{Enc}'$, namely c_1, c_2, together with the public-key pk' belong to the language $\mathcal{L} = \{\langle \mathsf{pk}_1, \mathsf{pk}_2, \mathtt{Enc}(\mathsf{pk}_1, m), \mathtt{Enc}(\mathsf{pk}_2, m) \rangle \mid m\}$.
- The decryption $\mathtt{Dec}'$ verifies the non-interactive proof of language membership and if it is correct it returns the decryption of one of the two ciphertexts; otherwise $\mathtt{Dec}'$ returns $\perp$.

In [FP01] the following theorem was shown:

Theorem 18 *[FP01] The cryptosystem $\langle \mathtt{Gen}', \mathtt{Enc}', \mathtt{Dec}' \rangle$ described above is* CCA2*-secure provided that (i) the underlying cryptosystem $\langle \mathtt{Gen}, \mathtt{Enc}, \mathtt{Dec} \rangle$ is* CPA*-secure, and (ii) the non-interactive proof of language membership employed in $\mathtt{Enc}'$ is* simulation-sound*.*

A non-interactive proof of language-membership for a language $\mathcal{L}$ is called simulation-sound [Sah99] if it is hard for an adversary that possesses a pair $\langle x^*, c^* \rangle$ so that c^* is a valid noninteractive proof of the fact $x \in \mathcal{L}$ to produce another pair $\langle x, c \rangle$ for any $x \notin \mathcal{L}$ (i.e., the adversary should not be able to a forge a proof). More formally, a proof of language membership will be called simulation-sound, if for all PPT $\mathcal{A}$ it holds that the probability $\mathsf{Succ}_{\mathcal{A}}^{sim-nizk} = \mathbf{Prob}[\langle x, c \rangle \leftarrow \mathcal{A}(x^*, c^*) \mid x \notin \mathcal{L} \wedge \langle x, c \rangle \neq \langle x^*, c^* \rangle]$ is negligible.

Below we apply the twin-transform to the ElGamal variant we presented in the beginning of the section, from now on we work in the random oracle model (the hash function will be treated as idealized hash which is a random oracle):

- $\mathtt{Gen}'_{qr}$ samples $\langle g, n, p, q, p', q' \rangle \leftarrow \mathrm{desc}_{\mathsf{c}}(1^{\nu})$, selects $x_1, x_2 \leftarrow_R [p'q']$ and returns the $\mathsf{pk}' = \langle g, n, p, q, y_1 = g^{x_1}, y_2 = g^{x_2} \rangle$ and the secret-key $\mathsf{sk}' = \langle x_1, x_2 \rangle$.
- The encryption $\mathtt{Enc}'_{qr}$: in order to encrypt a message m, we form the two ciphertexts $\langle g^{r_1}, y_1^{r_1} m \rangle$ and $\langle g^{r_2}, y_1^{r_2} m \rangle$ with $r_1, r_2 \leftarrow [\lfloor n/4 \rfloor]$ and we attach a proof of language membership for the language:

$$\mathcal{L}_{qr} = \{\langle n, g, y_1, y_2, \langle g^{r_1}, y_1^{r_2} m \rangle, \langle g^{r_2}, y_2^{r_2} m \rangle \rangle \mid r_1, r_2 \in [\lfloor n/4 \rfloor], m \in QR(n)\}$$

  Note that we want to preserve the property that encryption does not use the factorization of n. In order to prove language membership of a tuple $\langle n, g, y_1, y_2, \langle G_1, Y_1 \rangle, \langle G_2, Y_2 \rangle \rangle$ to $\mathcal{L}_{qr}$ it suffices to present a proof of knowledge for the discrete-log relation set (see section 2.3) $\langle \rho_1, 0, 0, -1, 0, 0, 0 \rangle$, $\langle \rho_2, 0, 0, 0, 0, -1, 0 \rangle$, $\langle 0, \rho_1, \rho_2, 0, 1, 0, -1 \rangle$ defined over the base elements $g, (y_1)^{-1}, y_2, G_1, Y_1, G_2, Y_2$.

  It follows that the output of $\mathtt{Enc}'_{qr}$ is of the form $\langle G_1, Y_1, G_2, Y_2, \pi \rangle$, where π is the non-interactive proof of language membership in $\mathcal{L}_{qr}$. In definition 19 below we show how the proof π is derived from the methodology of section 2.3.
- The decryption $\mathtt{Dec}'_{qr}$ is as in the twin conversion description.

Definition 19 The proof of language membership for $\mathcal{L}_{qr}$. *Assuming that the values $r_1, r_2 \in S(2^{\ell}, 2^{\mu})$, where ℓ, μ are parameters such that $S(2^{\ell}, 2^{\mu}) \cong [\lfloor n/4 \rfloor]$, the proof of knowledge of a discrete-log relation set described above, suggest that the prover selects $t_1, t_2 \in \pm\{0,1\}^{\epsilon(\mu+k)}$, and transmit to the verifier the values $B_1 = g^{t_1}, B_2 = g^{t_2}, B_3 = y_2^{t_2}/y_1^{t_1}$. The verifier selects a challenge $c \in \{0,1\}^k$, and subsequently the prover computes $s_i = t_i - c(r_i - 2^{\ell})$ for $i = 1, 2$ and transmits to the verifier the values s_1, s_2. The verification check is the following: $g^{s_1} =_? B_1(g^{2^{\ell}}/G_1)^c$, $g^{s_2} =_? B_2(g^{2^{\ell}}/G_2)^c$ and $y_2^{s_2}/y_1^{s_1} = B_3(Y_1/Y_2)^c(y_2/y_1)^{c2^{\ell}}$. In order to*

make the proof non-interactive using a hash function $\mathcal{H} : \{0,1\}^* \to \{0,1\}^k$ and we perform the following: the non-interactive proof π in the description of $\mathtt{Enc}'_{qr}$ will have the form

$$\langle c = \mathcal{H}\Big(g, (y_1)^{-1}, y_2, G_1, Y_1, G_2, Y_2, B_1, B_2, B_3\Big), s_1, s_2\rangle$$

and the verification step that is part of $\mathtt{Dec}'_{qr}$, operates as follows: given the non-interactive proof $\pi = \langle c, s_1, s_2\rangle$, the check is implemented as:

$$c =_? \mathcal{H}\Big(g, (y_1)^{-1}, y_2, G_1, Y_1, G_2, Y_2, G_1^c g^{s_1 - c2^\ell}, G_2^c g^{s_2 - c2^\ell}, \frac{y_2^{s_2 - c2^\ell} Y_2^c}{y_1^{s_1 - c2^\ell} Y_1^c}\Big)$$

The proof π constructed as above will be denoted by $\mathtt{nizk}^{\mathcal{H}}[n, g, y_1, y_2, \langle G_1, Y_1\rangle, \langle G_2, Y_2\rangle]$.

From theorem 8 we know that the above proof is complete, sound and honest verifier zero-knowledge in the statistical sense, provided that $p', q' > 2^k$ (note that we are in the setting where the adversary is allowed to know the factorization of n — on the contrary we preserve the property that encryption in our twin ElGamal variant does not require the factorization of n).

Now observer that $\langle \mathtt{Gen}'_{qr}, \mathtt{Enc}'_{qr}, \mathtt{Dec}'_{qr}\rangle$ will be CCA2-secure based on theorems 17 and 18, as long as the non-interactive proof of knowledge described in definition 19 satisfies simulation-soundness. We argue about this fact in the following theorem:

Theorem 20 *Let $\mathcal{A}$ be a* PPT *adversary that is given (i) $n, p, q, g, y_1 = g^{x_1}, y_2 = g^{x_2}$ where $\langle g, n, p, q, y_1, y_2\rangle = \mathsf{pk}'$ is distributed according to $\mathtt{Gen}'_{qr}$, (ii) $\langle G_1, Y_1, G_2, Y_2, \pi\rangle \leftarrow \mathtt{Enc}'_{qr}(\mathsf{pk}', m)$ for any fixed known message m, and (iii) access to the random oracle $\mathcal{H}$, it returns a tuple $\langle G'_1, Y'_1, G'_2, Y'_2, \pi'\rangle$ so that the verification step of $\mathtt{Dec}'_{qr}$ (in definition 19) passes but it holds that $Y'_1/(G'_1)^{x_1} \neq Y'_2/(G'_2)^{x_2}$. Assuming that $p', q' > 2^k$, it holds that the success probability of $\mathcal{A}$ is negligible.*

Proof. Let g, n, p, q, y_1, y_2 be parameters distributed as in $\mathtt{Gen}'_{qr}$.

Consider now a procedure $\mathcal{P}$ that has access to the random oracle $\mathcal{H}$ and the reprogramming oracle $\mathcal{R}$. $\mathcal{P}$ takes as input g, n, p, q, y_1, y_2, and for a fixed message m it constructs a ciphertext as follows: $\langle G_1, G_2, Y_1, Y_2, \pi_{\mathsf{simul}}\rangle = \langle g^{r_1}, y_1^{r_1} M, g^{r_2}, y_2^{r_2} M, \pi_{\mathsf{simul}}\rangle$ where $\pi_{\mathsf{simul}} = \langle c, s_1, s_2\rangle$ is a simulated proof of language membership for G_1, Y_1, G_2, Y_2 obtained by virtue of theorem 8 item (ii). Note that $\mathcal{P}$ does not need to know r_1, r_2. $\mathcal{P}$ then uses $\mathcal{R}$ to reprogram $\mathcal{H}$ as follows

$$\mathcal{H} \text{ on input } \Big\langle g, (y_1)^{-1}, y_2, G_1, Y_1, G_2, Y_2, G_1^c g^{s_1 - c2^\ell}, G_2^c g^{s_2 - c2^\ell}, \frac{y_2^{s_2 - c2^\ell} Y_2^c}{y_1^{s_1 - c2^\ell} Y_1^c}\Big\rangle \text{ answers } c$$

Subsequently, $\mathcal{P}$ simulates $\mathcal{A}$ by providing the input $\langle g, n, p, q, y_1, y_2\rangle$ and $\langle G_1, Y_1, G_2, Y_2, \pi_{\mathsf{simul}}\rangle$ as above. Whenever $\mathcal{A}$ makes a query x to the random oracle $\mathcal{H}$, $\mathcal{P}$ passes it directly to the random oracle $\mathcal{H}$.

