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

RSA Laboratories,
Bedford, MA, USA, and
Columbia University
New York, NY, USA
moti@cs.columbia.edu

Abstract

To date, a group signature construction which is efficient, scalable, allows dynamic adversarial joins, and proven secure in a formal model has not been suggested. In this work we give the first such construction in the random oracle model. The demonstration of an efficient construction proven secure in a formal model that captures all intuitive security properties of a certain primitive is a basic goal in cryptographic design. To this end we adapt a formal model for group signatures capturing all the basic requirements that have been identified as desirable in the area and we construct an efficient scheme and prove its security. Our construction is based on the Strong-RSA assumption (as in the work of Ateniese et al.). In our system, due to the requirements of provable security in a formal model, we give novel constructions as well as innovative extensions of the underlying mathematical requirements and properties. Our task, in fact, requires the investigation of some basic number-theoretic techniques for arguing security over the group of quadratic residues modulo a composite when its factorization is known. Along the way we discover that in the basic construction, anonymity does not depend on factoring-based assumptions, which, in turn, allows the natural separation of user join management and anonymity revocation authorities. Anonymity can, in turn, be shown even against an adversary controlling the join manager.

*Research partly supported by NSF Career Award CNS-0447808.

Contents

1 Introduction

The notion of *group signature* is a central anonymity primitive that allows users to have anonymous non-repudiable credentials. The primitive was introduced by Chaum and Van Heyst [13] and it 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 and scalable group signature has been a research target for many years since its introduction with quite a slow progress, see e.g., [14, 12, 10, 11, 8, 27, 3, 2, 9, 24, 7]. In many of the early works the signature size was related to the group size. The first construction that appeared to provide sufficient heuristic security and efficiency properties and where user joins are performed by a manager that is not trusted to know their keys, was the scalable scheme of Ateniese, Camenisch, Joye and Tsudik [2]. It provided constant signature size and resistance to attacks by coalitions of users. This scheme was based on a novel use of the DDH assumption combined with the Strong-RSA assumption over groups of intractable order.

Recently, Bellare, Micciancio and Warinschi [4], noticing that the work of [2] claims a collection of individual intuitive security properties, advocated the need for a formal model for arguing the security of group signature. This basic observation is in line with the development of solid security notions in modern cryptography, where a formal model that captures the properties of a primitive is defined and a scheme implementation is formally proven (in some model) to satisfy the security definitions. 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, cf. [20]. The relaxation in the model of [4] amounts to replacing the dynamic adversarial join protocols of [2] where users get individual keys with a trusted party that generates and distributes keys securely (relevant in some settings but perhaps unlikely in others).

The above state of affairs ([2, 4]) indicates that there exists a gap in the long progression 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 on the one hand, and efficient more ad-hoc constructions with intuitive claims on the other. 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 (that convincingly captures all desired intuitive security properties). To summarize the above, it is apparent that the following question remained open by earlier works:

> Design an **efficient** group signature with dynamic joins (and no trusted parties) which is **provably secure** within a formal model.

One of our contributions is solving the above open question by, both, adapting a new model for group signatures (based on the model of traceable signatures of [23]), which follows the paradigm of [22] for the security of signature schemes, as well as providing an efficient provably secure construction (in the sense of the scheme of [2]), and a comprehensive security proof.

These contributions reveal many subtleties regarding the exact construction parameters, and in particular issues regarding what intractability assumptions are actually necessary for achieving the security properties. For example, the anonymity property in our treatment is totally disassociated from any factoring related assumption. We note that, methodologically, in order to reveal such issues, a complete proof is needed following a concrete model. This has not been done in the realm of (efficient) group signatures and concrete proof and model are unique to our work. (We note that even though we try to build our constructions on prior assumptions and systems as much as possible, we need to modify them extensively as required by the constraints imposed by following formal model and arguments).

Our investigation also reveals delicate issues regarding the proper formal modeling of the group signature primitive with regards to the work of [4]. For example, the need of formalizing security against attacks by any internal or external entity that is active in the scheme (i.e., no trusted parties). Lack of such treatment, while proper for the non-dynamic setting of [4], is insufficient for proving the security of schemes that follow the line of work of [2] (i.e., where there are no trusted key generators).

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

1. MODELING. To model schemes like the scheme of [2] with dynamic (yet sequential) joins and no trusted parties we adapt the model of [23] which is the first formal model in the area of group signing without added trusted parties. In particular, our model has the three types of attacks that involve the GM and the users similarly to [23]. We extend the model to allow adversarial opening of signatures (see the next paragraph). 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. We note that this follows the basic approach of [22] for modeling security of digital signatures, yet in the complicated system with various parties, a few attacks which can co-exist are possible, and needed to be described as part of the system security.

2. ADVERSARIAL OPENING IN EFFICIENT SCHEMES. As mentioned above, our formal model extends the security requirements given by the list of security properties of [2] by allowing the adversary to request that the system opens signatures of its choice. In the work of [2], opening of signatures was implicitly assumed to be an internal operation of the GM. We note that such stronger adversarial capability was put forth for the first time in the formal model of [4]. For achieving an efficient scheme with adversarial opening we needed to develop novel cryptographic constructs. (Note that adversarial opening can also be applied to strengthen the notion of traceable signatures).

3. STRONGER ANONYMITY PROPERTY. In the scheme of [2] anonymity is claimed 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 and allow them to join the group (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 who is the "Join Manager" still needs to know the factorization. The opening authority, on the other hand, must know the ElGamal key. This split of functions (separation of authorities) is not a relaxation of group signatures but rather a constraining of the primitive. One should observe that the introduction of such additional functionalities in a primitive potentially leads to new attacks and to a change in the security model. Indeed in the separated authorities setting, we must allow the anonymity adversary to *corrupt the GM as well.*

4. NUMBER-THEORETIC RESULTS AND CRYPTOGRAPHIC PRIMITIVES. The last 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 appears to be hard even if the adversary knows the factorization. In particular, we prove that any adversary that knows the factorization p, q and solves the DDH problem over the 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 (chosen ciphertext attack) secure 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, cf. [29, 16, 19] 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, cf. [18, 1]).

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 motivated by [2] (and originally we tried to rely on it as much as possible), our scheme, nevertheless, possesses many subtle and important differences. These differences enable the proof of security of our scheme whereas the scheme presented by [2] claims security in heuristic arguments that are not complete and, in particular, cannot be proven secure in our model: There are many reasons for this, e.g., the scheme of [2] lacks an appropriate cca2 secure identity embedding mechanism. Moreover, our efficient construction can support formally (if so desired), the separation of group management and opening capability – something not apparent in the prior scheme of [2]. Finally, we note that a syntactically degenerated version of our construction (that retains its efficiency) can be proven secure in the model of [4] (and is, in fact, a non-dynamic group signature scheme of the type they have suggested).

An interesting technical result with respect to anonymity compared to previous work is highlighted in our investigation. Anonymity was argued in the work of [2] 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 trapdoor. Thus, we disassociate anonymity from the factoring problem. Taking this result independently it also implies the separability between the opening authority and the group manager. In addition, we note that many other technical and subtle details are different in our provable scheme from prior designs.

An extended abstract of the present paper appeared in [26].

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 which is multiple of two strong primes, 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 Group Manager (GM) and the Opening Authority (OA)).

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 that is parameterized by ν we will write $\mathrm{dist}_{\mathcal{A}}(\mathcal{D}_1, \mathcal{D}_2)$ to denote the computational 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 ν. Similarly, we will write $\mathrm{dist}(\mathcal{D}_1, \mathcal{D}_2)$ to denote the maximum distance among all PPT predicates $\mathcal{A}$. Note that the statistical distance of the distributions $\mathcal{D}_1, \mathcal{D}_2$, namely $\frac{1}{2}\sum_x |\mathbf{Prob}_{\mathcal{D}_1}[x] - \mathbf{Prob}_{\mathcal{D}_2}[x]|$ might be much larger than the computational distance.

If n is any number, we will denote by $[n]$ the set $\{1, \ldots, \lfloor n \rfloor\}$. 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 $\mathbb{Z}_n$. A function $f : \mathbb{N} \rightarrow \mathbb{R}$ will be called negligible if for all $c > 0$ there exists a ν_c such that for all $\nu \geq \nu_c$, $f(\nu) < \nu^{-c}$. In this case we will write $f(\nu) = \mathsf{negl}(\nu)$. 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). In general we will use the letter ν to denote the security parameter (i.e., this value will be polynomially related to the sizes of all quantities involved). 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 $\mathbb{Z}_n^*$ 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 $u \in \mathbb{Z}_n^*$ 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, cf. [15], 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 [15]).

The second assumption that we employ is the Decisional Diffie-Hellman Assumption (see e.g., [6] for a survey). We state it below for a general group G and later on in definition 5 we will specialize this definition to two specific groups.

Decisional Diffie-Hellman Given a description of a cyclic (sub)group G that includes 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 2 Range-bounded Discrete-Logarithm with known factorization. *Given two values a, b that belong to the set of quadratic residues modulo n with known factorization $n = pq$, so that there is an $x \in \Lambda \subseteq [p'q'] : a^x = b$, p, q are safe primes, $\#\Lambda = \Theta(n^\epsilon)$ for a given constant $\epsilon > 0$, it is infeasible to find in time polynomial in ν the integer x so that $a^x = b(\mathrm{mod} n)$.*

3 DDH over $QR(n)$ with known Factorization

Our constructions will require the investigation of the number-theoretic results presented in this section that albeit entirely elementary they have not being observed in the literature to the best of our knowledge. In particular we will show that DDH over $QR(n)$ does not depend on the hardness of factoring.

Let n be a composite, $n = pq$ with $p = 2p' + 1$ and $q = 2q' + 1$ (p, q, p', q' primes). Recall that elements of $\mathbb{Z}_n^*$ are in a 1-1 correspondence with the set $\mathbb{Z}_p^* \times \mathbb{Z}_q^*$. Indeed, given $\langle b, c\rangle \in \mathbb{Z}_p^* \times \mathbb{Z}_q^*$, consider the system of equations $x \equiv b(\mathrm{mod} p)$ 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 $\mathbb{Z}_n^*$. Alternatively for any $a \in \mathbb{Z}_n^*$ we can find the corresponding pair $\langle b, c\rangle$ in $\mathbb{Z}_p^* \times \mathbb{Z}_q^*$ 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 $\mathbb{Z}_p^* \times \mathbb{Z}_q^*$ to $\mathbb{Z}_n^*$ is called the Chinese remaindering mapping. Observe that ρ preserves quadratic residuosity: $\rho(QR(p) \times QR(q)) = QR(n)$.

The following two lemmas will be useful in the sequel. They show (1) how the Chinese remaindering mapping behaves when given inputs expressed as powers inside the two groups $QR(p)$ and $QR(q)$, and (2) how discrete-logarithms over $QR(n)$ can be decomposed.

Lemma 3 *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 $q' \in \mathbb{Z}_{p'}^*$ and as a result q' is an invertible element of $\mathbb{Z}_{p'}^*$. 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 $p'(\bmod q') \in \mathbb{Z}_{q'}^*$ 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 $\mathbb{Z}_n^*$ 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 argument 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 $\mathbb{Z}_n^*$ it follows that $\beta = \alpha^{q'x_1+p'x_2}(\bmod n)$ is the solution of the system. □

Lemma 4 *Fix a generator α of $QR(n)$ and an integer $t \in \mathbb{N}$. The mapping $\tau_\alpha : \mathbb{Z}_{p'} \times \mathbb{Z}_{q'} \to QR(n)$, 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 $\langle x_1, x_2\rangle, \langle x_1', x_2'\rangle \in \mathbb{Z}_{p'} \times \mathbb{Z}_{q'}$ 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 $\mathbb{Z}_{p'}, \mathbb{Z}_{q'}$ 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 $y_2 = p^*x \mod q'$. Let π_1, π_2 be integers 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 $2^{\nu-1} \le \tilde{d}_G.\mathsf{ord} < 2^\nu$, 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:

- $\mathrm{desc}_{\mathsf{p}}$: 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 non-trivial 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 $\mathrm{desc}_{\mathsf{p}}$ 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 $\tilde{d}.\mathsf{gen} = g$.

- $\mathrm{desc}_{\mathsf{c}}$: 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 $q = 2q' + 1$. The descriptor $\mathrm{desc}_{\mathsf{c}}$ 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 $\mathrm{desc}_{\mathsf{c}}$ that we will employ is the following: execute $\mathrm{desc}_{\mathsf{p}}$ 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 $\tilde{d}$. Note that in the (unlikely) event $p = q$ the procedure is repeated.

Definition 5 *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:*

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

where $\mathcal{D}^{\mathrm{desc}}_\nu$ 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 $\mathcal{R}^{\mathrm{desc}}_\nu$ 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: $\mathsf{Adv}^{DDH}_{\mathrm{desc}}(\nu) = \max_{PPT\,\mathcal{A}} \mathsf{Adv}^{DDH}_{\mathrm{desc},\mathcal{A}}(\nu)$.