It is easy to see that the output distributions of $\mathcal{P}$ and $\mathcal{A}$ are statistically indistinguishable. Now let α be the non-negligible probability of the event that $\mathcal{P}$ outputs a $\langle G'_1, Y'_1, G'_2, Y'_2, \pi\rangle$ so that it is a valid ciphertext according to the test of $\mathtt{Dec}'_{qr}$ and moreover, $Y'_1/(G'_1)^{x_1} \neq Y'_2/(G'_2)^{x_2}$ which is equivalent to $Y'_1/Y'_2 \neq (G'_1)^{x_1}/(G'_2)^{x_2}$.

Based on lemma 6, we can derive a procedure $\mathcal{P}'$ that outputs with the proper probability two valid language membership proofs π_1, π_2 for the ciphertext $\langle G'_1, Y'_1, G'_2, Y'_2\rangle$. Based on the soundness of the proof of language membership as argued in theorem 8 item (iii)$'$, we can extract with non-negligible probability a witness r_1, r_2 so that $G'_1 = g^{r_1}, G'_2 = g^{r_2}, Y'_1/Y'_2 = y_1^{r_1}/y_2^{r_2}$. It follows that it holds $(G'_1)^{x_1}/(G'_2)^{x_2} \neq y_1^{r_1}/y_2^{r_2}$. But this is contradiction by the definition of y_1, y_2 and the condition imposed on G'_1, G'_2. It follows that our assumption that α is non-negligible is inconsistent and thus α must be negligible from which the theorem follows. □

5 Group Signatures: Model and Definitions

The parties that are involved in a group signature scheme are the Group Manager (GM) and the users. In the definition below we give a formal syntax of the five procedures the primitive is based on.

Our formalization is geared towards schemes as the [ACJT00] scheme where users are joining the system by executing a join-dialog with the GM (and not any other trusted entity or tamper-proof element exists). Naturally, this formalization can capture *also* the case where a third party creates the user signing keys privately and distributes them through private channels and with trusted parties, however we do not deal with these easier case in our model (we remark later how such simplifying assumptions can be introduced and how properties specified in our model apply to them). We emphasize that our join dialog does not require a private channel between the GM and the user.

Definition 21 *A group signature scheme is a digital signature scheme that comprises of the following five procedures;*

`SETUP`*: On input a security parameter 1^ν, this probabilistic algorithm outputs the group public key $\mathcal{Y}$ (including all system parameters) and the secret key $\mathcal{S}$ for the GM. Note that* `SETUP` *is* not *supposed to output the members' signing keys. Moreover* `SETUP` *initializes a public-state string St with two components $St_{users} = \emptyset$ and $St_{trans} = \epsilon$.*

`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 denote the i-th user's membership certificate by cert_i and the corresponding membership secret by sec_i. Since* `JOIN` *is a protocol, it is made out of two interactive Turing Machines (ITM) $\mathsf{J_{user}}, \mathsf{J_{GM}}$. Only $\mathsf{J_{user}}$ has a private output tape. An execution of the protocol is denoted as $[\mathsf{J_{user}}(1^\nu, \mathcal{Y}), \mathsf{J_{GM}}(1^\nu, \mathcal{Y}, \mathcal{S})]$ and has two "output" components: the private output of the user, $\langle i, \mathsf{cert}_i, \mathsf{sec}_i\rangle \leftarrow \mathsf{U}[\mathsf{J_{user}}(1^\nu, \mathcal{Y}), \mathsf{J_{GM}}(1^\nu, St, \mathcal{Y}, \mathcal{S})]$ and the public communication transcript, $\langle i, \mathsf{transcript}_i\rangle \leftarrow \mathsf{T}[\ \mathsf{J_{user}}(1^\nu, \mathcal{Y}), \mathsf{J_{GM}}(1^\nu, St, \mathcal{Y}, \mathcal{S})]$. After a successful execution of* `JOIN` *the following public updates are made: $St_{users} = St_{users} \cup \{i\}$ and $St_{trans} = St_{trans}||\langle i, \mathsf{transcript}_i\rangle$.*

`SIGN`*: A probabilistic algorithm that given a group's public-key, a membership certificate, a membership secret, and a message m outputs a 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.*

`VERIFY`*: An algorithm for establishing the validity of an alleged group signature of 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 $\texttt{OPEN}(m, \sigma, \mathcal{Y}, \mathcal{S}, St) \in St_{users} \cup \{\bot\}$.*

Notation. We will write $\langle \mathsf{cert}_i, \mathsf{sec}_i\rangle \leftrightharpoons_{\mathcal{Y}} \langle \mathsf{transcript}_i\rangle$ to denote the relationship between the private output of J_{user} and the public-transcript when the protocol is executed based on the group public-key $\mathcal{Y}$. Moreover, any given cert, based on $\mathcal{Y}$, has a unique corresponding sec; we will also denote this relationship by $\mathsf{cert} \leftrightharpoons_{\mathcal{Y}} \mathsf{sec}$. We remark that $\leftrightharpoons_{\mathcal{Y}}$ in both cases, is a polynomial-time relationship in the parameter ν.

5.1 Correctness

The correctness of a group signature scheme is broken down in four individual properties: (i) *user tagging soundness* mandates that users are assigned a unique tag (depending on order of joining) by the `JOIN` protocol; (ii) *join soundness* mandates that the private output tape of $\mathsf{J_{user}}$ after a successful execution of the `JOIN` dialog contains a valid membership certificate and membership secret; (iii) *signing soundness* mandates that the group signature scheme behaves like a digital signature; (iv) *opening soundness* mandates that the `OPEN` algorithm succeeds in identifying the originator of any signature generated according to specifications. Formally,

Definition 22 *A group signature is correct if the following statements hold with very high probability over the coin tosses of all procedures. Let* $\langle \mathcal{Y}, \mathcal{S} \rangle \leftarrow \mathtt{SETUP}(1^\nu)$.

- User tagging soundness. *If* $\langle i, \mathsf{transcript}_i \rangle \leftarrow \mathsf{T}[\, \mathsf{J}_{\mathsf{user}}(1^\nu, \mathcal{Y}), \mathsf{J}_{\mathsf{GM}}(1^\nu, St, \mathcal{Y}, \mathcal{S})]$ *then* $i = \max St_{users} + 1$ *(where* $\max \emptyset = 0$*, and the occurrence of* St *here is before the update* $St_{users} = St_{users} \cup \{i\}$*).*
- Join soundness. *If* $\langle i, \mathsf{cert}_i, \mathsf{sec}_i \rangle \leftarrow \mathsf{U}[\, \mathsf{J}_{\mathsf{user}}(1^\nu, \mathcal{Y}), \mathsf{J}_{\mathsf{GM}}(1^\nu, St, \mathcal{Y}, \mathcal{S})]$ *then it holds that* $\mathsf{cert}_i \leftrightharpoons_{\mathcal{Y}} \mathsf{sec}_i$.
- Signing soundness. *For any* $\mathsf{cert} \leftrightharpoons_{\mathcal{Y}} \mathsf{sec}$*, and any message* m, $\mathtt{VERIFY}(\mathcal{Y}, m, \mathtt{SIGN}(\mathcal{Y}, \mathsf{cert}, \mathsf{sec}, m)) = \top$.
- Opening soundness. *For any* $\langle \mathsf{cert}_i, \mathsf{sec}_i \rangle \leftrightharpoons_{\mathcal{Y}} \langle \mathsf{transcript}_i \rangle$*, with* $\langle i, \mathsf{transcript}_i \rangle \in St_{trans}$*, any message* m*, and any* $\sigma \leftarrow \mathtt{SIGN}(\mathcal{Y}, \mathsf{cert}_i, \mathsf{sec}_i, m)$ *it holds that* $\mathtt{OPEN}(m, \sigma, \mathcal{Y}, \mathcal{S}, St) = i$.

5.2 Security

Below we present the general model for security. A number of oracles are specified. Through these oracles the adversary may interact with an Interface that represents the system in the real world, and simulates its operation (i.e., a simulator) in the security proof. This allows us to model adversaries with capabilities (modeled by subsets of the oracles) and attack goals in mind, in the spirit of [GMR84]. However, since we deal with a "privacy primitive" we have to deal with a number of goals of mutually distrusting and mutually attacking parties, thus we need more than one adversarial scenario. The interface $\mathcal{I}$ is an ITM that is initialized with a state $\mathsf{state}_{\mathcal{I}} = \langle St, \mathcal{Y}, \mathcal{S} \rangle \leftarrow \mathtt{SETUP}(1^\nu)$ that accepts the following types of queries (in a stateful fashion):

- $\mathcal{Q}_{\mathsf{pub}}$ and $\mathcal{Q}_{\mathsf{key}}$: the interface returns the public-and secret-key respectively.
- $\mathcal{Q}_{\mathsf{a-join}}$: the interface initiates a protocol dialog simulating J_{GM}. The user created from this interaction and entered in St_{users} is marked as U^a (adversarially controlled).
- $\mathcal{Q}_{\mathsf{p-join}}$: the interface simulates in private an instantiation of the JOIN protocol dialog. The user created from this interaction and entered in St_{users} is marked as U^p. The resulting membership certificate and membership secret will be appended in $\mathsf{state}_{\mathcal{I}}$.
- $\mathcal{Q}_{\mathsf{b-join}}$: the interface initiates a protocol dialog simulating $\mathsf{J}_{\mathsf{user}}$. The user created from this interaction will be also entered in St_{users} and will be marked by U^b. The resulting membership certificate and membership secret will be appended in $\mathsf{state}_{\mathcal{I}}$.
- $\mathcal{Q}_{\mathsf{corr}}(i)$: given that $i \in U^p \cup U^b$ the interface recovers $\langle \mathsf{cert}_i, \mathsf{sec}_i \rangle$ from $\mathsf{state}_{\mathcal{I}}$ and returns $\langle \mathsf{cert}_i, \mathsf{sec}_i \rangle$. If U is a set of users we denote by $\mathcal{Q}^{\neg U}_{\mathsf{corr}}(i)$ the operation of the corrupt oracle when queries for users in U are declined.
- $\mathcal{Q}_{\mathsf{sign}}(i, m)$: given that $i \in U^p \cup U^b$ the interface simulates a signature on m by looking up the membership certificate and membership secret available from the execution of either a $\mathcal{Q}_{\mathsf{p-join}}$ or $\mathcal{Q}_{\mathsf{b-join}}$ query and returns the corresponding signature.
- $\mathcal{Q}_{\mathsf{open}}(\sigma)$: the interface applies the opening algorithm to the given signature σ. If S is a set of signatures we denote by $\mathcal{Q}^{\neg S}_{\mathsf{open}}$ the operation of the opening oracle when queries for signatures in S are declined.

We remark that the interface $\mathcal{I}$ maintains a history of all queries posed to the above oracles (if these queries accepted an input); for instance, we use the notation $\mathsf{hist}_{\mathcal{I}}(\mathcal{Q}_{\mathsf{sign}})$ to denote the history of all signature queries.

Security Modeling. We next define our security model, which involve three attack scenarios and security against them. Our first security property relates to an outsider of the system. In an Outsider-Attack the adversary is allowed to observe the operation of the system by asking the interface to introduce users through $\mathcal{Q}_{\mathsf{p-join}}$ queries, obtain signatures from such users, open their signatures and finally produce a forged group signature (cf. an existential adaptive chosen message attack, [GMR84]).

The Outsider-Attack Game $G^{\mathcal{A}}_{\mathsf{out}}$ for security parameter ν (denoted by $G^{\mathcal{A}}_{\mathsf{out}}(1^\nu)$):

1. $\mathsf{state}_{\mathcal{I}} = \langle St, \mathcal{Y}, \mathcal{S}\rangle \leftarrow \mathtt{SETUP}(1^\nu)$;
2. $\langle m, \sigma\rangle \leftarrow \mathcal{A}^{\mathcal{I}[\mathcal{Q}_{\mathsf{pub}}, \mathcal{Q}_{\mathsf{p-join}}, \mathcal{Q}_{\mathsf{sign}}, \mathcal{Q}_{\mathsf{open}}]}(1^\nu)$
3. If $(\mathtt{VERIFY}(\mathcal{Y}, m, \sigma) = \top) \wedge (\forall i : (i, m) \notin \mathsf{hist}_{\mathcal{I}}(\mathcal{Q}_{\mathsf{sign}}))$ then return $\top$ else return $\bot$.

Our second security property relates to an insider type of attack. Here the whole system conspires against the user. The adversary is in control not only of coalitions of users but of the GM itself. It is allowed to introduce "good" users into the system by issuing $\mathcal{Q}_{\mathsf{b-join}}$ queries to the interface and obtain signatures from them. Finally the adversary produces a signature that opens to one of the "good" users (cf. this attack is akin of an existential adaptive chosen message attack [GMR84] but with an "opening" challenge in mind, since it is protecting the user side).

The Insider-Attack Game $G^{\mathcal{A}}_{\mathsf{in}}$ for security parameter ν (denoted by $G^{\mathcal{A}}_{\mathsf{in}}(1^\nu)$):

1. $\mathsf{state}_{\mathcal{I}} = \langle St, \mathcal{Y}, \mathcal{S}\rangle \leftarrow \mathtt{SETUP}(1^\nu)$;
2. $\langle m, \sigma\rangle \leftarrow \mathcal{A}^{\mathcal{I}[\mathcal{Q}_{\mathsf{pub}}, \mathcal{Q}_{\mathsf{key}}, \mathcal{Q}_{\mathsf{b-join}}, \mathcal{Q}_{\mathsf{sign}}]}(1^\nu)$
3. $i = \mathtt{OPEN}(m, \sigma, \mathcal{Y}, \mathcal{S}, St)$
4. If $(\mathtt{VERIFY}(\mathcal{Y}, m, \sigma) = \top) \wedge (i \in U^b) \wedge ((i, m) \notin \mathsf{hist}_{\mathcal{I}}(\mathcal{Q}_{\mathsf{sign}}))$ then return $\top$ else return $\bot$.

Finally we model anonymity. In a deanonymizer-attack the adversary operates in two stages choose and guess. In the choose stage the adversary is allowed to introduce users through $\mathcal{Q}_{\mathsf{p-join}}$ queries, corrupt users through $\mathcal{Q}_{\mathsf{corr}}$ queries, join the system through $\mathcal{Q}_{\mathsf{a-join}}$ queries, as well open signatures through $\mathcal{Q}_{\mathsf{open}}$ queries. The adversary terminates the choose stage by providing a pair of membership certificates/secrets (that were potentially obtained either through corrupt queries or $\mathsf{a-join}$ queries). The adversary obtains a "challenge signature" using one of the two membership certificate/secrets it provided at random, and then proceeds in the guess stage that operates identically to the choose stage with the exception that the adversary is not allowed to open the challenge signature. Note that we don't give to the adversary access to the $\mathcal{Q}_{\mathsf{sign}}$ oracle since the adversary can simulate this oracle easily using the $\mathcal{Q}_{\mathsf{corr}}$ oracle. Note that this attack is similar to a CCA2 attack when an individual group signature is considered an identity concealing ciphertext.

The Deanonymizer Game $G^{\mathcal{A}}_{\mathsf{anon}}$ for security parameter ν (denoted by $G^{\mathcal{A}}_{\mathsf{anon}}(1^\nu)$):

1. $\mathsf{state}_{\mathcal{I}} = \langle St, \mathcal{Y}, \mathcal{S}\rangle \leftarrow \mathtt{SETUP}(1^\nu)$;
2. $\langle aux, m, \mathsf{cert}_1, \mathsf{sec}_1, \mathsf{cert}_2, \mathsf{sec}_2, \rangle \leftarrow \mathcal{A}^{\mathcal{I}[\mathcal{Q}_{\mathsf{pub}}, \mathcal{Q}_{\mathsf{p-join}}, \mathcal{Q}_{\mathsf{a-join}}, \mathcal{Q}_{\mathsf{corr}}, \mathcal{Q}_{\mathsf{open}}]}(\mathsf{choose}, 1^\nu)$
3. if $\neg((\mathsf{cert}_1 \leftrightharpoons_{\mathcal{Y}} \mathsf{sec}_1) \wedge (\mathsf{cert}_2 \leftrightharpoons_{\mathcal{Y}} \mathsf{sec}_2))$ then terminate and return $\bot$;
4. Choose $b \leftarrow_R \{1, 2\}$;
5. $\sigma \leftarrow \mathtt{SIGN}(\mathcal{Y}, \mathsf{cert}_b, \mathsf{sec}_b, m)$;
6. $b^* \leftarrow \mathcal{A}^{\mathcal{I}[\mathcal{Q}_{\mathsf{pub}}, \mathcal{Q}_{\mathsf{p-join}}, \mathcal{Q}_{\mathsf{a-join}}, \mathcal{Q}_{\mathsf{corr}}, \mathcal{Q}^{\neg\{\sigma\}}_{\mathsf{open}}]}(\mathsf{guess}, aux)$;
7. if $b = b^*$ return $\top$ else return $\bot$;

Definition 23 *A group signature scheme is secure if for all PPT $\mathcal{A}$ it holds that (i)* $\mathbf{Prob}[G^{\mathcal{A}}_{\mathsf{in}}(1^\nu) = \top] = \mathsf{negl}(\nu)$ *(ii)* $\mathbf{Prob}[G^{\mathcal{A}}_{\mathsf{out}}(1^\nu) = \top] = \mathsf{negl}(\nu)$ *and (iii)* $2\mathbf{Prob}[G^{\mathcal{A}}_{\mathsf{anon}}(1^\nu) = \top] - 1 = \mathsf{negl}(\nu)$

5.3 Discussion

Bellare et al. [BMW03] concentrated on designing a formal model for group signatures and a generic (inefficient) construction that can be proven secure in this model. We note that a preliminary suggestion for a formal model for the related primitive of identity escrow, was presented by Camenisch and Lysyanskaya [CL01] in the style of ideal model vs. real model. The [BMW03] model compressed the series of security requirements of [ACJT00] into two formal security conditions. While this model was a step towards the realization of a secure model for group signatures it was in fact modeling a weaker primitive, a relaxed group signature, compared to the primitive

realized by [ACJT00]. This fact is noted by the authors themselves. We note that we believe that, methodologically, it is sound to introduce relaxed notions for understanding better the possibility of formal modeling of complex primitives, as was done in [BMW03].

To understand the relaxation, note that, in particular, the syntax of [BMW03] suggested that the tamper-proof key-setup algorithm produces all members' signing keys (which may be a useful model in certain settings). While this may seem to be a minor issue, it is in fact quite crucial, since it prevents any attempt to formalize the exculpability property of [ACJT00] in a natural way without adding additional trusted parties. Indeed, [BMW03] introduce a "fourth" trusted tamper-proof party, a key-issuing authority, which is trusted to generate all keys and distribute them to the GM and the users. Clearly, such was not the approach of [ACJT00] who instead emphasized a group join protocol involving the GM and the user that does not rely on external trusted parties (whose employment relaxes much of the underlying difficulties of group signatures).

The group signature model of Bellare et al. has two security properties called "full-traceability" and "full-anonymity." Naturally, if one wishes to take this model and adapt it somehow to capture the operation of the [ACJT00]-scheme (that assumes no trusted third parties) is immediately in trouble: "full-traceability" (that is the basis of unforgeability) gives to the adversary the GM's secret key. In a scheme where users Join in the sense of Ateniese et al. the adversary would simply create a new membership certificate and thus forge a signature. As a result a scheme in the sense of Ateniese et al. proven secure in the Bellare et al. model can potentially have the adversary forging signatures at will. This point has not escaped the authors themselves and is, in fact, mentioned in Bellare et al. when they motivate the fourth party and discuss partially dynamic groups (when users can join dynamically). They propose simply that the adversary should be denied the key-issuing mechanism which is a clear relaxation when compared to the goals of [ACJT00].