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 6 *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 $\tilde{d}_2 = \langle g_2, q, q' \rangle$. Consider a mapping ρ^* defined as follows:*

$$\rho^*(\langle \tilde{d}_1, A_1, B_1, C_1 \rangle, \langle \tilde{d}_2, A_2, B_2, C_2 \rangle)$$

$$=_{\mathsf{df}} \begin{cases} \langle \tilde{d}, \rho(A_1, A_2), \rho(B_1, B_2), \rho((C_1)^{q'}, (C_2)^{p'}) \rangle \\ \qquad\qquad\qquad \perp \end{cases}$$

so that the $\perp$ output is given if and only if $\tilde{d}_1.\mathsf{ord} = \tilde{d}_2.\mathsf{ord}$. Then it holds, that ρ^* satisfies the properties (i) $\mathrm{dist}(\rho^*(\mathcal{D}^{\mathrm{desc}_{\mathsf{p}}}_{\nu/2}, \mathcal{D}^{\mathrm{desc}_{\mathsf{p}}}_{\nu/2}), \mathcal{D}^{\mathrm{desc}_{\mathsf{c}}}_{\nu}) \leq \frac{3 \log 2 \cdot \nu}{2^{\nu/2}}$ and (ii) $\mathrm{dist}(\rho^*(\mathcal{R}^{\mathrm{desc}_{\mathsf{p}}}_{\nu/2}, \mathcal{R}^{\mathrm{desc}_{\mathsf{p}}}_{\nu/2}), \mathcal{R}^{\mathrm{desc}_{\mathsf{c}}}_{\nu}) \leq \frac{3 \log 2 \cdot \nu}{2^{\nu/2}}$.

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 3 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 4 and standard results on the distribution of primes we can deduce easily that $\mathrm{dist}(\rho^*(\mathcal{D}^{\mathrm{desc}_{\mathsf{p}}}_{\nu/2}, \mathcal{D}^{\mathrm{desc}_{\mathsf{p}}}_{\nu/2}), \mathcal{D}^{\mathrm{desc}_{\mathsf{c}}}_{\nu}) \leq \frac{3 \log 2 \cdot \nu}{2^{\nu/2}}$, i.e., the two distributions are statistically indistinguishable. We conclude that the distribution defined by ρ^* when applied to two distributions of DDH triples from $\mathcal{D}^{\mathrm{desc}_{\mathsf{p}}}_{\nu/2}$ over the respective groups is statistically close to the distribution $\mathcal{D}^{\mathrm{desc}_{\mathsf{c}}}_{\nu}$. 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 3 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, using lemma 4, $\mathrm{dist}(\rho^*(\mathcal{R}^{\mathrm{desc}_{\mathsf{p}}}_{\nu/2}, \mathcal{R}^{\mathrm{desc}_{\mathsf{p}}}_{\nu/2}), \mathcal{R}^{\mathrm{desc}_{\mathsf{c}}}_{\nu}) \leq \frac{3 \log 2 \cdot \nu}{2^{\nu/2}}$, i.e., the two distributions are statistically indistinguishable. □

The lemma is used for the proof of the theorem below:

Theorem 7 $\mathsf{Adv}^{DDH}_{\mathrm{desc}_{\mathsf{c}}}(\nu) \leq 2\mathsf{Adv}^{DDH}_{\mathrm{desc}_{\mathsf{p}}}(\nu/2) + (6 \log 2 \cdot \nu)/2^{\nu/2}$.

Proof. Let $\mathcal{A}$ be any DDH-distinguisher for $\mathrm{desc}_{\mathsf{c}}$. Using property (i) of lemma 6, we have that

$$\mathrm{dist}_{\mathcal{A}}(\mathcal{D}^{\mathrm{desc}_{\mathsf{c}}}_{\nu}, \rho^*(\mathcal{D}^{\mathrm{desc}_{\mathsf{p}}}_{\nu/2}, \mathcal{D}^{\mathrm{desc}_{\mathsf{p}}}_{\nu/2})) \leq \frac{3 \log 2 \cdot \nu}{2^{\nu/2}}$$

and given that

$$\mathrm{dist}_{\mathcal{A}}(\rho^*(\mathcal{D}^{\mathrm{desc}_{\mathsf{p}}}_{\nu/2}, \mathcal{D}^{\mathrm{desc}_{\mathsf{p}}}_{\nu/2}), \rho^*(\mathcal{R}^{\mathrm{desc}_{\mathsf{p}}}_{\nu/2}, \mathcal{D}^{\mathrm{desc}_{\mathsf{p}}}_{\nu/2})) \leq$$

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

we obtain

$$\text{(Fact 1)} \qquad \mathrm{dist}_{\mathcal{A}}(\mathcal{D}_{\nu}^{\mathrm{desc}_{\mathsf{c}}}, \rho^{*}(\mathcal{R}_{\nu/2}^{\mathrm{desc}_{\mathsf{p}}}, \mathcal{D}_{\nu/2}^{\mathrm{desc}_{\mathsf{p}}})) \leq$$

$$\leq \mathsf{Adv}_{\mathrm{desc}_{\mathsf{p}}}^{DDH}(\nu/2) + \frac{3\log 2 \cdot \nu}{2^{\nu/2}}$$

Now using property (ii) of lemma 6 we have that

$$\mathrm{dist}_{\mathcal{A}}(\mathcal{R}_{\nu}^{\mathrm{desc}_{\mathsf{c}}}, \rho^{*}(\mathcal{R}_{\nu/2}^{\mathrm{desc}_{\mathsf{p}}}, \mathcal{R}_{\nu/2}^{\mathrm{desc}_{\mathsf{p}}})) \leq \frac{3\log 2 \cdot \nu}{2^{\nu/2}}$$

and given that

$$\mathrm{dist}_{\mathcal{A}}(\rho^{*}(\mathcal{R}_{\nu/2}^{\mathrm{desc}_{\mathsf{p}}}, \mathcal{D}_{\nu/2}^{\mathrm{desc}_{\mathsf{p}}}), \rho^{*}(\mathcal{R}_{\nu/2}^{\mathrm{desc}_{\mathsf{p}}}, \mathcal{R}_{\nu/2}^{\mathrm{desc}_{\mathsf{p}}})) \leq$$

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

we obtain

$$\text{(Fact 2)} \qquad \mathrm{dist}_{\mathcal{A}}(\rho^{*}(\mathcal{R}_{\nu/2}^{\mathrm{desc}_{\mathsf{p}}}, \mathcal{D}_{\nu/2}^{\mathrm{desc}_{\mathsf{p}}}), \mathcal{R}_{\nu}^{\mathrm{desc}_{\mathsf{c}}}) =$$

$$= \mathsf{Adv}_{\mathrm{desc}_{\mathsf{p}}}^{DDH}(\nu/2) + \frac{3\log 2 \cdot \nu}{2^{\nu/2}}$$

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

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

$$\leq 2 \cdot \mathsf{Adv}_{\mathrm{desc}_{\mathsf{p}}}^{DDH}(\nu/2) + \frac{6\log 2 \cdot \nu}{2^{\nu/2}}$$

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

We proceed to state explicitly the two variants of the DDH assumption:

Definition 8 *The following are two Decisional Diffie Hellman Assumptions:*
• *The DDH assumption over quadratic residues modulo a safe prime (*DDH-Prime*) asserts that:* $\mathsf{Adv}_{\mathrm{desc}_{\mathsf{p}}}^{DDH}(\nu) = \mathsf{negl}(\nu)$.
• *The DDH assumption over quadratic residues modulo a safe composite with known Factorization (*DDH-Comp-KF*) asserts that:* $\mathsf{Adv}_{\mathrm{desc}_{\mathsf{c}}}^{DDH}(\nu) = \mathsf{negl}(\nu)$.

We conclude the section with the following theorem (where $\Longrightarrow$ stands for logical implication):

Theorem 9 DDH-Prime $\Longrightarrow$ DDH-Comp-KF.

Proof. An immediate corollary of theorem 7 and the easy fact that if f_1, f_2 are negligible functions in ν then $2 \cdot f_1(\nu) + f_2(\nu)$ is also a negligible function.

□

4 PK-Encryption over $QR(n)$ with split n

Our constructions will require a special identity embedding mechanism that is $\mathsf{cca2}$ 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 symbol $\perp$. The correctness of a public-key encryption scheme 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, cf. [21, 29, 31, 17]; below we will be interested in the so-called CPA and $\mathsf{cca2}$ security in the indistinguishability sense. For completeness we define these notions below:

A $\mathsf{cca2}$ 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 $\nu$ (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 "$\mathtt{if}\ x \neq \psi^*\ \mathtt{then\ return\ Dec}(\mathsf{s}, x)\ \mathtt{else\ return}\ \perp$";
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 10 *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* $2\mathbf{Prob}[G^{\mathcal{A}}_{\mathsf{X}}(1^\nu) = \top] - 1 = \mathsf{negl}(\nu)$.

Now 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 $\langle g^r \bmod n, h^r M \bmod n \rangle$.
- 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 11 *The cryptosystem* $\langle \mathtt{Gen}_{qr}, \mathtt{Enc}_{qr}, \mathtt{Dec}_{qr} \rangle$ *described above satisfies* CPA*-security under the assumption* $\mathsf{DDH\text{-}Compo\text{-}KF}$*, and thus under the assumption* $\mathsf{DDH\text{-}Prime}$ *(theorem 9).*

Proof. The proof of CPA-security for the ElGamal variant we define is very similar to the proof of CPA-security for regular ElGamal encryption as formulated by [33], and we omit it (in fact the simplification of this proof was one reason for introducing DDH-Compo-KF in the first place). □

We remark that ElGamal variants over composite order groups have been considered before, e.g., [28]; 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. To achieve this goal we will employ the double encryption approach. Double encryption has been employed as a tool to obtain chosen-ciphertext security originally in [29]. Based on double encryption the so called "twin-conversion" has been formalized in [19]: it transforms a CPA-secure cryptosystem into a cca2-cryptosystem by employing proofs of language membership that are "simulation-sound" , cf. [32].

In the remaining of the section we will present a transformation of the ElGamal variant we presented above in the general spirit of the twin-transform. For various technical reasons we cannot employ the transform in a generic fashion and below we will provide a direct stand-alone argumentation for the security of the construction. We start by presenting the cryptosystem:

- $\texttt{Gen}'_{qr}$ 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 $\texttt{Enc}'_{qr}$: in order to encrypt a message m, we form the two ciphertexts $\langle g^{r_1}, y_1^{r_1} m\rangle$ and $\langle g^{r_2}, y_2^{r_2} m\rangle$ 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 $\mathcal{L}_{qr}$ we will use a proof of language membership defined below in definition 12.

  It follows that the output of $\texttt{Enc}'_{qr}$ 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 $\mathcal{L}_{qr}$.
- The decryption $\texttt{Dec}'_{qr}$, operates as follows: first it checks the proof π, and if the check fails it returns $\perp$, otherwise it applies x_1 to G_1 and returns $(Y_1 \cdot G_1^{-x_1})^2 \bmod n$. Note that the decryption does not return M but rather $M^2 \bmod n$. We will explain the reason for this choice later on in the construction of the group signature.

Definition 12 The proof of language membership for $\mathcal{L}_{qr}$. *Suppose that the values $r_1, r_2 \in [\lfloor n/4 \rfloor]$. The interaction between the prover and the verifier is as follows: the prover selects $t_1, t_2 \in [-2^{k+l}\lfloor n/4 \rfloor, \ldots, 2^{k+l}\lfloor n/4 \rfloor]$ and transmits to the verifier the values $B_1 = g^{t_1}, B_2 = g^{t_2}, B_3 = y_1^{t_1}/y_2^{t_2}$. The verifier selects a challenge $c \in \{0,1\}^k$, and subsequently the prover computes $s_i = t_i - c \cdot r_i$ for $i = 1, 2$ and transmits to the verifier the values s_1, s_2. The verification check is the following: $g^{s_1}(G_1)^c =_? B_1$, $g^{s_2}(G_2)^c =_? B_2$ and $(y_1^{s_1}/y_2^{s_2})(Y_1/Y_2)^c =_? B_3$. In order to make the proof non-interactive using a hash function $\mathcal{H} : \{0,1\}^* \rightarrow \{0,1\}^k$ we perform the following: the non-interactive proof π in the description of* $\texttt{Enc}'_{qr}$ *will have the form*

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

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:*

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

The set of proofs π *constructed as above for a given ciphertext will be denoted by*

$$\mathtt{nizk}^{\mathcal{H}}[n, g, y_1, y_2, \langle G_1, Y_1\rangle, \langle G_2, Y_2\rangle]$$

Given the description of constructing the non-interactive proof of knowledge it is easy to verify that valid encryptions of messages m will never result in the decryption function $\mathtt{Dec}'_{qr}$ returning $\perp$. The introduction of π though along with each ciphertext introduces a possible security concern since the random coins used for encryption are employed in the construction of π. To settle this issue we willfirst present the following technical lemma:

Lemma 13 *Consider a fixed* $x \in [L, R]$ *with* $m = R-L$ *and the random variables* $t \in_R [-2^{k+l}m, 2^{k+l}m]$, $c \in_R \{0,1\}^k$. *The statistical distance of the random variable* $\hat{s} = t-c(x-L)$ *from the random variable* $s \in_R [-2^{k+l}m, 2^{k+l}m]$ *is less than* 2^{-l}.

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

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

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

Now consider the following algorithm $\mathcal{S}$: given $n, g, y_1, y_2, \langle G_1, Y_1\rangle, \langle G_2, Y_2\rangle$ and parameters k, l it selects a random $c \in \{0,1\}^k$ and random $s_1, s_2 \in [-2^{k+l}\lfloor n/4\rfloor, 2^{k+l}\lfloor n/4\rfloor]$ and then produces the values:

$$B_1 = g^{s_1}G_1^c, B_2 = g^{s_2}G_2^c, B_3 = (y_1^{s_1}/y_2^{s_2})(Y_1/Y_2)^c, c, s_1, s_2$$

In the following proposition we will establish that in the random oracle model an adversary is incapable of taking any significant advantage of the extra information provided by the attachment of π to a ciphertext.

Proposition 14 *Consider the following two experiments executed with any probabilistic polynomial-time adversary $\mathcal{A}^{\mathcal{H}}$ that has access to a random oracle $\mathcal{H}$ and operates in two stages: in the first stage it receives a public-key of the encryption scheme $\langle \mathtt{Gen}'_{qr}, \mathtt{Enc}'_{qr}, \mathtt{Dec}'_{qr}\rangle$ and it outputs two plaintexts m_0, m_1; in the second stage it receives an encryption of m_b under $\mathtt{Enc}'_{qr}$, where b is a random bit and produces a single bit output. The experiments are defined as follows:*

- *(Experiment 1.) Simulate $\mathcal{A}^{\mathcal{H}}$ so that queries to $\mathcal{H}$ are answered on-the-fly by generating the table of $\mathcal{H}$. When $\mathcal{A}$ outputs m_0, m_1, encrypt m_b using $\mathtt{Enc}'_{qr}$ and finish the simulation of $\mathcal{A}^{\mathcal{H}}$ by returning the output of $\mathcal{A}$.*
- *(Experiment 2.) Proceed as in experiment 1, with the following modification: the proof π in the encryption of $\mathtt{Enc}'_{qr}$ is substituted by a string $\pi = (c, s_1, s_2)$ where c, s_1, s_2 are obtained from an output $\langle B_1, B_2, B_3, c, s_1, s_2\rangle$ of a simulation of $\mathcal{S}$. The table of $\mathcal{H}$ is modified so that*

$$\langle X = (n, g, y_1, y_2, G_1, Y_1, G_2, Y_2, B_1, B_2, B_3), c\rangle$$

  is an entry of the table. If no such modification is possible (i.e., an entry X exists already in the table of $\mathcal{H}$) the experiment fails.

Let Exp1 *(resp.* Exp2*) be the event that experiment 1 (resp. 2) returns* 1. *It holds that* $|\mathbf{Prob}[\mathsf{Exp1}] - \mathbf{Prob}[\mathsf{Exp2}]| \leq q_{\mathcal{H}} \cdot 2^{-2k} + 2^{-l+1}$ *where $q_{\mathcal{H}}$ is the random oracle queries allowed to $\mathcal{A}$ during its first stage, assuming that $p'q' > 2^k$.*

Proof. First observe that the probability space over which the two experiments are defined is essentially identical: the only difference is that experiment 1 selects t_1, t_2 where experiment 2 selects s_1, s_2 (the domain in either case is identical). Consider now the following event Bad that refers to the first stage of the adversary and is defined as the event that the adversary produces a query to the random oracle $\mathcal{H}$ that is equal to $(n, g, y_1, y_2, G_1, Y_1, G_2, Y_2, B_1, B_2, B_3)$, where G_1, Y_1, G_2, Y_2 is the ciphertext produced after $\mathcal{A}$ terminates the first stage. It is easier to compare the two games as long as $\neg$Bad happens. Indeed in this case it is easy to see that the statistical distance between the two games is at most $2 \cdot 2^{-l} = 2^{-l+1}$ based on lemma 13. It follows that we can bound the statistical distance between the two games by $\mathbf{Prob}[\mathsf{Bad}] + 2^{-l+1}$. Now observe that the values G_1, G_2 are unknown to the adversary as they are selected on the fly after the stage 1 terminates. Given that $p'q' > 2^k$ it follows that the probability that $\mathcal{A}$ makes a single query to $\mathcal{H}$ and fixes G_1, G_2 is less than 2^{-2k}. The statement of the theorem follows easily. $\square$

The above proposition ensures that it was not harmful to attach a proof π along with our ciphertext since π carries a negligible amount of information about the random coin tosses used to encrypt M or about the message itself (at least in the random oracle model). Of course it is still not apparent whether the attachment of π to any ciphertext can be of any use for proving cca2 security. We establish the connection in the following proposition that we show that an adversary is incapable of producing a twin ciphertext and a string π that can convince the decryption test to not return $\perp$ when the twin ciphertext is inconsistent (i.e., each ciphertext encrypts different plaintexts). It follows that as long as a decryption oracle deems the ciphertext as valid this means that both siblings in the twin ciphertext encrypt the same message.

Proposition 15 *Consider the following probabilistic polynomial-time adversary $\mathcal{A}^{\mathcal{H}}$ that has access to a random oracle $\mathcal{H}$ and operates as follows: it receives a public-key of the encryption scheme $\langle \mathtt{Gen}'_{qr}, \mathtt{Enc}'_{qr}, \mathtt{Dec}'_{qr}\rangle$ and the factorization of n. $\mathcal{A}$ outputs a ciphertext $\psi = \langle G_1, H_1, G_2, H_2, \pi\rangle$. Consider the event* Cheat *to be the event that* $\mathtt{Dec}'_{qr}(\psi) \neq \perp$ *and* $(Y_1 \cdot G^{-x_1})^2 \neq (Y_2 \cdot G^{-x_2})^2 (\bmod n)$. *Suppose that* $\mathbf{Prob}[\mathsf{Cheat}] > 2^{-k}$*; then it holds that* $\mathbf{Prob}[\mathsf{Cheat}] \leq 2\sqrt{2} \cdot q_{\mathcal{H}} \cdot 2^{-k/2}$ *where $q_{\mathcal{H}}$ is the number of queries $\mathcal{A}$ poses to $\mathcal{H}$, assuming that $p', q' > 2^k$.*

Proof. Consider all ciphertexts ψ to be of the form ρ_1, c, ρ_2 where $\rho_1 = \langle n, g, y_1, y_2, G_1, Y_1, G_2, Y_2, g^{s_1}G_1^c, g^{s_2}G_2^c, (y_1^{s_1}Y_1^c)/(y_2^{s_2}Y_2^c)\rangle$ and $\rho_2 = \langle s_1, s_2\rangle$. Let Q be a predicate operating over a ciphertext such that $Q(\rho_1, c, \rho_2) = \top$ if and only if the event Cheat as defined in the theorem's statement is satisfied.

It follows that based on lemma 30 there is an algorithm that succeeds in producing two ciphertexts with the same first value (the ρ_1) and different challenges $c \neq c'$. We call the success probability of this the event Imp. From lemma 30 we obtain that $\mathsf{Pro}[\mathsf{Imp}] \geq \mathbf{Prob}[\mathsf{Cheat}]^2/(4q_{\mathcal{H}}) - (q_{\mathcal{H}}+1)2^{-k}$.

Consider now the setting when Imp happens. We have the following: $g^{s_1 - s_1^*} = G_1^{c^*-c}$, $g^{s_2 - s_2^*} = G_2^{c^*-c}$, and $y_1^{s_1 - s_1^*}/y_2^{s_2 - s_2^*} = (Y_1/Y_2)^{c^*-c}$. Now recall that $c, c^* < 2^k < p', q'$ thus it holds that $c^* - c$ is an invertible element in $\mathbb{Z}_{p'q'}$. From this we know that there exists an integer t such that $t = (c^* - c)^{-1} \mod p'q'$. For such integer t we can rewrite $G_1 = \sigma_1 g^{r_1}$ and $G_2 = \sigma_2 g^{r_2}$ where $r_1 = (s_1 - s_1^*)t \bmod p'q'$ and $r_2 = (s_2 - s_2^*)t \bmod p'q'$, and $\sigma_1, \sigma_2 \in \mathbb{Z}_n^*$ are elements of order 2. In a similar fashion we rewrite $y_1^{r_1}/y_2^{r_2} = Y_1/Y_2\sigma$ where σ is also ane element of order 2 inside $\mathbb{Z}_n^*$. From the above it follows that if $Y_2 = y_2^{r_2}m$, i.e., G_2, Y_2 is a ciphertext encrypting m then we have that G_1, Y_1 is a ciphertext encrypting $\sigma \cdot m$. This contradicts the fact that $(Y_1 \cdot G^{-x_1})^2 \neq (Y_2 \cdot G^{-x_2})^2 (\mathrm{mod} n)$. It follows that $\mathbf{Prob}[\mathsf{Imp}] = 0$. Given that $\mathbf{Prob}[\mathsf{Imp}] = 0$ we have that $\mathbf{Prob}[\mathsf{Cheat}]^2 \leq 4q_{\mathcal{H}}(q_{\mathcal{H}}+1)2^{-k}$. From this we obtain that $\mathbf{Prob}[\mathsf{Cheat}] \leq 2\sqrt{2} \cdot q_{\mathcal{H}} \cdot 2^{-k/2}$. □

Theorem 16 *The cryptosystem* $\mathtt{Gen}'_{qr}, \mathtt{Enc}'_{qr}, \mathtt{Dec}'_{qr}$ *satisfies* $\mathsf{cca2}$ *security in the indistinguishability sense under the* $\mathsf{DDH\text{-}Comp\text{-}KF}$ *in the Random-Oracle model.*

Proof. We will show how to transform any $\mathsf{cca2}$ adversary $\mathcal{A}^{\mathcal{H}}$ against the "twin" cryptosystem $\mathtt{Gen}'_{qr}, \mathtt{Enc}'_{qr}, \mathtt{Dec}'_{qr}$ to a CPA adversary $\mathcal{B}$ in the standard model against the cryptosystem $\langle \mathtt{Gen}_{qr}, \mathtt{Enc}_{qr}, \mathtt{Dec}_{qr}\rangle$. The CPA adversary $\mathcal{B}$ receives as input the public-key pk. Then it sets $\mathsf{pk}_1 = \mathsf{pk}$ and prepares pk_2 by selecting x_2 from the appropriate domain. In this way the twin public-key $\mathsf{pk}' = \langle \mathsf{pk}_1, \mathsf{pk}_2\rangle$ is formed. Then $\mathcal{B}$ starts the simulation of $\mathcal{A}$ giving pk'. Whenever $\mathcal{A}$ submits a query to the random oracle $\mathcal{H}$, $\mathcal{B}$ uses the on-the-fly generated table for $\mathcal{H}$ to answer consistently. Whenever $\mathcal{A}$ submits a twin ciphertext for decryption to its decryption oracle, $\mathcal{B}$ parses the ciphertext as $\langle G_1, Y_1, G_2, Y_2, \pi\rangle$, verifies the non-interactive proof π, and if the proof is valid it responds by $(G_2^{-x_2}Y_2)^2$ (contrary to the standard $\mathsf{cca2}$ simulation where the answer $(G_1^{-x_1}Y_1)^2$ is given instead).

When $\mathcal{A}$ provides the two challenge plaintexts m_0, m_1, $\mathcal{B}$ forwards them to its own challenge oracle to obtain the challenge ciphertext G_1^*, Y_1^*. Then, $\mathcal{B}$ computes G_2^*, Y_2^* at random from the underlying group and produces a simulated proof π^* by inserting the appropriate value into the random oracle table $\mathcal{H}$ (if this is not possible then $\mathcal{B}$ simply fails). Observe that the challenge ciphertext $\psi^* = \langle G_1^*, Y_1^*, G_2^*, Y_2^*, \pi^*\rangle$ is not valid (i.e., the two components encrypt different plaintexts). $\mathcal{B}$ proceeds with the simulation of $\mathcal{A}$ by providing the challenge ciphertext and continues the simulation of

$\mathcal{A}$. The simulation of the decryption oracle for $\mathcal{A}$ in the second stage is similar to the first stage with the following difference: if a ciphertext query of the form $\langle G_1', Y_1', G_2', Y_2', \pi' \rangle \neq \psi^*$ is such that π' is a valid proof and $\langle G_1', Y_1' \rangle = \langle G_1^*, Y_1^* \rangle$ then $\mathcal{B}$ fails; in other cases whenever π' is a valid proof then $\mathcal{B}$ replies as in the first stage of the simulation. Finally $\mathcal{B}$ terminates by returning the output that $\mathcal{A}$ returns. Clearly $\mathcal{B}$ is a CPA adversary for $\langle \mathtt{Gen}_{qr}, \mathtt{Enc}_{qr}, \mathtt{Dec}_{qr} \rangle$.

We will show that the success probability of $\mathcal{B}$ is close by a negligible fraction to the success probability of $\mathcal{A}^{\mathcal{H}}$. We will establish this by observing the following sequence of games. First consider game G_0 to be the standard indistinguishability cca2 game that $\mathcal{A}^{\mathcal{H}}$ plays and wins with some probability of success.

Consider the following modification to game G_0 that results in game G_1. Instead of answering the decryption queries of $\mathcal{A}$ as $(G_1^{-x_1} Y_1)^2$ we answer them by $(G_2^{-x_2} Y_2)^2$. Clearly the distance between the two games would be bounded by $\mathbf{Prob}[\mathsf{Bad}_0]$ where Bad_0 is the event that the adversary $\mathcal{A}$ produces a valid non-interactive proof π for a twin ciphertext $\langle G_1, Y_1, G_2, Y_2, \pi \rangle$ that satisfies $(G_1^{-x_1} Y_1)^2 \neq (G_2^{-x_2} Y_2)^2 (\bmod n)$. It is easy to bound the probability of the event Bad_0 using proposition 15: based on the statement of the proposition we know that this happens with probability $\mathsf{c} \cdot q_{\mathcal{H}} 2^{-k/2}$ where c is a small constant and thus distance between game G_0 and game G_1 is negligible (assuming that $q_{\mathcal{H}} q_{\mathsf{dec}} 2^{-k/2}$ is negligible, where q_{dec} is the number of decryption oracle queries). Note that game G_1 does not employ the secret key of x_1 at all.

Next we modify game G_1 into game G_2 in the generation of the challenge ciphertext: instead of preparing the challenge ciphertext according to the specifications we produce a fake proof by inserting the appropriate value into the table of $\mathcal{H}$ and using the simulator $\mathcal{S}$ in the same way that this is performed in proposition 14. Now observe the following: the only difference between game G_1 and game G_2 is in the way that the non-interactive proof π^* in the challenge ciphertext is computed. Observe now that games G_1 and G_2 define the two experiments of proposition 14 and thus it follows that the statistical distance between G_1 and G_2 isat most $2^{-2k} q_{\mathcal{H}} + 2^{-l+1}$.