In contrast, in order to cope with the above subtleties of the security modeling, our security against "outsider attack" will prevent adversaries from forging signatures even when they obtain signatures and open them at will. Our "insider attack," on the other hand, will *allow the adversary to corrupt the GM totally and irrevocably* and, in fact, no party is untouchable in this attack. These two attacks imply that there is no other party, but the GM that holds the group private key which is used for membership certificate generation. This is in contrast with the [BMW03] model where in their "partially dynamic group" formulation (i.e., the [ACJT00] setting), they deny to the adversary access to the GM's membership certificate generation mechanism altogether (assuming it is a tamper-proof party); this is a significant relaxation of the model which eases the attack scenario substantially. For further comparison between the two models we refer to the observation #2 at the end of the next section.

The differences above as well as the intuitive goals in [ACJT00] and the extended notion of a scheme with separable authorities motivated our model. We next discuss our scheme and show its correctness and security.

6 Building a Secure Group Signature

The scheme we will prove security will be built based on the state-of-the-art scheme of [ACJT00]. We note that it is impossible to prove security of the [ACJT00]-scheme in our model.

The public-parameters of the group signature are a composite modulus n of ν bits, such that $n = pq$ with $p = 2p' + 1$ and $q = 2q' + 1$ (where p, q, p', q' are primes), as well as a sequence of elements inside $QR(n)$ denoted by a_0, a, g, y and two lengths ℓ, μ, so that $S(2^\ell, 2^\mu) \subseteq \{1, \ldots, p'q'\}$. The membership certificates are of the form $\langle A, e\rangle$ so that $A \in QR(n)$ and e is a prime number in $S(2^\ell, 2^\mu)$. The membership secret is a value x such that $a_0 a^x = A^e$. Given the above structure, the basic functions of the group signature scheme employ two hash functions $\mathcal{G}, \mathcal{H}$ and are implemented as follows:

SETUP: On input a security parameter ν, this probabilistic algorithm first samples a group description for $\langle g, n, p, q, p', q'\rangle \leftarrow \mathrm{desc}_{\mathsf{c}}(1^\nu)$. Then, it selects $x, \hat{x} \leftarrow_R \mathbf{Z}^*_{p'q'}$, $a_0, a, h \leftarrow_R QR(n)$ and publishes the group public key $\mathcal{Y} =_{\mathsf{df}} \langle n, a_0, a, g, h, y = g^x, \hat{y} = g^{\hat{x}}\rangle$ and the secret key is set to $\mathcal{S} =_{\mathsf{df}} \langle p, q, x, \hat{x}\rangle$. The procedure also selects the parameters $\ell, \mu, k \in \mathbb{N}$ and $\epsilon > 1$ as functions of ν so that the following condition is satisfied

$S(2^{\ell}, 2^{\epsilon(\mu+k)+2}) \subseteq \{5, \ldots, \min\{p', q'\} - 1\}$.

JOIN: A protocol between the GM and a user that allows the joint computation of a membership certificate $\langle A, e\rangle$ so that only the user obtains the membership secret x. The specification of the protocol is as follows: $\mathsf{J}_{\mathsf{user}}(1^{\nu}, \mathcal{Y})$ selects $x \leftarrow_R \lfloor n/4 \rfloor$ and writes to the communication tape the value $C = a^x \bmod n$. $\mathsf{J}_{\mathsf{GM}}(1^{\nu}, \mathcal{Y}, \mathcal{S})$ reads C from the communication tape, it selects a prime $e \leftarrow_R S(2^{\ell}, 2^{\mu}) - \{p', q'\}$ and computes $A = (a_0 a)^{1/e} (\mathrm{mod} n)$; finally it writes $\langle i, A, e\rangle$ in the communication tape where i is the next available user tag (a counter is employed) and terminates. $\mathsf{J}_{\mathsf{user}}$ reads $\langle A, e\rangle$ from the communication tape and writes $\langle i, A, e, x\rangle$ in its private output tape.

In the above description, $\mathsf{cert}_i = \langle A, e\rangle$, $\mathsf{sec}_i = x$, $\mathsf{transcript}_i = \langle C, A, e\rangle$. If $\mathsf{transcript}_i = \langle C_t, A_t, e_t\rangle$ and $\mathsf{cert}_i = \langle A_c, e_c\rangle, \mathsf{sec}_i = x_t\rangle$, the relationship $\langle \mathsf{transcript}_i\rangle \leftrightharpoons_{\mathcal{Y}} \langle \mathsf{cert}_i, \mathsf{sec}_i\rangle$ is true iff $A_c = A_t$ and $e_t = e_c$ whereas the relationship $\mathsf{cert}_i \leftrightharpoons_{\mathcal{Y}} \mathsf{sec}_i$ is true iff $A_t^{e_t} = a_0 a^{x_t}$.

SIGN: The signing algorithm is based on a proof of knowledge that is preceded by the values $\langle T_1, T_2, \hat{T}_1, \hat{T}_2, T_3, T_4\rangle$ defined as follows when invoked by the i-th user:

$$r, \hat{r} \leftarrow_R \lfloor n/4 \rfloor \ : \ T_1 = A_i y^r, \ T_2 = g^r, \ \hat{T}_1 = A_i \hat{y}^{\hat{r}}, \ \hat{T}_2 = g^{\hat{r}}, \ T_3 = g^{e_i} h^r$$

$$T_4 = \mathtt{nizk}^{\mathcal{H}}[n, g, y_1, y_2, \langle T_2, T_1\rangle, \langle \hat{T}_2, \hat{T}_1\rangle]$$

The noninteractive proof of knowledge T_4 ensures that the twin ciphertext $T_1, T_2, \hat{T}_1, \hat{T}_2$ is properly formed (see section 4.1). To complete the description of the signature, consider the discrete-log relation set over the free variables r, e, x, s', s'':

$$D := \left[\begin{array}{r|cccccccccccc} & g & g^2 & h & (T_2)^{-1} & y & (T_1)^{-1} & a & a_0 & T_3 & \hat{T}_1 & \hat{T}_2 \\ \hline T_2 = g^r: & r & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ T_3 = g^e h^r: & e & 0 & r & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ T_2^e = g^{s'}: & s' & 0 & 0 & e & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_0 a^x y^{s'} = T_1^e: & 0 & 0 & 0 & 0 & s' & e & x & 1 & 0 & 0 & 0 \\ T_3 = g(g^2)^{s''} h^r: & 1 & s'' & r & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \end{array}\right]$$

The above proof ensures that T_1, T_2 is the ElGamal encryption of a value A that if raised to an odd integer e, it can be split by the prover in the form $a_0 a^x$. Note that D is clearly triangular; the values $\hat{T}_1, \hat{T}_2$ are included in the base so that they will be included inside the hash when we transform the proof in the non-interactive setting; we also require that the whole proof T_4 is also included in the hash. The signature on a message M will be formed by employing the Fiat-Shamir transform over the proof of knowledge in the discrete-log relation set R_D: $\mathtt{sgn}_{\mathcal{G}}^{D}(M)$. It follows from the above that the output $\mathtt{SIGN}(\mathcal{Y}, \mathsf{cert}_i, \mathsf{sec}_i, M)$ has the following form:

$$\langle c, s_1, s_2, s_3, s_4, s_5\rangle \text{ where } c = \mathcal{G}(M, g, g^2, h, (T_2)^{-1}, y, (T_1)^{-1}, a, a_0, T_3, \hat{T}_1, \hat{T}_2, T_4, B_1, B_2, B_3, B_4, B_5)$$

where $B_1, \ldots, B_5, s_1, \ldots, s_5$ are defined based on the structure of the matrix D above and the description of section 2.3.

VERIFY: given a signature $\sigma = \langle c, s_1, s_2, s_3, s_4, s_5\rangle$ the verification algorithm is implemented according to the verification algorithm $\mathtt{ver}_{\mathcal{G}}^{D}$ as described in section 2.3 and verifying T_4.

OPEN: The opening procedure given a signature σ is as follows:

1. Verify σ using the public verification procedure VERIFY.
2. Parse σ to recover the values T_1, T_2.
3. Verify that the noninteractive proof of knowledge T_4 is correct.
4. Compute $A = T_1 (T_2^x)^{-1} \bmod n$.
5. Match A to some user's first component of the membership certificate $\langle A_i, e_i\rangle$ (as available in the database maintained during the JOIN protocols).
6. If either steps 1 or 3 or 5 fail, return $\perp$, else return the user found in step 5.

6.1 Correctness and Security of the Construction

Theorem 24 *The group signature* $\langle \texttt{SETUP}, \texttt{JOIN}, \texttt{SIGN}, \texttt{VERIFY}, \texttt{OPEN} \rangle$ *defined above is correct.*

Proof. Regarding user tagging soundness, it follows immediately since the GM maintains a counter for i that is incremented after each successful join. Regarding join soundness, it follows immediately since by construction the user obtains $\langle i, A, e, x \rangle$ so that $\mathsf{cert}_i = \langle A, e \rangle$ and $\mathsf{sec}_i = x$ that satisfy the relationship $\mathsf{cert}_i \leftrightharpoons \mathsf{sec}_i$, which is $A^e = a_0 a^x (\mathrm{mod} n)$. Regarding signing soundness, observe that a user that holds the membership certificate $\langle A, e \rangle$ and the membership secret x, if she follows the specifications in the construction of the values $T_1, T_2, \hat{T}_1, \hat{T}_2, T_3, T_4$ she knows a witness for the discrete-log relation set D (by setting $s' = er$ and $s'' = \frac{e-1}{2}$). Based on the completeness of the proof of knowledge she can create a valid signature. Finally, regarding the opening soundness, observe that for any valid signature, the OPEN algorithm will recover the value $A = T_1 (T_2)^{-x}$ which is equal to the first component of the membership certificate $\langle A, e \rangle$ that corresponds to the originator of the signature. By matching this to the database St_{trans} that contains all JOIN transcripts of the form $\langle C, A, e \rangle$ the identity of the user (the number i) will be revealed, as long as every user is assigned a unique A component. The probability that the JOIN dialog assigns to a user the same A component is negligible. Indeed, if two users are assigned the same A-value in their certificate, it must be the case that $(a_0 C)^{1/e} = (a_0 C')^{1/e'}$ for a random choice of e, e' from the space $S(2^\ell, 2^\mu) - \{p', q'\}$ and a random choice of C, C'. In this case it must hold that $(a_0 C)^{e'} = (a_0 C')^e$ which is a negligible probability event, since C, C' are uniformly distributed over $QR(n)$ and both $f(a) = a^e (\mathrm{mod} n)$, $f'(a) = a^{e'} (\mathrm{mod} n)$ are bijections over $QR(n)$. □

The proof of security of our scheme is naturally more involved and will be broken down into the following three theorems:

Theorem 25 *(Security against outsider attacks) For any* PPT $\mathcal{A}$ *it holds that* $\mathbf{Prob}[G^{\mathcal{A}}_{\mathsf{out}}(1^\nu) = \top] = \mathsf{negl}(\nu)$ *assuming that the Strong-RSA problem is hard in the random oracle model.*

Proof. Let $\langle n, a \rangle$ be a challenge for the Strong-RSA problem, and let $\mathcal{A}$ be any outsider adversary that has access to the two random oracles employed in the scheme: $\mathcal{G}, \mathcal{H}$.