Finally we perform the following modfication to G_2 to obtain a game G_3: we modify again the challenge ciphertext so that G_2, Y_2 are selected at random from $QR(n)$. It is easy to see that this modification can incur a distance between the games G_2 and game G_3 that is bounded by the best possible advantage a polynomial-time distinguisher may have against DDH-Comp-KF. Finally observe that G_3 is identical to the operation of $\mathcal{B}$ as defined above. Given that $\mathcal{B}$ is a CPA indistinguishability attacker against a cryptosystem that is secure under DDH-Comp-KF we conclude the proof.

□

Remark. Having completed the presentation of the cryptosystem $\mathtt{Gen}'_{qr}, \mathtt{Enc}'_{qr}, \mathtt{Dec}'_{qr}$ a number of observations are in place (that will be of importance later in the construction of the group signature):

1. The encryption and decryption functions *do not* require the factorization of n.

2. The factorization n is made available to the adversary.

3. The decryption does not invert the encryption operation entirely as it returns the square of the encrypted plaintext. While the availability of the factorization can recover the plaintext, such recover will be unnecessary in the group signature construction that we will present.

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 scheme of [2] 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 this simpler case in our model.

Definition 17 *A group signature scheme is a digital signature scheme that comprises 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. Moreover* `SETUP` *initializes a public-state string St with two components $St_{users} = \emptyset$ (a set data structure) and $St_{trans} = \epsilon$ (a string data structure).*

`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) $\mathsf{J}_{\mathsf{user}}, \mathsf{J}_{\mathsf{GM}}$. Only $\mathsf{J}_{\mathsf{user}}$ has a private output tape. An execution of the protocol is denoted as $[\mathsf{J}_{\mathsf{user}}(1^\nu, \mathcal{Y}), \mathsf{J}_{\mathsf{GM}}(1^\nu, St, \mathcal{Y}, \mathcal{S})]$ and has two "output" components: the private output of the user, $\langle i, \mathsf{cert}_i, \mathsf{sec}_i\rangle \leftarrow \mathsf{U}[\mathsf{J}_{\mathsf{user}}(1^\nu, \mathcal{Y}), \mathsf{J}_{\mathsf{GM}}(1^\nu, St, \mathcal{Y}, \mathcal{S})]$ and the public transcript, $\langle i, \mathsf{transcript}_i\rangle \leftarrow \mathsf{T}[\ \mathsf{J}_{\mathsf{user}}(1^\nu, \mathcal{Y}), \mathsf{J}_{\mathsf{GM}}(1^\nu, St, \mathcal{Y}, \mathcal{S})]$. After a successful execution of* `JOIN` *the following (public) updates are made to the state: $St_{users} = St_{users} \cup \{i\}$ 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* `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* `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* `OPEN`$(m, \sigma, \mathcal{Y}, \mathcal{S}, St) \in St_{users} \cup \{\bot\}$.

Note: the identity of the user that gets the i-th user's membership certificate is assumed to be authenticated and thus associated with i.

Notation. We will write $\langle i, \mathsf{cert}_i, \mathsf{sec}_i\rangle \leftrightharpoons_{\mathcal{Y}} \langle i, \mathsf{transcript}_i\rangle$ 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 $\mathcal{Y}$ and a state St (note that we omit St in the subscript for convenience). Moreover, any given cert, based on a public-key $\mathcal{Y}$, has a corresponding sec; we will also denote this relationship by $\mathsf{cert} \leftrightharpoons_{\mathcal{Y}} \mathsf{sec}$ (overloading the notation). We remark that $\leftrightharpoons_{\mathcal{Y}}$ in both cases, will be considered a polynomial-time relationship in the parameter ν.

Given a $\langle \mathcal{Y}, \mathcal{S}\rangle \leftarrow$ `SETUP`(1^ν), a public-state St is called *well-formed* if it is effectively produced by a Turing machine M that has unlimited access to a J_{GM} oracle (following the public state update procedures as in definition 17). A well-formed state St' is said to extend state St, if it is effectively produced by a Turing machine as above but with the public-state initially set to St instead of $\langle \emptyset, \epsilon\rangle$.

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 $\mathsf{J}_{\mathsf{user}}$ 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 $\texttt{OPEN}$ algorithm succeeds in identifying the originator of any signature generated according to specifications. Formally,

Definition 18 *A group signature is correct if the following statements hold with very high probability over the coin tosses of all procedures. Let* $\langle \mathcal{Y}, \mathcal{S}\rangle \leftarrow \texttt{SETUP}(1^\nu)$.

- User tagging soundness. *In every well formed public-state* St *it holds that the cardinality of the set* St_{users} *equals the number of transcripts in the string* St_{trans}.
- 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*,*

$$\texttt{VERIFY}(\mathcal{Y}, m, \texttt{SIGN}(\mathcal{Y}, \mathsf{cert}, \mathsf{sec}, m)) = \top$$

- Opening soundness. *For any certificate* cert_i *and secret* sec_i*, transcript* $\mathsf{transcript}_i$ *and well-formed public-state* St *s.t.* $\langle i, \mathsf{cert}_i, \mathsf{sec}_i\rangle \leftrightharpoons_{\mathcal{Y}} \langle i, \mathsf{transcript}_i\rangle$*, if* St' *is a well-formed public-state that extends* St *with* $\langle i, \mathsf{transcript}_i\rangle \in St'_{trans}$*, then for any message* m*, and any* $\sigma \leftarrow \texttt{SIGN}(\mathcal{Y}, \mathsf{cert}_i, \mathsf{sec}_i, m)$ *it holds that* $\texttt{OPEN}(m, \sigma, \mathcal{Y}, \mathcal{S}, St') = i$.

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 [22]. 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 employs a data structure called state $\mathsf{state}_{\mathcal{I}}$ and is initialized as $\langle St, \mathcal{Y}, \mathcal{S}\rangle \leftarrow \texttt{SETUP}(1^\nu)$. The interface accepts the types of queries listed below. We remark that during an attack the adversary interacts with the interface and the oracles in a stateful fashion and the interface performs a number of bookkeeping operations that involve $\mathsf{state}_{\mathcal{I}}$ as explained below.

- $\mathcal{Q}_{\mathsf{pub}}$ and $\mathcal{Q}_{\mathsf{key}}$: the interface looks up $\mathsf{state}_{\mathcal{I}}$ and 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 (if it is successfully terminated) will be entered in St_{users} and the transcript in St_{trans} following the updating rules of definition 17. Additionally the user will be marked as U^a (adversarially controlled).
- $\mathcal{Q}_{\mathsf{b-join}}$: the interface initiates a protocol dialog simulating $\mathsf{J}_{\mathsf{user}}$. The user created from this interaction (if successfully terminated) will be entered in St_{users} and the transcript into St_{trans} as described in the update procedure of definition 17. Additionally, the user will be marked as U^b. Upon successful termination the resulting membership certificate and membership secret (i.e., the whole output of the user protocol including the user name tag) will be appended in a *private* area of $\mathsf{state}_{\mathcal{I}}$.

  Following the above we note that the adversary when executing the $\mathcal{Q}_{\mathsf{b-join}}$ query will be effectively required by the interface to choose a unique and properly defined tag for the current user (according to definition 17). This is not a restriction since this can be enforced in practice by having the user checking the public user name database during normal protocol executions. Note that even when assuming an adversarial GM the user name database St_{users} is not entirely adversarially controlled; indeed, if St_{users} is compromised then clearly no group signature scheme can have any form of identification robustness.

- $\mathcal{Q}_{\mathsf{read}}, \mathcal{Q}_{\mathsf{write}}$: these two queries allow to the adversary to read and write respectively $\mathsf{state}_{\mathcal{I}}$. The query $\mathcal{Q}_{\mathsf{read}}$ returns the whole $\mathsf{state}_{\mathcal{I}}$ excluding the public and secret-key as well as the private area of $\mathsf{state}_{\mathcal{I}}$ that is used for the $\mathcal{Q}_{\mathsf{b-join}}$ queries. The query $\mathcal{Q}_{\mathsf{write}}$ is allowed to perform arbitrary changes as long as it does not remove/corrupt elements from St_{users}, St_{trans} (but e.g., insertion to these structures is allowed).

- $\mathcal{Q}_{\mathsf{sign}}(i, m)$: given that $i \in U^b$ the interface simulates a signature on m by looking up the membership certificate and membership secret available in the private area of $\mathsf{state}_{\mathcal{I}}$ and returns a corresponding signature.

- $\mathcal{Q}_{\mathsf{open}}(\sigma)$: the interface applies the opening algorithm to the given signature σ using the current St. If S is a set of signatures we denote by $\mathcal{Q}_{\mathsf{open}}^{\neg S}$ 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 involves three attack scenarios and corresponding security definitions. These security properties are based on our modeling of Traceable Signatures, [23], and are ported from the traceable signature setting to the group signature setting, augmenting them with adversarial opening capability. In particular, we use the same terminology for the attacks to facilitate the comparison between these two primitives.

The first security property relates to an adversary that wishes to misidentify itself. In a misidentification attack the adversary is allowed to join the system through $\mathcal{Q}_{\mathsf{a-join}}$ queries and open signatures at will; finally he produces a forged group signature (cf. an existential adaptive chosen message attack, [22]) that does not open into one of the users he controls (actually without loss of generality the adversary controls all users of the system; thus the adversary wins if the opening algorithm returns $\perp$).

```
The Misidentification-Attack Game $G^{\mathcal{A}}_{\mathsf{mis}}$ (denoted by $G^{\mathcal{A}}_{\mathsf{mis}}(1^\nu)$):
1. $\mathsf{state}_{\mathcal{I}} = \langle St, \mathcal{Y}, \mathcal{S} \rangle \leftarrow \mathtt{SETUP}(1^\nu)$;
2. $\langle m, \sigma \rangle \leftarrow \mathcal{A}^{\mathcal{I}[\mathcal{Q}_{\mathsf{pub}}, \mathcal{Q}_{\mathsf{a-join}}, \mathcal{Q}_{\mathsf{read}}, \mathcal{Q}_{\mathsf{open}}]}(1^\nu)$
3. $i = \mathtt{OPEN}(m, \sigma, \mathcal{Y}, \mathcal{S}, St)$
4. If $(\mathtt{VERIFY}(\mathcal{Y}, m, \sigma) = \top) \wedge (i \notin U^a)$ then return $\top$ else return $\perp$.
```

Our second security property relates to a framing 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. Note that the adversary can take advantage of $\mathcal{Q}_{\mathsf{write}}$ to create dummy users if it so wishes.

```
The Framing-Attack Game $G^{\mathcal{A}}_{\mathsf{fra}}$ (denoted by $G^{\mathcal{A}}_{\mathsf{fra}}(1^\nu)$):
1. $\mathsf{state}_{\mathcal{I}} = \langle St, \mathcal{Y}, \mathcal{S} \rangle \leftarrow \mathtt{SETUP}(1^\nu)$;
2. $\langle m, \sigma \rangle \leftarrow \mathcal{A}^{\mathcal{I}[\mathcal{Q}_{\mathsf{pub}}, \mathcal{Q}_{\mathsf{key}}, \mathcal{Q}_{\mathsf{b-join}}, \mathcal{Q}_{\mathsf{read}}, \mathcal{Q}_{\mathsf{write}}, \mathcal{Q}_{\mathsf{sign}}]}(1^\nu)$
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 $\perp$.
```

Finally we model anonymity. In an anonymity-attack the adversary operates in two stages play and guess. In the play stage the adversary is allowed to join the system through $\mathcal{Q}_{\mathsf{a-join}}$ queries, as well open signatures through $\mathcal{Q}_{\mathsf{open}}$ queries. The adversary terminates the play stage by providing a pair of membership certificates/secrets (that were possibly obtained through $\mathcal{Q}_{\mathsf{a-join}}$ 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 play stage with the exception that the adversary is not allowed to open the challenge signature. Note that this attack is similar to a $\mathsf{cca2}$ attack when an individual group signature is considered an identity concealing ciphertext.

```
The Anonymity-attack Game $G^{\mathcal{A}}_{\mathsf{anon}}$ (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. $\langle aux, m, \mathsf{cert}_1, \mathsf{sec}_1, \mathsf{cert}_2, \mathsf{sec}_2, \rangle \leftarrow \mathcal{A}^{\mathcal{I}[\mathcal{Q}_{\mathsf{pub}}, \mathcal{Q}_{\mathsf{a-join}}, \mathcal{Q}_{\mathsf{read}}, \mathcal{Q}_{\mathsf{open}}]}(\mathsf{play}, 1^{\nu})$
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 $\perp$;
4. Choose $b \leftarrow_R \{1, 2\}$;
5. $\sigma \leftarrow \mathtt{SIGN}(\mathcal{Y}, \mathsf{cert}_b, \mathsf{sec}_b, m)$;
6. $b^* \leftarrow \mathcal{A}^{\mathcal{I}[\mathcal{Q}_{\mathsf{pub}}, \mathcal{Q}_{\mathsf{a-join}}, \mathcal{Q}_{\mathsf{read}}, \mathcal{Q}_{\mathsf{open}}^{\neg\{\sigma\}}]}(\mathsf{guess}, aux)$;
7. if $b = b^*$ return $\top$ else return $\perp$;
```

Definition 19 *A group signature scheme is secure if for all PPT $\mathcal{A}$ it holds that (i)* $\mathbf{Prob}[G^{\mathcal{A}}_{\mathsf{mis}}(1^{\nu}) = \top] = \mathsf{negl}(\nu)$ *(ii)* $\mathbf{Prob}[G^{\mathcal{A}}_{\mathsf{fra}}(1^{\nu}) = \top] = \mathsf{negl}(\nu)$ *and (iii)* $2\mathbf{Prob}[G^{\mathcal{A}}_{\mathsf{anon}}(1^{\nu}) = \top] - 1 = \mathsf{negl}(\nu)$.

Capturing the intuitive security properties put forth by [2]. Given the above security model is relatively straightforward to see that the informal security properties that were put forth by [2] are captured by the above three security properties. In particular, (1) *Unforgeability*: an adversary that given the public-key forges a signature, will either produce a signature that opens to $\perp$ or a signature that opens to one of the users; such an attack is prevented by both misidentification and framing security above; (2) *Anonymity*, is captured by the anonymity security property above, (3) *Unlinkability*, is also captured by the anonymity security property, (4) *Exculpability* is captured by framing security (since the secret-key of the GM is released to the adversary), (5) *Traceability*, is ensured by misidentification (a signer cannot produce a signature that opens to $\perp$) and framing security (a signer cannot frame another user). (6) *Coalition resistance* is built-in into our security properties since w.r.t. misidentification we allow the adversary to adaptively build a coalition of malicious users, whereas in the case of framing attack the adversary has the GM's key (and as a result it can build a coalition if it wishes it).

We remark that independently of the present work (which originally appeared in [25]), [5] presented a group signature formal model that captures the dynamic group case as the present paper does. We note that the construction presented in that paper is a feasibility result rather than a practical construction since it employs generic zero-knowledge techniques. That work also explicitly design the underlying authenticated channel (assumed here and in prior schemes) with public key infrastructure.

6 Building a Secure Group Signature

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 parameters k, l.

The membership certificates are of the form $\langle A, e \rangle$ so that $A \in QR(n)$ and e is a prime number in Γ. The membership secret is a value $x \in \Lambda$ such that $a_0 a^x = A^e$. Note that $\Gamma = [\gamma_0, \gamma_1], \Lambda = [\lambda_0, \lambda_1]$ are integer ranges within $\{1, \ldots, p'q'\}$ such that the following two conditions are satisfied: (i)

$2^{k+l+2} \cdot (\lambda_1 - \lambda_0 + \gamma_1 - \gamma_0) + \lambda_1 + 2^{\nu/2-1-\mathcal{O}(\log \nu)} < \gamma_0$ (ii) $(\gamma_0)^2 > 2^{k+l+2} \cdot (\gamma_1 - \gamma_0) + \gamma_1$. In particular we may select the values as follows: let ν be the number of bits for which an RSA safe composite modulus is considered to be hard to factor; then we set $\Lambda = [\lambda_0, \lambda_1] = [2^{\nu/2-4} - 2^{\nu/2-k-l-7}, 2^{\nu/2-4}]$ and $\Gamma = [\gamma_0, \gamma_1] = [2^{\nu/2-2}, 2^{\nu/2-2} + 2^{\nu/2-k-l-7}]$.

Before we advance to the description of the construction we will prove the following basic lemma that shows that given a set of certificates, under the Strong-RSA assumption, it is hard to produce one additional certificate, even if the produced certificate is allowed to be slightly "malformed." We note that a weaker formulation of the lemma below is part of the exposition of [2] (note that such weaker formulation seems insufficient for the proof of security).

Lemma 20 *Let n be an RSA modulus and $k, l \in \mathbb{Z}$ as above and $a, a_0 \in QR(n)$ be two random quadratic residues. Also, for $i = 1, \ldots, K$, let $\langle A_i, e_i, x_i \rangle$ be such that $A_i^{e_i} = a_0 a^{x_i} (\mathrm{mod} n)$ $e_i \in \Gamma \cap \mathsf{Prime}$, $x_i \in \Lambda$, where $\Gamma = [\gamma_0, \gamma_1]$ and $\Lambda = [\lambda_0, \lambda_1]$ are integer ranges as defined above. Let $\mathcal{A}$ be an PPT such that given $n, a, a_0, \{\langle A_i, e_i, x_i \rangle\}_{i=1}^K$ it returns a tuple $\langle A, e, x \rangle$ such that $A^e = \pm a_0 a^x (\mathrm{mod} n)$, $\langle A, e, x \rangle \notin \{\langle A_i, e_i, x_i \rangle\}_{i=1}^K$ and $x \in [\lambda_0 - 2^{k+l+2}(\lambda_1 - \lambda_0), \lambda_1 + 2^{k+l+2}(\lambda_1 - \lambda_0)]$ and $e \in [\gamma_0 - 2^{k+l+2}(\gamma_1 - \gamma_0), \gamma_1 + 2^{k+l+2}(\gamma_1 - \gamma_0)] \cap \mathsf{Odd}$. Then the Strong-RSA assumption fails.*

Proof. Let z, n be a challenge for the Strong-RSA problem. Consider first the following PPT $\mathcal{B}_1$:

$\mathcal{B}_1$, selects random elements $x_i \in_R \Lambda$ and $e_i \in_R \Gamma \cap \mathsf{Prime}$. Then it sets $a_0 = z^{r e_1 \ldots e_K}$ and $a = z^{e_1 \ldots e_K}$ where $r \in_R [0, 2^{\nu/2-1-\mathcal{O}(\log \nu)}]$, $e_i \in_R \Gamma \cap \mathsf{Prime}$, and $x_i \in_R \Lambda$. Next $\mathcal{B}_1$ computes $A_i = (a_0 a_i)^{1/e_i} (\mathrm{mod} n)$ (observe that the factorization of n is not needed). As proved in [23], assuming the hardness of factoring, the selection of $r \in_R [0, 2^{\nu/2-1-\mathcal{O}(\log \nu)}]$ is sufficient to make a_0^r indistinguishable from a random element of $QR(n)$ if a_0 is a generator of $QR(n)$ (and in fact a_0 is with overwhelming probability). This ensures that the public-key (as well as the values A_i) are selected in a manner indistinguishable to the main protocol.

$\mathcal{B}_1$ proceeds to simulate $\mathcal{A}$ to obtain the value A, e, x with the stated range constraints. If it holds that $\gcd(e, e_1 \ldots e_K) > 1$ then $\mathcal{B}_1$ aborts. Otherwise, $\mathcal{B}_1$ proceeds as follows: first, denote by $\rho = e_1 \ldots e_K (x + r)$. $\mathcal{B}_1$ computes $\delta = \gcd(e, \rho) = \gcd(e, x + r)$. Observe that due to the properties of ranges Γ, Λ it holds necessarily that $\delta < e$.

It follows that $\mathcal{B}_1$ can compute A, e, ρ so that $A^e \pm z^\rho$, e is odd and $\delta = \gcd(e, \rho) < e$. We show how to solve the given String-RSA instance given such values below. If $\delta = 1$, $\mathcal{B}_1$ computes α, β such that $\alpha e + \beta \rho = 1$ and it follows that $z = z^{e\alpha} z^{\rho \beta} = z^{e\alpha} (\pm A^e)^\beta = \pm (z^\alpha A^\beta)^e = (\pm z^\alpha A^\beta)^e$ where the last equality holds since e is odd. Obviously $\mathcal{B}_1$ can recover a solution to the strong-RSA challenge by using A, α, β in this case ($\delta = 1$).

In case $\delta > 1$ $\mathcal{B}_1$ will proceed as follows: let $\tilde{\delta} = \gcd(\delta, 2p'q')$ (recall $2p'q'$ is the exponent of $\mathbb{Z}_n^*$). If $\tilde{\delta} = p'q'$ a multiple of $\phi(n)$ can be easily obtained from δ and thus the factorization of n can be recovered. On the other hand if $\tilde{\delta} = 1$ it holds that $A^{\frac{e}{\delta}} = \pm z^{\frac{\rho}{\delta}}$ from which we can solve the Strong-RSA problem as described in the previous paragraph.

Next, we consider the case that $\tilde{\delta} = p'$. This means that $\delta = p' \cdot s$, i.e., $\mathcal{B}_1$ can compute a multiple of p'. It follows that if $b \in_R \mathbb{Z}_n^*$ with probability $1/2$ it will hold that $b^\delta = 1 \bmod p$, and thus we can see that $\gcd(b^\delta \bmod n - 1, n)$ will reveal a factor of n. Note that the case $\tilde{\delta} = q'$ is of course identical. Finally observe that the case $\tilde{\delta} = 2$ would mean that δ is even and thus e is even as well something that has been excluded in the theorem's statement. This completes the description and analysis of $\mathcal{B}_1$ that solves the given Strong-RSA instance with non-negligible probability as long as $\mathcal{A}$ finds a new certificate with non-negligible probability conditioned on the fact that $\gcd(e, e_1 \ldots e_K) = 1$.

Now we describe a second algorithm called $\mathcal{B}_2$: $\mathcal{B}_2$, selects random elements $x_i \in_R \Lambda$ and $e_i \in_R \Gamma \cap \mathsf{Prime}$. It also selects j at random from $1, \ldots, K$. Then it sets $A_j = z^{r^{\frac{e_1 \cdots e_K}{e_j}}}$ and $a = z^{\frac{e_1 \cdots e_K}{e_j}}$

where $r \in_R [0, 2^{\nu/2-1-\mathcal{O}(\log\nu)}]$, $e_i \in_R \Gamma \cap \mathsf{Prime}$, and $x_i \in_R \Lambda$. Next $\mathcal{B}_1$ computes $a_0 = A_j^{e_j}/a^{x_j}$ and for $i = 1, \ldots, K$, $i \neq j$, $A_i = (a_0 a_i)^{1/e_i} (\mathrm{mod} n)$ (observe that the factorization of n is not needed). As shown in [23] the selection of $r \in_R [0, 2^{\nu/2-1-\mathcal{O}(\log\nu)}]$ is sufficient to make the two elements a_0, a indistinguishable from random in $QR(n)$ (under the hardness of factoring).

Given the above values $\mathcal{B}_2$ proceeds to simulate $\mathcal{A}$ and obtains values A, e, x that satisfy the theorem's constraints. $\mathcal{A}$ computes $\delta = \gcd(e, e_j)$ and if $\delta = 1$ it aborts. Otherwise observe that due to the fact that e_j is a prime it holds that $\delta = e_j$. It follows that $e = \tilde{e}e_j$ and we can write $A^e = a_0 a^x$ as $A^{\tilde{e}e_j} = \pm z^{\frac{e_1 \ldots e_K}{e_j}(r+x-x_j)}$. Let $\delta = \gcd(\tilde{e}e_j, \frac{e_1 \ldots e_K}{e_j}(r + x - x_j))$. First observe that due to the range properties that relate to Γ it is impossible to have that some e_i with $i \neq j$ divides $\tilde{e}$ (this would make $e = \tilde{e}e_j$ too large). Thus it holds that $\delta = \gcd(\tilde{e}e_j, r + x - x_j)$. Moreover, again due to the properties of Λ and Γ we have that $e_j > r + x - x_j$ thus $\delta < e_j \leq e$. It follows that if $\rho = \frac{e_1 \ldots e_K}{e_j}(r + x - x_j)$, $\mathcal{B}_2$ can compute values A, e, ρ such that $A^e = \pm z^\rho$ with e an odd number and so that $\delta = \gcd(e, \rho) < e$. It follows that $\mathcal{B}_2$ can solve the Strong-RSA instance in a similar fashion as $\mathcal{B}_1$.

Now consider the following Strong-RSA solver that employs both $\mathcal{B}_1, \mathcal{B}_2$: $\mathcal{B}$ flips a coin and simulates either $\mathcal{B}_1$ or $\mathcal{B}_2$. It is easy to see that if $\mathcal{A}$ is successful with non-negligible probability then $\mathcal{B}$ will produce a solution to the given Strong-RSA instance with non-negligible probability. $\square$

6.1 The Construction

In this section we provide our construction for the group signature scheme. Note that the construction makes use of a hash function $\mathcal{H}$ that is modeled as a random oracle.

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 $x, \hat{x} \leftarrow_R \mathbb{Z}^*_{p'q'}$, $a_0, a, h \leftarrow_R QR(n)$ 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 $k, l \in \mathbb{N}$ as polynomially related functions in ν. The ranges Γ, Λ are also defined as in the beginning of the section.

JOIN: A protocol between the GM and a user that allows the joint computation of a membership certificate $\langle A_i, e_i \rangle$ so that only the user obtains the membership secret x_i. We give the functionality of the protocol using a trusted party T: first $\mathsf{J}^T_{\mathsf{user}}(1^\nu, \mathcal{Y})$ sends "go" to the trusted party T, who in turn selects $x_i \leftarrow_R \Lambda$ and writes to the GM's communication tape the value $C_i = a^{x_i} \bmod n$ and writes to the user's private tape the value x_i. $\mathsf{J}^T_{\mathsf{GM}}(1^\nu, \mathcal{Y}, \mathcal{S})$ reads C_i from the communication tape with T, it selects a prime $e_i \leftarrow_R \Gamma - \{p', q'\}$ and computes $A_i = (a_0 a)^{1/e_i} (\mathrm{mod} n)$; finally it writes $\langle i, A_i, e_i \rangle$ in the communication tape where i is the next available user tag (a counter is employed) and terminates. $\mathsf{J}^T_{\mathsf{user}}$ reads $\langle i, A_i, e_i \rangle$ from the communication tape and writes $\langle i, A_i, e_i, x_i \rangle$ in its private output tape. As shown in the "non-adaptive drawings of random powers" protocol of [23] it is possible to derive an efficient protocol $\mathsf{J}_{\mathsf{user}}, \mathsf{J}_{\mathsf{GM}}$ that does not employ a trusted party and achieves the above ideal functionality. We remark that the GM is accepting join protocols only in a sequential fashion.