Below we detail a procedure $\mathcal{P}$ that operates on $\langle n, a \rangle$ and has access to a random oracle $\mathcal{G}$ and to an oracle reprogramming process $\mathcal{R}$ (cf. lemma 6). $\mathcal{P}$ is a simulation of $\mathcal{A}$. Prior to the beginning of the simulation, $\mathcal{P}$ computes two tuples $\mathcal{Y}, \mathcal{S}$ as follows: $\mathcal{Y} := \langle n, a_0, a, g, h, y, \hat{y} \rangle$ where $h \leftarrow_R QR(n), x, \hat{x} \leftarrow_R [\lfloor n/4 \rfloor]$, $r \leftarrow_R [n^2]$, $a_0 = a^r$, $y = g^x, \hat{y} = g^{\hat{x}}$, and $\mathcal{S} := \langle x, \hat{x} \rangle$. Moreover $\mathcal{P}$ initializes a table $T_{\mathcal{H}}$ that will be employed for the simulation of the random oracle $\mathcal{H}$. In the simulation of $\mathcal{A}$ by $\mathcal{P}$, the queries of $\mathcal{A}$ are answered as follows:

- $\mathcal{Q}_{\mathsf{pub}}$ query: $\mathcal{P}$ returns $\mathcal{Y}$. Observe that this answer to the $\mathcal{Q}_{\mathsf{pub}}$ query is indistinguishable from the answer in the actual outsider attack game. This is because both distributions $a_0 \leftarrow a^r$ and $y \leftarrow g^x, \hat{y} \leftarrow g^{\hat{x}}$ are statistically indistinguishable from the uniform over $QR(n)$ (cf. lemma 5).
- $\mathcal{Q}_{\mathsf{p-join}}$ query: note that $\mathcal{P}$ cannot really simulate the JOIN protocol since the factorization of n is not known. Instead, $\mathcal{P}$ selects e_i as in the actual JOIN protocol and a value A_i at random from $QR(n)$; the $\mathcal{P}$ does not select the x_i value. Note that random sampling from $QR(n)$ is still possible even if one doesn't know the factorization of n, and moreover based on lemma 5 the distribution of A_i is indistinguishable from the distribution of A_i in real JOIN protocol executions. $\mathcal{P}$ maintains a user counter and every time a $\mathcal{Q}_{\mathsf{p-join}}$ query is submitted it increases the counter and returns it as output to $\mathcal{A}$. $\mathcal{P}$ stores the values $\langle i, A_i, e_i \rangle$ as part of its internal state.
- $\mathcal{Q}_{\mathsf{sign}}$ query: such a query includes the tuple $\langle i, M \rangle$, where i corresponds to one of the users that were introduced through $\mathcal{Q}_{\mathsf{p-join}}$ queries. $\mathcal{P}$ answers this query to $\mathcal{A}$, by forming $T_1, T_2, \hat{T}_1, \hat{T}_2, T_3, T_4$ exactly as in the description of the actual scheme. Note first, that this is possible since no knowledge of $x_i = \log_a (A_i^{e_i} / a_0)$ is required in the formation of these values; second, the computation of T_4 requires using the random oracle $\mathcal{H}$. $\mathcal{P}$ will select the challenge at random and enter the value in the corresponding

location of the $T_{\mathcal{H}}$ table (or use the existing value if it exists in $T_{\mathcal{H}}$). To complete the signature, the proof of knowledge for the discrete-log relation set must be simulated (since $\mathcal{P}$, lacking knowledge of $x_i = \log_a(A_i^{e_i}/a_0)$ does not have a witness for the relation). This proof of knowledge will be simulated by selecting a challenge c at random from $\{0,1\}^k$ and then using the honest-verifier zero-knowledge simulator from theorem 8 to produce the signature. Finally, $\mathcal{P}$ will need to reprogram the oracle $\mathcal{G}$ so that $c = \mathcal{G}(M, g, g^2, h, (T_2)^{-1}, y, (T_1)^{-1}, a, a_0, T_3, \hat{T}_1, \hat{T}_2, T_4, B_1, B_2, B_3, B_4, B_5)$. This is done by invoking the query $\mathcal{R}$.

- $\mathcal{Q}_{\mathsf{open}}$ query: such queries are answered following the `OPEN` algorithm; note that $\mathcal{P}$ possesses both decryption keys $x, \hat{x}$.
- $\mathcal{H}$ queries are answered by table-lookup using the table $T_{\mathcal{H}}$; if a query x to $\mathcal{H}$ does not exist in $T_{\mathcal{H}}$, $\mathcal{P}$ selects $c \leftarrow \{0,1\}^k$ and answers by c, while it inserts $\langle x, c\rangle$ into $T_{\mathcal{H}}$.
- $\mathcal{G}$ queries are answered by asking $\mathcal{G}$ directly.

Observe now that the above procedure $\mathcal{P}$ satisfies all requirements of lemma 6. In particular with respect to reprogramming queries $\langle t, c\rangle$ that are submitted by $\mathcal{P}$ to $\mathcal{R}$ it holds that c is indeed selected at random from $\{0,1\}^k$ and moreover, the first component t is distributed in such a way so that no single individual t has probability more than $2/2^k$ (a trivial result using the random choice of $B_1, B_2, \ldots, B_5$ and the fact that $k \leq \lfloor \log(n) \rfloor - 2$). It follows that based on lemma 6 we can derive an algorithm $\mathcal{P}'$ that produces (with the proper probability) two distinct proofs of knowledge with the same first move (and of course with the same header $T_1, T_2, \hat{T}_1, \hat{T}_2, T_3, T_4$). Now, using the soundness property of the proof of knowledge we may recover a witness for the proof of knowledge that includes the values r, e, x, that in turn will reveal the tuple $\langle A = T_1/h^r, e, x\rangle$ satisfying the property $A^e = a_0 a^x (\mathrm{mod} n)$. Let us denote this modified $\mathcal{P}'$ procedure that results in the values A, e, x by $\mathcal{P}''$.

The procedure $\mathcal{P}''$, for the given Strong-RSA challenge $\langle n, a\rangle$ provides A, e, x such that $A^e = a^{r+x}$ with $e > 1$ an odd number and $a \in QR(n)$. Observe that the conditions of lemma 4 are satisfied and thus, if $\delta = \gcd(e, r+x) < e$ it follows immediately from lemma 4 that we can either factor n or solve the Strong-RSA challenge $\langle n, a\rangle$. To complete the proof we use the following claim that is based on [CL02]:

Claim With probability at least $1/2$ over the coin tosses of the procedure $\mathcal{P}''$ it holds that $\delta < e$.

Suppose the claim is false. It follows that with probability greater than $1/2$ over the coin tosses of the above simulation it holds that $\delta = e$, i.e., $e \mid r + x$. Let z, x' be integers such that $r = x' + \phi(n)z$. Indeed z is independent of the view of the adversary. On a random choice of $r \leftarrow_R [n^2]$, the value z is a random variable from $\mathbf{Z}_n$; it follows that for more than half choices of z we have that $e \mid x' + \phi(n)z + x$. By the pigeonhole principle there must exist $z_0 \in \mathbf{Z}_n$ such that $e \mid x' + z_0\phi(n) + x$ and $e \mid x' + (z_0+1)\phi(n) + x$; it follows that $e \mid \phi(n) = 4p'q'$; since $e > 4$ by size restrictions (the proof enforces $e \in S(2^\ell, 2^{\epsilon(\mu+k)+2})$) it holds that e has a large common prime factor with $\phi(n)$. Using the same techniques as in the proof of lemma 4, we conclude that knowledge of e allow us to factor n. □

Theorem 26 *(Security against insider attacks) For any* PPT $\mathcal{A}$ *it holds that* $\mathbf{Prob}[G_{\mathsf{in}}^{\mathcal{A}}(1^\nu) = \top] = \mathsf{negl}(\nu)$ *assuming that the Discrete-logarithm problem is hard over the* $QR(n)$ *with known factorization, in the random oracle model.*

Proof. Let $\langle n, p, q, a, A\rangle$ be an instance of the discrete-logarithm problem over $QR(n)$ with known factorization p, q with $p = 2p' + 1$ and $q = 2q' + 1$ (p', q' primes) where ν is the number of bits of n. Let $\mathcal{A}$ be any outsider adversary that has access to the two random oracles employed in the scheme: $\mathcal{G}, \mathcal{H}$.