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

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:

$$r, \hat{r}, \tilde{r} \leftarrow_R \lfloor n/4 \rfloor \ : \ T_1 = A_i y^r,$$

$$T_2 = g^r, \ \hat{T}_1 = A_i \hat{y}^{\hat{r}}, \ \hat{T}_2 = g^{\hat{r}}, \ T_3 = g^{e_i} h^{\tilde{r}}$$

To complete the description of the signature, we need a proof of knowledge for the variables $r, \hat{r}, e_i, x_i, s', s''$, so that they satisfy the following relations: $T_2 = g^r, \hat{T}_2 = g^{\hat{r}}, T_1/\hat{T}_1 = y^r/\hat{y}^{\hat{r}}, T_3 = g^{e_i}h^r, T_2^e = g^{s'}, a_0 a^{x_i} y^{s'} = T_1^{e_i}, T_3 = g(g^2)^{s''}h^r$.

This proof ensures that $T_1, T_2, \hat{T}_1, \hat{T}_2$ is a "twin" ElGamal encryption of a value A that if raised to an odd integer e_i, it can be split by the prover in the form $a_0 a^{x_i}$. The signature on a message M will be formed by employing the Fiat-Shamir transform over the proof of knowledge. The proof of knowledge itself is an extension of the protocol of definition 12 and we describe it in detail in definition 21 below.

VERIFY: given a signature $\sigma = \langle T_1, T_2, \hat{T}_1, \hat{T}_2, T_3, c, s_1, s_2, s_3, s_4, s_5, s_6, s_7 \rangle$ the verification algorithm will apply the verification algorithm of the non-interactive proof of knowledge.

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. Compute $A = (T_1 T_2^{-x})^2 \bmod n$.

4. Match A to the square of some user's first component of the membership certificate $\langle A_i, e_i \rangle$ (as available in the database St_{trans} maintained during the JOIN protocols).

5. If either steps 1 or 3 or 5 fail, return $\perp$, else return the user found in step 5.

Remark. In order to ensure non-repudiation in the opening procedure it will be useful that each user signs his C_i value based on a PKI. Then, based on the PKI, the GM will be capable of proving that a signature opens to a certain user in a non-repudiable fashion. It is also possible for the GM to issue a proof that it performs the decryption correctly. We chose not to include these functionalities into the formal model of group signatures for the sake of keeping the model simple (note that they can be modularly added easily).

Definition 21 The group signature proof of knowledge. *We describe it as an interactive protocol first between a prover and a verifier. Both prover and verifier have input the public parameters as well as* $T_1 = A_i y^r, T_2 = g^r, \hat{T}_1 = A_i \hat{y}^{\hat{r}}, \hat{T}_2 = g^{\hat{r}}, T_3 = g^{e_i} h^{\tilde{r}}$, *and the prover has additional input the values* $r, \hat{r}, \tilde{r}, e_i, x_i$. *The prover computes also the values* $s' = e_i \cdot r$, $s'' = (e_i - 1)/2$. *The interaction between the prover and the verifier is as follows: the prover selects* t_r, $t_{\hat{r}}$, $t_{\tilde{r}}$ $\in_R [-2^{k+l}\lfloor n/4 \rfloor, 2^{k+l}\lfloor n/4 \rfloor]$ $t_{e_i} \in_R [-2^{k+l}\Delta\gamma, 2^{k+l}\Delta\gamma]$, *where* $\Delta\gamma = \gamma_1 - \gamma_0$, $t_{x_i} \in_R [-2^{k+l}\Delta\lambda, 2^{k+l}\Delta\lambda]$, *where* $\Delta\lambda = \lambda_1 - \lambda_0$, $t_{s'} \in_R [-2^{k+l}\Delta\tau, 2^{k+l}\Delta\tau]$, *where* $\Delta\tau = \tau_1 - \tau_0$ *and* $\tau_1 = \gamma_1$, $\tau_0 = \lfloor (\gamma_0 - 1)/2 \rfloor$, $t_{s''} \in_R [-2^{k+l}\Delta\mu, 2^{k+l}\Delta\mu]$, *where* $\Delta\mu = \mu_1 - \mu_0$ *and* $\mu_1 = \gamma_1 \cdot \lfloor n/4 \rfloor$, $\mu_0 = 0$.

The prover transmits to the verifier the values $B_1 = g^{t_r}, B_2 = g^{t_{\hat{r}}}, B_3 = y_1^{t_r}/y_2^{t_{\hat{r}}}, B_4 = g^{t_{e_i}} h^{t_{\tilde{r}}}$, $B_5 = (T_2^{-1})^{t_{e_i}} g^{t_{s'}}$, $B_6 = a^{t_{x_i}} y^{t_{s'}} (T_1^{-1})^{t_{e_i}}$, $B_7 = (g^2)^{t_{s''}} h^{\tilde{r}}$. *The verifier responds by a challenge* $c \in \{0,1\}^k$, *and subsequently the prover computes* $s_r = t_r - c \cdot r$, $s_{\hat{r}} = t_{\hat{r}} - c \cdot \hat{r}$, $s_{\tilde{r}} = t_{\tilde{r}} - c \cdot \tilde{r}$, $s_{e_i} = t_{e_i} - c \cdot (e_i - \gamma_0)$, $s_{x_i} = t_{x_i} - c \cdot (x_i - \lambda_0)$, $s_{s'} = t_{s'} - c \cdot s'$, $s_{s''} = t_{s''} - c \cdot (s'' - \tau_0)$ *(all over* $\mathbb{Z}$*) and transmits to the verifier the values* $s_r, s_{\hat{r}}, s_{\tilde{r}}, s_{e_i}, s_{x_i}, s_{s'}, s_{s''}$. *The verification check is as follows:* $g^{s_r}(T_2)^c =_? B_1$, $g^{s_{\hat{r}}}(\hat{T}_2)^c =_? B_2$ *and* $(y^{s_r}/\hat{y}^{s_{\hat{r}}})(T_1/\hat{T}_1)^c =_? B_3$, $g^{s_{e_i}} h^{s_{\tilde{r}}} (T_3 g^{-\gamma_0})^c =_? B_4$, $(T_2^{-1})^{s_{e_i}} g^{s_{s'}} (T_2^{\gamma_0})^c =_? B_5$, $a^{s_{x_i}} y^{s_{s'}} (T_1^{-1})^{s_{e_i}} (a_0^{-1})^c (a^{-\lambda_0} T_1^{\gamma_0})^c =_? B_6$, $(g^2)^{s_{s''}} h^{s_{\tilde{r}}} (T_3 g^{-1})^c (g^{2\tau_0})^{-c} =_? B_7$. *To produce a signature out of the above proof of knowledge we use the Fiat-Shamir heuristics as follows: suppose that* $\mathcal{H} : \{0,1\}^* \to \{0,1\}^k$ *is a hash function. To compute a signature* σ *for a message* M, *the signer will compute the* $B_1, \ldots, B_7$ *values as above and then compute the signature as follows:* $\sigma = \langle c, s_1, \ldots, s_7 \rangle$, *where*

$$c = \mathcal{H}(M, n, g, a, a_0, g, h, y, \hat{y}, T_1, T_2, \hat{T}_1, \hat{T}_2, T_3, B_1, \ldots, B_7)$$

and $\langle s_1, s_2, s_3, s_4, s_5, s_6, s_7\rangle = \langle s_r, s_{\hat{r}}, s_{\tilde{r}}, s_{e_i}, s_{x_i}, s_{s'}, s_{s''}\rangle$. *The verification on the signature on the other hand requires the computation of all the lefthand-sides of the verification equations performed by the verifier and the comparison with the hash c. Moreover the verifier will verify the range restrictions* $s_4 \in_? [-2^{k+l}\Delta\gamma - (2^k - 1)\Delta\gamma, 2^{k+l}\Delta\gamma]$, $s_5 \in_? [-2^{k+l}\Delta\lambda - (2^k - 1)\Delta\lambda, 2^{k+l}\Delta\lambda]$.

Lemma 22 *(1) Suppose $\mathcal{A}$ is a PPT that given the public-parameters of the system $n, g, a, a_0, h, y, \hat{y}$ produces $T_1, T_2, \hat{T}_1, \hat{T}_2, T_3$ and two accepting conversations of the proof of knowledge with the same first move but different second moves. Then, it holds that we can either solve the Strong-RSA problem or extract witnesses $r, \hat{r}, \tilde{r}, e, x', s', s''$ so that (i) $x \in [\lambda_0 - 2^{k+l+2}\Delta\lambda, \lambda_1 + 2^{k+l+2}\Delta\lambda]$ and $e \in [\gamma_0 - 2^{k+l+2}\Delta\gamma, \gamma_1 + 2^{k+l+2}\Delta\gamma]$. (ii) $T_2 = \pm g^r, \hat{T}_2 = \pm g^{\hat{r}}, T_1/\hat{T}_1 = \pm y^r/\hat{y}^{\hat{r}}, T_3 = \pm g^e h^r, T_2^e = \pm g^{s'}, a_0 a^x y^{s'} = \pm T_1^e, T_3 = \pm g(g^2)^{s''} h^r$.*

(2) Suppose $\mathcal{A}$ is a PPT that given the public-parameters of the system $n, g, a, a_0, h, y, \hat{y}$ as well as the factorization of n, produces $T_1, T_2, \hat{T}_1, \hat{T}_2, T_3$ and two accepting conversations of the proof of knowledge with the same first move but different second moves. Then, it holds that we can extract witnesses $r, \hat{r}, \tilde{r}, e, x', s', s''$ so that $T_2 = b_1 g^r, \hat{T}_2 = b_2 g^{\hat{r}}, T_1/\hat{T}_1 = \sigma_3 y^r/\hat{y}^{\hat{r}}, T_3 = b_4 g^e h^r, T_2^e = b_5 g^{s'}, a_0 a^x y^{s'} = b_6 T_1^e, T_3 = b_7 g(g^2)^{s''} h^r$, where $b_1, \ldots, b_7$ are order 2 elements in $\mathbb{Z}_n^*$.

Proof. First consider part (1). Let $B_1, \ldots, B_7, c, s_1, \ldots, s_7, c^*, s_1^*, \ldots, s_7^*$ to be the two accepting conversations with the same first move. Based on the verification equations we have the following: first, $g^{s_r} T_2^c = g^{s_r^*} T_2^{c^*}$ from which we have that $g^{s_r - s_r^*} = T_2^{c^* - c}$. Using a standard argument we conclude that either $c^* - c$ divides $s_r - s_r^*$ or we can turn $\mathcal{A}$ into a Strong-RSA solver. Thus we conclude that $T_2 = \sigma g^{\frac{s_r - s_r^*}{c^* - c}}$ where $\sigma \in \mathbb{Z}_n^*$ and is a k-bit order element. Given that $k < p', q'$ we have that σ is an order 2 elements and we conclude that under the hardness of factoring it must hold that $\sigma = \pm 1$. We set $r = \frac{s_r - s_r^*}{c^* - c}$. In a similar fashion we conclude that $g^{\hat{r}} = \hat{T}_2$ where $\hat{r} = \frac{s_{\hat{r}} - s_{\hat{r}}^*}{c^* - c}$.

Then, we proceed to $y^{s_r}/\hat{y}^{s_{\hat{r}}}(T_1/\hat{T}_1)^c = y^{s_r^*}/\hat{y}^{s_{\hat{r}}^*}(T_1/\hat{T}_1)^{c^*}$ from which we obtain: $y^{s_r - s_r^*}/\hat{y}^{s_{\hat{r}} - s_{\hat{r}}^*} = (T_1/\hat{T}_1)^{c^* - c}$. Based on the previous calculations we have that $y^{\frac{s_r - s_r^*}{c^* - c}}/\hat{y}^{\frac{s_{\hat{r}} - s_{\hat{r}}^*}{c^* - c}} = \sigma \cdot T_1/\hat{T}_1$ where σ is a k-bit order element within $\mathbb{Z}_n^*$. As before we can ensure based on the hardness of factoring that $\sigma = \pm 1$. We conclude that $T_1/\hat{T}_1 = \pm y^r/\hat{y}^{\hat{r}}$.

We proceed then to the relation of B_4, $g^{s_{e_i}} h^{s_{\tilde{r}}}(T_3 g^{-\gamma_0})^c = g^{s_{e_i}^*} h^{s_{\tilde{r}}^*}(T_3 g^{-\gamma_0})^{c^*}$ which implies that, $g^{s_{e_i} - s_{e_i}^*} h^{s_{\tilde{r}} - s_{\tilde{r}}^*} = (T_3 g^{-\gamma_0})^{c^* - c}$. It follows that under the Strong-RSA assumption we have that $g^{\frac{s_{e_i} - s_{e_i}^*}{c^* - c}} h^{\frac{s_{\tilde{r}} - s_{\tilde{r}}^*}{c^* - c}} = \sigma \cdot T_3 g^{-\gamma_0}$ from which we obtain that $g^e h^{\tilde{r}} = \pm T_3 g^{-\gamma_0}$ (assuming factoring is hard) where $e = \frac{s_{e_i} - s_{e_i}^*}{c^* - c} + \gamma_0$ and $\tilde{r} = \frac{s_{\tilde{r}} - s_{\tilde{r}}^*}{c^* - c}$. Note that by the verification of the ranges of $s_{e_i}, s_{e_i}^*$ we have that e satisfies the stated range constraints. Specifically, due to the fact that $s_{e_i}, s_{e_i}^* \in [-2^{k+l}\Delta\gamma - (2^k - 1)\Delta\gamma, 2^{k+l}\Delta\gamma]$ we obtain that $s_{e_i} - s_{e_i}^* \in [-2^{k+l+1}\Delta\gamma - 2(2^k - 1)\Delta\gamma, 2^{k+l+1}\Delta\gamma]$ and as a result we have that : $e \in [\gamma_0 - 2^{k+l+1}\Delta\gamma - 2(2^k - 1)\Delta\gamma, \gamma_0 + 2^{k+l+1}\Delta\gamma]$ which is a subset of the range $[\gamma_0 - 2^{k+l+2}\Delta\gamma, \gamma_1 + 2^{k+l+2}\Delta\gamma]$.

For the relation of B_5, from which we have $(T_2^{-1})^{s_{e_i}} g^{s_{s'}}(T_2^{\gamma_0})^c = (T_2^{-1})^{s_{e_i}^*} g^{s_{s'}^*}(T_2^{\gamma_0})^{c^*}$ which implies that $(T_2^{-1})^{s_{e_i} - s_{e_i}^*} g^{s_{s'} - s_{s'}^*} = (T_2^{\gamma_0})^{c^* - c}$; conditioning on the previous calculations we have that $g^{s_{s'} - s_{s'}^*} = (T_2^{\frac{s_{e_i} - s_{e_i}^*}{c^* - c} + \gamma_0})^{c^* - c}$ from which we obtain under the Strong-RSA assumption that $c^* - c$ must divided $s_{s'} - s_{s'}^*$ as well and setting $s' = s_{s'} - s_{s'}^*$ we have that $g^{s'} = \pm T_2^{e_i}$.

For the relation of B_6 from which we have

$$a^{s_{x_i}} y^{s_{s'}} (T_1^{-1})^{s_{e_i}} (a_0^{-1})^c (a^{-\lambda_0} T_1^{\gamma_0})^c =$$

$$= a^{s_{x_i}^*} y^{s_{s'}^*} (T_1^{-1})^{s_{e_i}^*} (a_0^{-1})^{c^*} (a^{-\lambda_0} T_1^{\gamma_0})^{c^*}$$

which implies $a^{s_{x_i}-s^*_{x_i}} y^{s_{s'}-s^*_{s'}} (T_1^{-1})^{s_{e_i}-s^*_{e_i}} = (a_0^{-1})^{c^*-c}(a^{-\lambda_0} T_1^{\gamma_0})^{c^*-c}$. From this equality and conditioning on our previous extraction we obtain that $a^{s_{x_i}-s^*_{x_i}} = (y^{-s'} a_0^{-1} a^{-\lambda_0} T_1^{e_i})^{c^*-c}$ from which we obtain that under the Strong-RSA it must be that $c^* - c$ divides $s_{x_i} - s^*_{x_i}$ as well and thus if we set $x_i = \frac{s_{x_i}-s^*_{x_i}}{c^*-c} + \lambda_0$ we obtain $a_0 a^{x_i} = \pm T_1^{e_i}/y^{s'}$. Note that $x_i \in [\lambda_0 - 2^{k+l+2}\Delta\lambda, \lambda_1 + 2^{k+l+2}\Delta\lambda]$. Finally, from the relation of B_7 we have $(g^2)^{s_{s''}} h^{s_{\tilde{r}}} (T_3 g^{-1})^c (g^{2\tau_0})^{-c} = (g^2)^{s^*_{s''}} h^{s^*_{\tilde{r}}} (T_3 g^{-1})^{c^*} (g^{2\tau_0})^{-c^*}$ which implies $(g^2)^{s_{s''}-s^*_{s''}} h^{s_{\tilde{r}}-s^*_{\tilde{r}}} = (T_3 g^{-1})^{c^*-c}(g^{-2\tau_0})^{c^*-c}$. Conditioning on previous extractions we rewrite this as $(g^2)^{s_{s''}-s^*_{s''}} = (h^{-\tilde{r}} T_3 g^{-1} g^{-2\tau_0})^{c^*-c}$ from which we obtain that under the Strong-RSA assumption it must be that $c^* - c$ divides $s_{s''} - s^*_{s''}$ and if we set $s'' = \frac{s_{s''}-s^*_{s''}}{c^*-c} + \tau_0$ we have that $T_3 = \pm g^{2s''+1} h^{\tilde{r}}$.

Regarding part (2) we proceed as in case (1) with the following modifications: when confronted with an equation of the form $g^{s_r - s^*_r} = T^{c^*-c}$ we use the fact that $c, c^* < 2^k < p', q'$ to argue that $c^* - c$ is invertible in $\mathbb{Z}^*_{p'q'}$ and thus we can compute $(c^* - c)^{-1} \bmod p'q'$. Given this we set $r = (s_r - s^*_r)(c^* - c)^{-1} (\mathrm{mod} p'q')$ as the reconstructed witness and by raising both sides of the equation to $(c^* - c)^{-1}$ we have that $g^r = b \cdot T$ where $b^2 = 1(\mathrm{mod} n)$. Using this idea the proof of the second part of the lemma is completed easily following the same plan as in part (1). $\square$

6.2 Correctness and Security of the Construction

Theorem 23 *The group signature* $\langle \mathtt{SETUP}, \mathtt{JOIN}, \mathtt{SIGN}, \mathtt{VERIFY}, \mathtt{OPEN} \rangle$ *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 $T_1, T_2, \hat{T}_1, \hat{T}_2, T_3$ she will know a witness for the discrete logarithm relation she is required to prove by setting $s' = er$ and $s'' = \frac{e-1}{2}$. Based on the completeness (which can be shown easily based on definition 21) of the proof of knowledge she can create a valid signature. Finally, regarding the opening soundness, observe that for any valid signature, the $\mathtt{OPEN}$ algorithm will recover the value $A = (T_1(T_2)^{-x})^2$ which is equal to the square of 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 $\mathtt{JOIN}$ 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 square A component. The probability that the $\mathtt{JOIN}$ dialog assigns to a user the same square A component is negligible. Indeed, if two users are assigned the same square A-value in their certificate, it must be the case that $(a_0 C)^{1/e} = \sigma (a_0 C')^{1/e'}$ where σ is an order 2 element of $\mathbb{Z}^*_n$ for a random choice of e, e' from the space $\Gamma - \{p', q'\}$ 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 $QR(n)$ (also recall that e, e' are prime numbers). $\square$

The proof of security of our scheme is naturally more involved and will be broken down into three theorems one for each security property.

Theorem 24 *(Security against misidentification attacks) For any* PPT $\mathcal{A}$ *it holds that* $\mathbf{Prob}[G^{\mathcal{A}}_{\mathsf{mis}}(1^\nu) = \top] = \mathsf{negl}(\nu)$ *assuming the Strong-RSA assumption (definition 1) in the random oracle model.*

Proof. We will assume the Strong-RSA assumption and show that the existence of a PPT misidentification attacker that succeeds with non-negligible advantage leads to a contradiction. We will use lemma 20 as a main tool for refuting the Strong-RSA. Let $n, a, a_0 \in QR(n), \Gamma, \Lambda$ be as specified in the claim and let $\langle x_i, e_i, A_i\rangle$ be K tuples such that $A_i^{e_i} = a_0 a^{x_i} (\mathrm{mod} n)$ (following the specifications of lemma 20).

Below we describe a procedure $\mathcal{P}^{\mathcal{H},\mathcal{R}}$ that employs the misidentification adversary $\mathcal{A}^{\mathcal{H}}$ and has access to the two oracles as defined in lemma 30 (note that we will not need to employ the oracle $\mathcal{R}$ in this proof). 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 $h \leftarrow_R QR(n), x, \hat{x} \leftarrow_R [\lfloor n/4 \rfloor], y = g^x, \hat{y} = g^{\hat{x}}$, and $\mathcal{S} := \langle x, \hat{x}\rangle$. 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 misidentification attack game.
- $\mathcal{Q}_{\mathsf{a-join}}$ query: based on the simulation properties of the non-adaptive drawings of random powers protocol that we employ during the `JOIN` protocol, we can assume that $\mathcal{A}$ simply submits go to the trusted party in order to obtain its certificate. $\mathcal{P}$ will simulate such trusted party and supply to the i-th `JOIN` instantiation the certificate $\langle x_i, e_i, A_i\rangle$ that $\mathcal{P}$ has as input.
- $\mathcal{Q}_{\mathsf{open}}$ 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 simply forwarding them to the $\mathcal{P}$'s own $\mathcal{H}$ oracle.

In the above fashion the simulation of $\mathcal{A}$ is completed and $\mathcal{A}$ produces a group signature $T_1, T_2, \hat{T}_1, \hat{T}_2, T_3, c, s_1, \ldots, s_7$ that opens to none of the adversarially controlled users (i.e., it opens to $\perp$). Specifically this means that $(T_2^{-x}T_1)^2 \notin A_1^2, \ldots, A_K^2$. If we call $A = T_2^{-x}T_1$ then we have that $A \neq \pm A_i \bmod n$ for all $i = 1, \ldots, K$.

Based now on lemma 30 and the soundness property of the employed proof of knowledge (lemma 22) we can obtain a PPT $\mathcal{P}'$ that under the Strong-RSA assumption, it succeeds in constructing a witness for the proof of knowledge employed in a group signature. The witness yields the values $r, \hat{r}, \tilde{r}, e, s', x', s''$ such that $a_0 a^{x'} y^{s'} = \pm T_1^e, T_3 = \pm g^e h^{\tilde{r}}, T_3 = \pm g(g^2)^{s''} h^{\tilde{r}}, T_2^e = \pm g^{s'}, T_2 = \pm g^r$. Based on these equalities we obtain that $2s' = 2e\cdot r$ (in particular, if this equality does not hold it is easy to factor n). As a result $T_1^{2e} = (a_0 a^{x'} y^{e\cdot r})^2$. From these relations we obtain that $(T_2^{-x}T_1)^{2e} = (a_0 a_1^{x'})^2$ i.e. the decryption of the ciphertext T_1, T_2 (squared) is an e-th root of the value $a_0 a_1^{x'}$ (also squared).

As a result, if $A = T_2^{-x}T_1 \bmod n$ it follows that $A^e = \pm a_0 a^{x'}$. Note that the range constraints that are required for lemma 20 are ensured by the soundness of the proof of knowledge. Finally we argue that e is indeed odd. Observe that $T_3^2 = g^{2(2s''+1)} h^{2\tilde{r}}$ and also $T_3^2 = g^{2e} h^{2\tilde{r}}$. From this we obtain that $g^{2(2s''+1)} = g^{2e}$. Given that g generates $QR(n)$, a $p'q'$ order subgroup of $\mathbb{Z}_n^*$, it follows that $e = 2s'' + 1 \bmod p'q'$. From the above it follows that $e = 2s'' + 1$ which implies that e is an odd number (otherwise $2s'' + 1 - e$ would be a multiple of $p'q'$ from which information we can factor n). We conclude by observing that the conditions of lemma 20 are all satisfied and thus the Strong-RSA assumption is violated.

□

Theorem 25 *(Security against framing attacks) For any* PPT $\mathcal{A}$ *it holds that* $\mathbf{Prob}[G_{\mathsf{fra}}^{\mathcal{A}}(1^\nu) = \top] = \mathsf{negl}(\nu)$ *assuming that the Discrete-logarithm problem is hard over the* $QR(n)$ *with known factorization (cf. definition 2), 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 framing adversary that has access to the random oracle $\mathcal{H}$.

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 $\mathcal{H}$ and to an oracle reprogramming process $\mathcal{R}$ (cf. lemma 30).

Prior to the beginning of the simulation, $\mathcal{P}$ computes two tuples $\mathcal{Y}, \mathcal{S}$ as follows: first it selects a $j \in \{1, \ldots, K\}$ at random to be used later; then, it computes, $\mathcal{Y} := \langle n, a_0, a, g, h, y, \hat{y} \rangle$ where $g, h \leftarrow_R QR(n), x, \hat{x} \leftarrow_R [p'q']$, $a_0 = a^{r_0}$ where $r_0 \in_R [p'q']$, $y = g^x, \hat{y} = g^{\hat{x}}$, and $\mathcal{S} := \langle p, q, x, \hat{x} \rangle$.

$\mathcal{P}^{\mathcal{R},\mathcal{H}}$ will simulate $\mathcal{A}^{\mathcal{H}}$. 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 indistinguishable from the answer in the actual framing 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. Suppose that this is the i-th instantiation of the query. If $i \neq j$, $\mathcal{P}$ selects $x_i \leftarrow_R \Lambda$ and submits to the adversary the value $C_i = a^{x_i}$. This must be done using the simulatability of the drawing of random powers protocol as demonstrated in [23]. On the other hand, in case $i = j$, it sets $C_j = A$. Subsequently the adversary replies by $\langle i, A_i, e_i \rangle$ so that $A_i^{e_i} = a_0 C_i$ 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. Note that $\mathcal{P}$ cannot answer this query by following the protocol due to the fact that $\mathcal{P}$ does not know the membership secret sec_i of the i-th user. In order to answer the query, $\mathcal{P}$ first forms $T_1, T_2, \hat{T}_1, \hat{T}_2, T_3$ as in the description of the actual scheme. 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. To complete the signature, the proof of knowledge $(c, s_1, \ldots, s_7)$ for the discrete-log relation set must be simulated. The proof of knowledge will be simulated by selecting a challenge c at random from $\{0,1\}^k$ as well as $s_1, \ldots, s_7$ from their respective domains and then forming the $B_1, \ldots, B_7$ values to satisfy the verification equations of definition 21. No knowledge of any witness is required for this calculation. Finally, $\mathcal{P}$ will need to reprogram the oracle $\mathcal{H}$ so that the simulation is consistent and tuple $\langle c, s_1, \ldots, s_7 \rangle$ together with $T_1, T_2, \hat{T}_1, \hat{T}_2, T_3$ becomes a signature of M. This is done by invoking the reprogramming oracle $\mathcal{R}$. Note that the entropy of the reprogramming query satisfies the requirements of lemma 30. Based on lemma 13 it follows that the statistical distance between the real signature and a simulated one as above is negligible.
- $\mathcal{H}$ queries are answered by forwarding them to $\mathcal{P}$'s own $\mathcal{H}$ oracle.