Below we will detail a procedure $\mathcal{P}$ that operates on $\langle n, p, q, g, A\rangle$ and has access to a random oracle $\mathcal{G}$ and to an oracle reprogramming process $\mathcal{R}$ (cf. lemma 6). $\mathcal{P}$ is a simulation of $\mathcal{A}$. Prior to the beginning of the simulation, $\mathcal{P}$ computes two tuples $\mathcal{Y}, \mathcal{S}$ as follows: $\mathcal{Y} := \langle n, a_0, a, g, h, y, \hat{y}\rangle$ where $g, h \leftarrow_R QR(n), x, \hat{x} \leftarrow_R [p'q']$, $r \leftarrow_R [p'q']$, $a_0 = a^r$, $y = g^x$, $\hat{y} = g^{\hat{x}}$, and $\mathcal{S} := \langle p, q, x, \hat{x}\rangle$. Moreover $\mathcal{P}$ initializes a table $T_{\mathcal{H}}$ that will be employed for the simulation of the random oracle $\mathcal{H}$. In the simulation of $\mathcal{A}$ by $\mathcal{P}$, the queries of $\mathcal{A}$ are answered as follows:

- $\mathcal{Q}_{\mathsf{pub}}$ or $\mathcal{Q}_{\mathsf{key}}$ query: $\mathcal{P}$ returns $\mathcal{Y}$ or $\mathcal{S}$ respectively. Observe that this answer to the $\mathcal{Q}_{\mathsf{pub}}$ query is the identical to the answer in the actual insider attack game.
- $\mathcal{Q}_{\mathsf{b-join}}$ query: $\mathcal{P}$ upon receiving such a query it should initiate a JOIN protocol dialog with the adversary. Indeed, in the i-th $\mathcal{Q}_{\mathsf{b-join}}$ invocation, $\mathcal{P}$ selects $x_i \leftarrow_R [p'q']$ and submits to the adversary the value $C = a^{r_i}A$. Observe that C is uniformly distributed in $QR(n)$ and the adversary will not notice any difference from real executions of the J_{user} protocol's steps. Subsequently the adversary replies by $\langle i, A, e\rangle$ so that $A^e = a_0 C$ and the protocol dialog terminates. $\mathcal{P}$ stores the values $\langle i, r_i, A_i, e_i\rangle$ as part of its internal state.
- $\mathcal{Q}_{\mathsf{sign}}$ query: such a query includes the tuple $\langle i, M\rangle$, where i corresponds to one of the users that were introduced through $\mathcal{Q}_{\mathsf{b-join}}$ queries. $\mathcal{P}$ answers this query to $\mathcal{A}$, by forming $T_1, T_2, \hat{T}_1, \hat{T}_2, T_3, T_4$ exactly as in the description of the actual scheme. Again, this is possible since no knowledge of $x_i = \log_a(A_i^{e_i}/a_0)$ is required in the formation of these values; second, the computation of T_4 requires using the random oracle $\mathcal{H}$. $\mathcal{P}$ will select the challenge at random and enter the value in the corresponding location of the $T_{\mathcal{H}}$ table (or use the existing value if it exists in $T_{\mathcal{H}}$). To complete the signature, the proof of knowledge for the discrete-log relation set must be simulated (since $\mathcal{P}$, lacking knowledge of $x_i = \log_a(A_i^{e_i}/a_0)$ does not have a witness for the relation). This proof of knowledge will be simulated by selecting a challenge c at random from $\{0,1\}^k$ and then using the honest-verifier zero-knowledge simulator from theorem 8 to produce the signature. Finally, $\mathcal{P}$ will need to reprogram the oracle $\mathcal{G}$ so that $c = \mathcal{G}(M, g, g^2, h, (T_2)^{-1}, y, (T_1)^{-1}, a, a_0, T_3, \hat{T}_1, \hat{T}_2, T_4, B_1, B_2, B_3, B_4, B_5)$. This is done by $\mathcal{P}$ by invoking the oracle reprogramming query $\mathcal{R}$.
- $\mathcal{H}$ queries are answered by table-lookup using the table $T_{\mathcal{H}}$; if a query x to $\mathcal{H}$ does not exist in $T_{\mathcal{H}}$, $\mathcal{P}$ selects $c \leftarrow \{0,1\}^k$ and answers by c, while it inserts $\langle x, c\rangle$ into $T_{\mathcal{H}}$.
- $\mathcal{G}$ queries are answered by asking $\mathcal{G}$ directly.

Similarly to the proof of theorem 26, it holds that $\mathcal{P}$ satisfies the requirements of lemma 6, and based on it we can derive with the proper probability an algorithm $\mathcal{P}'$ that produces two distinct proofs of knowledge with the same first move (and of course with the same header $T_1, T_2, \hat{T}_1, \hat{T}_2, T_3, T_4$). Now, using the soundness property of the proof of knowledge we may recover a witness for the proof of knowledge that includes the values r', e', x', that, in turn, will reveal the tuple $\langle A' = T_1/h^{r'}, e', x'\rangle$ satisfying the property $(A')^{e'} = a_0 a^{x'} (\mathrm{mod} n)$. Based on the conditions of the insider game there will be an i_0 such that $A' = A_{i_0}$. Now observe that this implies that $(a_0 a^{x'})^{1/e'} = (a_0 a^{r_{i_0}} A)^{1/e_{i_0}}$ which is equivalent to $a^{(r+x')(e_{i_0}/e') - r - r_{i_0}} = A$, from which we obtain the discrete-logarithm of A. Note that the range restriction $e' \in S(2^\ell, 2^\mu)$ makes sure that $\gcd(e', p'q') = 1$, so that e' is invertible modulo $p'q'$. □

Theorem 27 *(Security against deanonymizer attacks) For any* PPT $\mathcal{A}$ *it holds that* $2\mathbf{Prob}[G^{\mathcal{A}}_{\mathsf{anon}}(1^\nu) = \top] - 1 = \mathsf{negl}(\nu)$ *assuming the* DDH-Compo-KF *in the random oracle model.*

Proof. Let $\mathcal{A}$ be an adversary for the deanonymizer game $G^{\mathcal{A}}_{\mathsf{anon}}$.

Below we describe a modified deanonymizer game G': We form the public-key as $\mathcal{Y} = \langle n, a_0, a, g, h, y, \hat{y}\rangle$ where $a_0, a \leftarrow_R QR(n)$, $x, \hat{x}, z \leftarrow [p'q']$ and $h = g^z, y = g^x, \hat{y} = g^{\hat{x}}$. The secret-key of the system is defined as $\mathcal{S} = \langle p, q, x, \hat{x}\rangle$. The adversary has access to the two random oracles $\mathcal{H}$ and $\mathcal{G}$. Game G' maintains tables $T_{\mathcal{H}}$ and $T_{\mathcal{G}}$ for the simulation of the random oracles and answers the adversary's queries as follows:

- The query $\mathcal{Q}_{\mathsf{pub}}$ is answered by returning the public-key of the system $\mathcal{Y}$.
- The query $\mathcal{Q}_{\mathsf{p-join}}$ is answered by having G' simulate the JOIN dialog in private and create the i-th user's membership secret and certificate (factorization is known).
- The query $\mathcal{Q}_{\mathsf{a-join}}$ initiates a JOIN dialog between the adversary and G' with G' playing the role of the interface executing J_{GM}. The adversary submits some value C and receives from the interface A, e so that $A^e = a_0 C$. Note that G' is capable of answering such queries as it possesses the factorization of n.

- The query $\mathcal{Q}_{\mathsf{corr}}$ is answered by recovering the membership certificate and membership secret of a user $i \in U^p$ and returning it to the adversary.
- In the query $\mathcal{Q}_{\mathsf{open}}$, the adversary submits a signature m, σ to be opened. G' parses σ for the values $T_1, T_2, \hat{T}_1, \hat{T}_2, T_3, T_4$ and verifies the correctness of the proof of language membership T_4. It returns $\perp$ if the proof verification fails. Otherwise G' returns $\hat{T}_1(\hat{T}_2)^x$ to the adversary. Note that the opening is performed in the second ciphertext component. The difference in behavior will only be noticed by the adversary in the event that the adversary produces a valid proof of language membership for a tuple $\langle T_1, T_2, \hat{T}_1, \hat{T}_2 \rangle$ where the ElGamal ciphertexts T_1, T_2 and $\hat{T}_1, \hat{T}_2$ encrypt different plaintexts; this is a negligible probability event based on the simulation soundness of the proof $\mathsf{nizk}^{\mathcal{H}}$.
- Queries to $\mathcal{H}$, $\mathcal{G}$ are answered using the tables $T_{\mathcal{H}}$ and $T_{\mathcal{G}}$ in the usual fashion.

In the end of phase choose the adversary returns $\langle aux, A_1, e_1, x_1, A_2, e_2, x_2, m \rangle$. The interface verifies that $A_1^{e_1} = a_0 a^{x_1}$ and $A_2^{e_2} = a_0 a^{x_2}$, selects $i_b \leftarrow_R \{1, 2\}$ and forms the signature σ as follows:

$$r, \hat{r} \leftarrow_R [p'q'] \ : \ T_1 = A_{i_b} y^r, \ T_2 = g^r, \ \hat{T}_1 = A_{i_b} \hat{y}^{\hat{r}}, \ \hat{T}_2 = g^{\hat{r}}, \ T_3 = g^{e_{i_b}} T_2^z$$

Subsequently, the interface simulates the proof of language membership T_4 (using its honest verifier zero-knowledge simulator and controlling the random oracle $T_{\mathcal{H}}$) and furthermore it simulates the proof for signature $\mathsf{sgn}^D(m)$ (again using the honest verifier zero-knowledge simulator and controlling the random oracle $T_{\mathcal{H}}$).

Subsequent oracle queries in the guess stage of the adversary are simulated as above (with the dictated modification in the opening oracle where the adversary is not allowed to submit the challenge signature).

The above game G' is indistinguishable from the actual game $G^{\mathcal{A}}_{\mathsf{anon}}$; this follows from the statistical indistinguishability of the proofs of knowledge and the simulation soundness. Now we modify game G' in the signing stage to result in the game G'':

$$r, r', \hat{r} \leftarrow_R [p'q'] \ : \ T_1 = A_{i_b} y^r, \ T_2 = g^{r'}, \ \hat{T}_1 = A_{i_b} \hat{y}^{\hat{r}}, \ \hat{T}_2 = g^{\hat{r}}, \ T_3 = g^{e_{i_b}} T_2^z$$

The modification from G' to G'' will only incur a difference of $\mathsf{Adv}^{DDH}_{\mathsf{desc_c}}(\nu)$ in the view of the adversary. This is the case since between the games G' and G'' the quadruple $\langle g, y, T_2, T_1/A_{i_b} \rangle$ behaves as a DDH challenge (valid DDH quadruple for game G' and random quadruple for game G'').

Now we modify again G'' to obtain a new game G''' again by doing two modifications (i) first we return the opening of signature to occur in the first ciphertext, i.e., given a valid signature for opening game G''' will simulate an opening query by decrypting on the first ciphertext. Moreover G''' will modify the signing stage of game G'' as follows:

$$r, r' \hat{r}, \hat{r}' \leftarrow_R [p'q'] \ : \ T_1 = A_{i_b} y^r, \ T_2 = g^{r'}, \ \hat{T}_1 = A_{i_b} \hat{y}^{\hat{r}}, \ \hat{T}_2 = g^{\hat{r}'}, \ T_3 = g^{e_{i_b}} T_2^z$$

again the modification from G'' to G''' will incur a $\mathsf{Adv}^{DDH}_{\mathsf{desc_c}}(\nu) + \epsilon$ difference in the adversary's view. In this case, ϵ accounts for the statistical distance that is due to the switch from the second ciphertext to the first ciphertext in the opening oracle simulation; note that based on the simulation-soundness of the underlying proof of language membership it holds that ϵ is negligible. Regarding the $\mathsf{Adv}^{DDH}_{\mathsf{desc_c}}(\nu)$ difference observe that the quadruple $\langle g, \hat{y}, \hat{T}_2, \hat{T}_1/A_{i_b} \rangle$ behaves as a DDH challenge (valid DDH quadruple for game G'' and random quadruple for game G''').

Finally we modify G''' to the game G'''' by modifying the signing oracle to return:

$$r, r', \hat{r}, \hat{r}', z' \leftarrow_R [p'q'] \ : \ T_1 = A_{i_b} y^r, \ T_2 = g^{r'}, \ \hat{T}_1 = A_{i_b} \hat{y}^{\hat{r}}, \ \hat{T}_2 = g^{\hat{r}'}, \ T_3 = g^{e_{i_b}} T_2^{z'}$$

This modification will again incur only an $\mathsf{Adv}^{DDH}_{\mathsf{desc_c}}(\nu)$ difference in the adversary's view. This is the case since $\langle g, h, T_2, T_3/g^{e_{i_b}} \rangle$ behaves as a DDH challenge (valid DDH quadruple for game G''' and random quadruple for game G'''').

Observe now that the success probability of the adversary in game G'''' is necessarily $1/2$ since all information about the random bit b is lost. It follows easily, that if the deanonymizer attack adversary $\mathcal{A}$ is has non-negligible advantage in the game $G^{\mathcal{A}}_{\mathsf{anon}}$, then this would violate the DDH-Compo-KF. □

Observation #1. The outsider and deanonymizer-attacks are not dependent on any factoring related assumption. This subtle fact eases on the one hand the intractability assumptions (and in fact also the proofs); moreover is crucial in the next section, where we consider an even stronger adversarial setting.

Observation #2. Recaping on the comparison of our setting to the one of [BMW03] we remark that our group signature design of this section can be degenerated to a design that adheres to their syntactic formulation (that employs a trusted party generating the keys) and then proven secure in their security model. In particular the SETUP procedure would be executed by the trusted party (as in their model) that will also simulate several JOIN protocols in order to create a number of membership certificates and secrets and subsequently distribute the membership secrets and certificates to the users as well as hand the opening trapdoor to the GM using secure channels (alternatively users may join with the trusted party as [BMW03] suggest in the "partially dynamic" formulation). After this step is performed, the trusted party does not participate in the protocol or in any attack and the key-issuing trapdoor (the factorization of the modulus) is untouchable by the adversary. The process of signing, verifying and opening remains the same as in our construction. Using similar proof arguments as in our insider and deanonymizer attacks we can prove "full-traceability" (under the discrete-log assumption) and "full-anonymity" (under the DDH-Prime assumption). Note that the modified scheme retains its efficiency, and doesn't depend on the factoring assumption at all. This suggests that *any* group signature scheme where the trapdoor for joining users is different from the opening trapdoor that is proven secure in our security model of section 5 can be modified as above, using the trusted party of [BMW03] and then proven secure in their model.

7 Group Signatures with Authority Separability : Anonymity from Trapdoor Holders

In a group signature with separated authorities we differentiate between the GM, who is responsible for group membership operations and an Opening Authority (OA), who is responsible for the revocation of anonymity (opening a signature). This separation is relevant to practice, since group management should be typically considered an ISP operation whereas revocation of anonymity must be performed by some (possible external) third-party authority (which can even be distributed). This authority separability is natural and is not designed to assure that certain processes are tamper-proof; note that it is a different notion of separability compared to what [CM99] considered.

The syntax of a group signature with authority separability is similar to the group signature syntax as presented in definition 21 with the following modifications:

Definition 28 *A group signature scheme with authority separability is a digital signature scheme comprises the following six procedures; the parties involved are the GM, the opening authority and the users.*

$\texttt{SETUP}_{\mathsf{GM}}$*: On input a security parameter* 1^{ν}*, this probabilistic algorithm outputs the group public key* $\mathcal{Y}_{\mathsf{GM}}$ *(including necessary system parameters) and the secret key* $\mathcal{S}_{\mathsf{GM}}$ *for the GM.* $\texttt{SETUP}_{\mathsf{GM}}$ *also initializes a public-state string* St *with two components* $St_{users} = \emptyset$ *and* $St_{trans} = \epsilon$*.*

$\texttt{SETUP}_{\mathsf{OA}}$*: On input a security parameter* 1^{ν}*, and the public-key* $\mathcal{Y}_{\mathsf{GM}}$*, this probabilistic algorithm generates the public and secret-key of the opening authority denoted by* $\mathcal{Y}_{\mathsf{OA}}$ *and* $\mathcal{S}_{\mathsf{OA}}$*.*

We will denote the concatenation of $\mathcal{Y}_{\mathsf{OA}}$ *and* $\mathcal{Y}_{\mathsf{GM}}$ *by* $\mathcal{Y}$*.*

JOIN*: The* JOIN *protocol is identical to that of definition 21 with the only exception* J_{GM} *requires only the secret key of the GM,* $\mathcal{S}_{\mathsf{GM}}$*.*

SIGN*: identical to definition 21.*

VERIFY*: identical to definition 21.*

OPEN*: the opening algorithm is the same as in definition 21 with the exception that only the opening authority's secret-key $\mathcal{S}_{\mathsf{OA}}$ is required.*

Correctness. Given the above minor syntactic differences, the correctness of a group-signature with separated authorities is defined in the same way as definition 22 by taking into account the above modifications that correspond to the fact that J_{GM} requires only $\mathcal{S}_{\mathsf{GM}}$ and OPEN requires only $\mathcal{S}_{\mathsf{OA}}$.

Security. The security properties of a group-signature with separated authorities must remain the same so that any secure group signature with separated authorities must also be a secure group signature (by collapsing the GM and the OA into a single entity).

Moreover in the separated authority setting the deanonymizer-attack can be made even stronger by *adding* the adversarial capability of corrupting the GM.

Regarding the security modeling, in the queries that can be posed to the interface, the query $\mathcal{Q}_{\mathsf{key}}$ will be substituted with two distinct queries $\mathcal{Q}_{\mathsf{keyGM}}$ and $\mathcal{Q}_{\mathsf{keyOA}}$ with the obvious results. The definition of the three attacks will remain unaltered with the following syntactic modifications:

(i) in an insider-attack the adversary will have at its disposal both the queries $\mathcal{Q}_{\mathsf{keyGM}}$ and $\mathcal{Q}_{\mathsf{keyOA}}$ (i.e., the adversary can corrupt *both* the GM and the OA)

(ii) in the deanonymizer attack, the adversary will be given *additional* access to the $\mathcal{Q}_{\mathsf{keyGM}}$ query — this is in addition to all the queries that are available to the adversary.

The above two modifications are straightforward and thus we will not list the security properties again in this section. The modified games will be denoted by $G^{\mathcal{A}}_{\mathsf{in-sep}}, G^{\mathcal{A}}_{\mathsf{out-sep}}, G^{\mathcal{A}}_{\mathsf{anon-sep}}$.

Definition 29 *A group signature scheme with separated authorities is secure if for all PPT $\mathcal{A}$ it holds that (i)* $\mathbf{Prob}[G^{\mathcal{A}}_{\mathsf{in-sep}}(1^\nu) = \top] = \mathsf{negl}(\nu)$ *(ii)* $\mathbf{Prob}[G^{\mathcal{A}}_{\mathsf{out-sep}}(1^\nu) = \top] = \mathsf{negl}(\nu)$ *and (iii)* $2\mathbf{Prob}[G^{\mathcal{A}}_{\mathsf{anon-sep}}(1^\nu) = \top] - 1 = \mathsf{negl}(\nu)$.

Note that any scheme secure under the above definition is also a secure group signature under definition 23.

Construction. The design of a group signature with separated authorities can be based directly on our construction of section 6 with the following modification: the $\mathtt{SETUP}_{\mathsf{GM}}$ procedure will produce $\mathcal{Y}_{\mathsf{GM}} = \langle n, a_0, a, g, h\rangle$ with $\mathcal{S}_{\mathsf{GM}} = \langle p, q\rangle$, whereas the $\mathtt{SETUP}_{\mathsf{OA}}$ will produce $\mathcal{Y}_{\mathsf{OA}} = \langle y, \hat{y}\rangle$ with $\mathcal{S}_{\mathsf{OA}} = \langle x, \hat{x}\rangle$. In all other respects the scheme will proceed in the same fashion. It is straightforward to split the SETUP procedure to these two authorities, with the condition (as specified in definition 28) that the GM should go first so that the value n is made available; afterwards the OA can select the values $y, \hat{y} \in QR(n)$ with known $\log_g y$ and $\log_g \hat{y}$ and publish the two additional elements to form the combined public key $\mathcal{Y} = \langle n, a_0, a, g, y, \hat{y}\rangle$. To allow the differentiation we specify $\mathcal{Y}_{\mathsf{GM}} = \langle n, a_0, a, g, h\rangle$, $\mathcal{S}_{\mathsf{GM}} = \langle p, q\rangle$, $\mathcal{Y}_{\mathsf{OA}} = \langle y, \hat{y}\rangle$, and $\mathcal{S}_{\mathsf{OA}} = \langle \log_g y, \log_g \hat{y}\rangle$. The design remains unaltered otherwise. In our security proofs we took special care to disassociate the hardness of factoring from anonymity. The following theorem is therefore implied:

Theorem 30 *The group signature with separated authorities presented above is correct and secure; in particular: (i) it is secure against outsider-attacks under the Strong-RSA assumption in the RO model. (ii) it is secure against insider-attacks under the Discrete-Log hardness assumption over $QR(n)$ with known factorization and the RO model. (iii) it is secure against deanonymizer-attacks under* DDH-Compo-KF *in the RO model.*

Proof. The proof is based directly on the proofs of theorems 25, 26 and 27. □

8 Identity Escrow

An identity escrow scheme [KP98] is an identification scheme that allows an entity to prove it belongs to a public group in anonymous fashion while it allows to an Escrow to recover the identity of the originator given any identification transcript. The relationship of this primitive to group signatures is well-known, see e.g., [ACJT00], and in fact the interactive version of any group signature that is based on the Fiat-Shamir transform yields an identity-escrow scheme.

While in this work we concentrated on providing a probably secure group signature, it is possible to transform our exposition to the identity escrow setting by considering the interactive version of our signing algorithm. In this case instead of use of Random Oracle hashes, to compute challenges the verifier is the entity that provides these challenges. We remark that while we use two different hash functions in our generation of a signature, this is not needed to result in two interactions between the prover and the verifier in the interactive setting of an identity escrow scheme. In fact the prover can show the validity of the twin ciphertext and the validity of the proof of knowledge that corresponds to the discrete-log relation set simultaneously, something that will result in a standard 3-move identity escrow scheme. The proofs though have to be made zero-knowledge agaist any verifier (rather than "honest verifier" proofs), and standard transformations are possible.

The security of the resulting identity escrow scheme can be based directly on our security modeling with the following standard constraints that pertain to the interactive setting: (i) Security can be shown against only honest verifiers, i.e., it is assumed that the random challenges submitted by the verifier are randomly selected. This can be enforced in many settings by either employing a beacon that will produce the challenges, or by having the prover and the verifier executing a coin-flipping protocol. (ii) a verifier will not accept concurrent sessions of identification protocols. Requirement (ii) can also be lifted by employing techniques such as those of [GMY03].

The notion of group signatures with separated authorities also transforms naturally to the identity escrow setting, where it yields an identity escrow scheme with separated authorities (one for group management and one for opening — the escrow agent).

References

[AABN02] Michel Abdalla, Jee Hea An, Mihir Bellare, and Chanathip Namprempre. From identification to signatures via the fiat-shamir transform: Minimizing assumptions for security and forward-security. In Lars Knudsen, editor, *Advances in Cryptology – EUROCRYPT ' 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 418–433, Amsterdam, The Netherlands, 2002. Springer.

[ACJT00] Giuseppe Ateniese, Jan Camenisch, Marc Joye, and Gene Tsudik. A practical and provably secure coalition-resistant group signature scheme. In Mihir Bellare, editor, *Advances in Cryptology – CRYPTO ' 2000*, volume 1880 of *Lecture Notes in Computer Science*. International Association for Cryptologic Research, Springer, 2000.

[AT99] G. Ateniese and G. Tsudik. Some open issues and new directions in group signatures. In Matthew 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.

[BMW03] Mihir Bellare, Daniele Micciancio, and Bogdan Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In Eli Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, Warsaw, Poland, 2003. Springer.

[Bon98] Dan Boneh. The decision diffie-hellman problem. In *the Third Algorithmic Number Theory Symposium*, volume 1423 of *Lecture Notes in Computer Science*, pages 48–63. Springer-Verlag, 1998.

[Bou00] Fabrice Boudot. Efficient proofs that a committed number lies in an interval. In Bart Preneel, editor, *Advances in Cryptology – EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 431–444. Springer-Verlag, 2000.

[Cam97] Jan Camenisch. Efficient and generalized group signatures. In Walter 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.

[CDS94] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In Yvo G. Desmedt, editor, *Advances in Cryptology – CRYPTO ' 94*, volume 839 of *Lecture Notes in Computer Science*, pages 174–187. International Association for Cryptologic Research, Springer-Verlag, Berlin Germany, 1994.

[CL01] Jan Camenisch and Anna Lysyanskaya. An identity escrow scheme with appointed verifiers. In Joe 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.

[CL02] Jan Camenisch and Anna Lysyanskaya. A signature scheme with efficient protocols. In *International Conference on Security in Communication Networks – SCN*, volume 2576 of *Lecture Notes in Computer Science*, pages 268–289. Springer Verlag, 2002.

[CM98] Jan Camenisch and Markus Michels. A group signature scheme with improved efficiency. In Kazuo Ohta and Dingyi 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.

[CM99] Jan Camenisch and Markus Michels. Separability and efficiency for generic group signature schemes (extended abstract). In Michael j. Wiener, editor, *19th International Advances in Cryptology Conference – CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*, pages 413–430. Springer, 1999.

[CP94] L. Chen and T. P. Pedersen. New group signature schemes (extended abstract). In Alfredo De 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.

[CS97] Jan Camenisch and Markus Stadler. Efficient group signature schemes for large groups. *Lecture Notes in Computer Science*, 1294:410–424, 1997.

[CS00] Ronald Cramer and Victor Shoup. Signature schemes based on the strong RSA assumption. *ACM Transactions on Information and System Security*, 3(3):161–185, August 2000.

[CvH91] D. Chaum and E. van Heyst. Group signatures. In D. W. Davies, editor, *Advances in Cryptology, Proc. of Eurocrypt '91 (Lecture Notes in Computer Science 547)*, pages 257–265. Springer-Verlag, April 1991. Brighton, U.K.

[DDN91] Danny Dolev, Cynthia Dwork, and Moni Naor. Non-malleable cryptography (extended abstract). In *Proceedings of the Twenty Third Annual ACM Symposium on Theory of Computing*, pages 542–552, New Orleans, Louisiana, 6–8 May 1991.

[DDN00] Danny Dolev, Cynthia Dwork, and Moni Naor. Non-malleable cryptography. *SICOMP*, 30(2):391–437, 2000. A preliminary version appeared in 23rst STOC, 1991.

[FO98] E. Fujisaki and T. Okamoto. A practical and provably secure scheme for publicly verifiable secret sharing and its applications. In Kaisa Nyberg, editor, *Advances in Cryptology – EUROCRYPT '98*, volume 1403 of *Lecture Notes in Computer Science*, pages 32–46. Springer, 1998.

[FP01] Pierre-Alain Fouque and David Pointcheval. Threshold cryptosystems secure against chosen-ciphertext attacks. In *ASIACRYPT: Advances in Cryptology – ASIACRYPT: International Conference on the Theory and Application of Cryptology*, volume 2248 of *Lecture Notes in Computer Science*, pages 351–368. Springer Verlag, 2001.

[FS86] Amos Fiat and Adi Shamir. How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In *Proceedings of CRYPTO'86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer Verlag, 1986.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer Security*, 28:270–299, 1984.

[GMR84] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. A "paradoxical" solution to the signature problem (extended abstract). In *25th Annual Symposium on Foundations of Computer Science*, pages 441–448, Singer Island, Florida, 24–26 October 1984. IEEE.

[GMY03] Juan Garay, Philip D. MacKenzie, and Ke Yang. Strengthening zero-knowledge protocols using signatures. In Eli Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 177–194, Warsaw, Poland, 2003. Springer.

[Gol97] Oded Goldreich. On the foundations of modern cryptography. In *Proc. 17th Annual International Cryptology Conference – CRYPTO '97*, pages 46–74, 1997.

[KP98] Joe Kilian and Erez Petrank. Identity escrow. In Hugo 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.

[KTY04] Aggelos Kiayias, Yiannis Tsiounis, and Moti Yung. Traceable signatures. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT ' 2004*, Lecture Notes in Computer Science, Interlaken, Switzerland, 2004. Springer.

[KY03] Aggelos Kiayias and Moti Yung. Extracting group signatures from traitor tracing schemes. In Eli Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 630–648, Warsaw, Poland, 2003. Springer.

[McC88] Kevin S. McCurley. A key distribution system equivalent to factoring. *Journal of Cryptology: the journal of the International Association for Cryptologic Research*, 1(2):95–105, 1988.

[NY90] M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In Baruch Awerbuch, editor, *Proceedings of the 22nd Annual ACM Symposium on the Theory of Computing*, pages 427–437, Baltimore, MY, May 1990. ACM Press.

[PS00] David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, March 2000.

[RS92] Charles Rackoff and Daniel R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In Joan Feigenbaum, editor, *Advances in Cryptology – CRYPTO ' 91*, volume 576 of *Lecture Notes in Computer Science*, pages 433–444. International Association for Cryptologic Research, Springer-Verlag, Berlin Germany, 1992.

[Sah99] Amit Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In IEEE, editor, *40th Annual Symposium on Foundations of Computer Science: October 17–19, 1999, New York City, New York,*, pages 543–553. IEEE Computer Society Press, 1999.

[Sha83] Adi Shamir. On the generation of cryptographically strong pseudorandom sequences. *ACM Transactions on Computer Systems*, 1(1):38–44, February 1983.

[TY98] Y. Tsiounis and M. Yung. On the security of ElGamal based encryption. In *Proc. 1st International Public Key Cryptography Conference*, Lecture Notes in Computer Science, pages 117–134, 1998.