In the above fashion the simulation of $\mathcal{A}$ is completed and $\mathcal{A}$ produces a group signature

$$\langle T_1, T_2, \hat{T}_1, \hat{T}_2, T_3, c, s_1, \ldots, s_7 \rangle$$

that opens to one of the honest users. Specifically this means that $(T_2^{-x} T_1)^2 \in A_1^2, \ldots, A_K^2$. If we define $A' = T_2^{-x} T_1$ then we have that $(A')^2 = (A_{i_0})^2 \bmod n$ for some $i_0 \in \{1, \ldots, K\}$ where K is the number of users that $\mathcal{A}$ created through the $\mathcal{Q}_{\mathsf{b-join}}$ queries. If $i_0 = j$ then $\mathcal{P}$ fails otherwise it continues.

Similarly to the proof of theorem 25, it holds that $\mathcal{P}$ satisfies the requirements of lemma 30, and based on it we can produce an algorithm $\mathcal{P}'$ that produces two distinct proofs of knowledge with the same first move (and of course with the same header $T_1, T_2, \hat{T}_1, \hat{T}_2, T_3$). Based on part (2) of lemma 22 we can reconstruct the witnesses for the proof of knowledge. In particular we obtain the values $r, \hat{r}, \tilde{r}, e, x', s', s''$ such that $T_2 = b_1 g^r$, $\hat{T}_2 = b_2 g^{\hat{r}}, T_1/\hat{T}_1 = \sigma_3 y^r/\hat{y}^{\hat{r}}, T_3 = b_4 g^e h^r, T_2^e = b_5 g^{s'}, a_0 a^x y^{s'} = b_6 T_1^e, T_3 = b_7 g (g^2)^{s''} h^r$ where $b_1, \ldots, b_7$ are order 2 elements in $\mathbb{Z}_n^*$. From this we obtain the following: (1) $T_2^{2e} = g^{2s'}$ which in combination to $T_2^2 = g^{2r}$ suggests that $g^{2er} = g^{2s'}$ from which we obtain that $er = s' (\text{mod} p'q')$. As a result $T_1^{2e} = (a_0 a^x y^{re})^2$ or equivalently that $(T_1 y^{-r})^{2e} = (a_0 a^x)^2$. Now given that $y = g^x$ we obtain that $(T_1 T_2^{-x})^{2e} = (a_0 a^x)^2$ or equivalently that $(A')^{2e} = (a_0 a^x)^2$.

Now recall that $(A')^2 = (A_{i_0})^2$ and $A_{i_0} = (a_0 A)^{1/e_{i_0}}$. These equations imply that

$$(a_0 a^x)^{2/e} = (a_0 A)^{2/e_{i_0}}$$

which is equivalent to $a^{(r_0+x)e_{i_0}} = a^{r_0 e} A^e$ since $a_0, a, A \in QR(n)$ and as a result $A = a^{(r_0+x)e_{i_0}/e - r_0}$, i.e., we can compute the discrete-logarithm of A base a. □

□

Theorem 26 *(Security against anonymity-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 anonymity-attack game $G^{\mathcal{A}}_{\mathsf{anon}}$. We will describe a transformation of this adversary to a CPA adversary against the cryptosystem $\langle \mathtt{Gen}_{qr}, \mathtt{Enc}_{qr}, \mathtt{Dec}_{qr} \rangle$ following the same strategy as theorem 16.

First, following a similar argument as that of proposition 14 we can show that any procedure $\mathcal{B}$ that has access to a random oracle $\mathcal{H}$ and produces two certificates $\langle \mathsf{sec}_0, \mathsf{cert}_0, \mathsf{sec}_1, \mathsf{cert}_1 \rangle$ and then receives a group signature on an arbitrary message under either of the two membership certificates is incapable of distinguishing between real and simulated signatures. Note that the simulation of the signature is produced based on lemma 13 in a standard fashion (selecting the $s_1, \ldots, s_7$ from their respective domains and computing the $B_1, \ldots, B_7$ values in the way that they are specified in the verification equations of definition 12. In particular the statistical distance between the two games is at most $q_{\mathcal{H}} 2^{-2k} + 7 \cdot 2^{-l}$.

Next, consider $\mathcal{L}_{\mathsf{sig}}$ be the language of all valid signature "headers", i.e., the set $\mathcal{L}_{\mathsf{sig}} = \{\langle T_1, T_2, \hat{T}_1, \hat{T}_2, T_3 \rangle \mid \exists r, \hat{r}, e_i, x_i, s', s'' : T_2 = g^r, \hat{T}_2 = g^{\hat{r}}, T_1/\hat{T}_1 = y^r/\hat{y}^{\hat{r}}, T_3 = g^{e_i} h^r, T_2^{e_i} = g^{s'}, a_0 a^{x_i} y^{s'} = T_1^{e_i}, T_3 = g(g^2)^{s''} h^r\}$. Following a similar argument as that of proposition 15 we can show that any procedure $\mathcal{B}$ that has access to a random oracle $\mathcal{H}$ and produces a group signature that does not open to $\perp$ and has a header $\langle T_1, T_2, \hat{T}_1, \hat{T}_2, T_3 \rangle$ that does not belong to the language $\mathcal{L}_{\mathsf{sig}}$, has probability of success that is bounded by $2\sqrt{2} q_{\mathcal{H}} 2^{-k/2}$.

Given the above two results we continue with a similar argument as that of theorem 16. Let G_0 be the attack game $G^{\mathcal{A}}_{\mathsf{anon}}$ that is played between the adversary $\mathcal{A}$ and the interface. The first three oracles used by $\mathcal{A}$, $\mathcal{Q}_{\mathsf{pub}}, \mathcal{Q}_{\mathsf{a-join}}, \mathcal{Q}_{\mathsf{read}}$ are all easily simulatable, given the factorization of n and we will not alter their simulation throughout the proof arguments. The fourth oracle used by $\mathcal{A}$ is $\mathcal{Q}_{\mathsf{open}}$; the simulation of $\mathcal{Q}_{\mathsf{open}}$ will be modified appropriately in the following arguments.

Define G_1, a slightly modified game where all $\mathcal{Q}_{\mathsf{open}}$ queries are simulated by using the $\hat{x} = \log_g \hat{y}$ key as opposed to the $x = \log_g y$.

Clearly the games G_0, G_1 are identical unless the adversary produces some group signature σ for which it holds that $(T_1 T_2^{-x})^2 \neq (\hat{T}_1 \hat{T}_2^{-x})^2$ and at the same time $\mathtt{OPEN}(\sigma) \neq \perp$. But then observe that such σ will have a header $\langle T_1, T_2, \hat{T}_1, \hat{T}_2, T_3 \rangle \notin \mathcal{L}_{\mathsf{sig}}$ and as a result the probability of such an event would be bounded by $2\sqrt{2} q_{\mathsf{open}} q_{\mathcal{H}} 2^{-k/2}$ where q_{open} is the number of $\mathcal{Q}_{\mathsf{open}}$ queries.

Consider now the following modification to game G_1 that results in game G_2: we modify the challenge oracle so that the proof of knowledge used for the signature it is simulated as opposed to computed properly (i.e., without the knowledge of the witnesses). The statistical distance between the two games can be at most $q_{\mathcal{H}} 2^{-2k} + 7 \cdot 2^l$ as argued above.

We now produce the final modification to game G_2 to obtain game G_3: we again modify the challenge oracle so that the values $\hat{T}_1, \hat{T}_2$ are selected at random from $QR(n)$. This modification violates the consistency of the "twin" ciphertexts T_1, T_2 and $\hat{T}_1, \hat{T}_2$ but has no impact on the proof of security which is simulated per the modification of game G_2. It follows that if there is any significant statistical distance between the two games we can transform the two games into a distinguisher for $\mathsf{DDH\text{-}Comp\text{-}KF}$.

Now observe that in game G_3 we do not employ the key $x = \log_g(y)$ and moreover the challenge oracle produces entirely simulatable data except for the ciphertext T_1, T_2 that with probability $1/2$ encrypts either the A_1 or A_2 depending on which user the challenge oracle is to issue a signature on behalf of. This suggests that we can turn $\mathcal{A}$ into a PPT cpa attacker $\mathcal{B}$ against the cryptosystem $\langle \mathtt{Gen}_{qr}, \mathtt{Enc}_{qr}, \mathtt{Dec}_{qr} \rangle$ that will have the same success probability as game G_3. Given that $\mathsf{DDH\text{-}Compo\text{-}KF}$ is assumed we conclude that $\mathcal{B}$ (and thus G_3) will have success probability that is different from $1/2$ only by a negligible fraction. Putting these arguments together using the triangular inequality we obtain that the advantage of $\mathcal{A}$ in the $G^{\mathcal{A}}_{\mathsf{anon}}$ game is negligible under the $\mathsf{DDH\text{-}Compo\text{-}KF}$.

□

7 Separability: Anonymity vs. the GM

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 (weaker) notion of separability compared to what [11] considered (who considered the full disassociation of all involved parties). The extension of the present formal model to stronger notions of separability, cf. [27], is possible. Nevertheless in this case we are interested in what can be achieved without incurring *any* additional cost at our basic construction. Stronger notions of separability can be achieved nevertheless at additional costs (both in terms of communication and computation).

The syntax of a group signature with authority separability is similar to the group signature syntax as presented in definition 17 with the modifications:

Definition 27 *A group signature scheme with authority separability is a digital signature scheme comprising the following six procedures; the parties involved are the GM, the opening authority and the users.*

$\mathtt{SETUP}_{\mathsf{GM}}$*: 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.* $\mathtt{SETUP}_{\mathsf{GM}}$ *also initializes a public-state string St with two components $St_{users} = \emptyset$ and $St_{trans} = \epsilon$.*

$\mathtt{SETUP}_{\mathsf{OA}}$*: 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 $\mathcal{Y}$.

$\mathtt{JOIN}$*: The* $\mathtt{JOIN}$ *protocol is identical to that of definition 17 with the only exception J_{GM} requires only the secret key of the GM, $\mathcal{S}_{\mathsf{GM}}$.*

$\mathtt{SIGN}$*: identical to definition 17.*

$\mathtt{VERIFY}$*: identical to definition 17.*

$\mathtt{OPEN}$*: the opening algorithm is the same as in definition 17 with the exception that only the opening authority's secret-key $\mathcal{S}_{\mathsf{OA}}$ is required.*

Note that above we consider that the setup procedure for the OA acts on the public-key of the GM. While our construction below will take advantage of this syntactic condition, it is not hard in general to avoid it at the expense of extending the length of the signature by a constant amount (and thus separate the GM and OA even in the setup phase).

Correctness. Given the above minor syntactic differences, the correctness of a group-signature with separated authorities is defined in the same way as definition 18 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 (1) the anonymity-attack can be made stronger by *adding* the adversarial capability of corrupting the GM. (2) the misidentification attack can be made stronger by *adding* the adversarial capability of corrupting the OA.

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 a misidentification-attack the adversary will have additionally at its disposal the query $\mathcal{Q}_{\mathsf{keyOA}}$ (i.e., the adversary can corrupt the OA). Note that this will obviate the $\mathcal{Q}_{\mathsf{open}}$ oracle in the definition of the property.

(ii) in a framing-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)

(iii) in an anonymity attack, the adversary will be given *additional* access to the $\mathcal{Q}_{\mathsf{keyGM}}, \mathcal{Q}_{\mathsf{write}}$ queries in both phases of the attack game. This will obviate the $\mathcal{Q}_{\mathsf{a-join}}$ oracle in the definition of the property.

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

Definition 28 *A group signature scheme with separated authorities is secure if for all PPT $\mathcal{A}$ it holds that (i)* $\mathbf{Prob}[G^{\mathcal{A}}_{\mathsf{in-sep}}(1^\nu) = \top] = \mathsf{negl}(\nu)$ *as well as (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 19.

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 the two authorities, with the condition (as specified in definition 27) 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 of section 6.2 we took special care to describe the proofs in a way that the extension to separated authorities will follow immediately. Taking advantage of this, the following theorem follows easily:

Theorem 29 *The group signature with separated authorities presented above is correct and secure; in particular: (i) it is secure against misidentification-attacks under the Strong-RSA assumption in the RO model. (ii) it is secure against framing-attacks under the Discrete-Log hardness assumption over $QR(n)$ with known factorization and the RO model. (iii) it is secure against anonymity-attacks under* $\mathsf{DDH\text{-}Compo\text{-}KF}$ *in the RO model.*

Proof. The proof is based directly on the proofs of theorems 24, 25 and 26. □

References

[1] M. Abdalla, J. H. An, M. Bellare, and C. Namprempre. From identification to signatures via the fiat-shamir transform: Minimizing assumptions for security and forward-security. In L. Knudsen, editor, *Advances in Cryptology – EUROCRYPT ' 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 418–433, Amsterdam, The Netherlands, 2002. Springer.

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

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

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

[5] M. Bellare, H. Shi, and C. Zhang. Foundations of group signatures: The case of dynamic groups. In A. Menezes, editor, *CT-RSA*, volume 3376 of *Lecture Notes in Computer Science*, pages 136–153. Springer, 2005.

[6] D. 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.

[7] D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In M. Franklin, editor, *Advances in Cryptology – CRYPTO ' 2004*, Lecture Notes in Computer Science. International Association for Cryptologic Research, Springer-Verlag, 2004.

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

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

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

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

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

[13] 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.

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

[15] R. Cramer and V. Shoup. Signature schemes based on the strong RSA assumption. *ACM Transactions on Information and System Security*, 3(3):161–185, Aug. 2000.

[16] D. Dolev, C. Dwork, and M. 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.

[17] D. Dolev, C. Dwork, and M. Naor. Non-malleable cryptography. *SICOMP*, 30(2):391–437, 2000. A preliminary version appeared in 23rd STOC, 1991.

[18] A. Fiat and A. 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.

[19] P.-A. Fouque and D. 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.

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

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

[22] S. Goldwasser, S. Micali, and R. 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 Oct. 1984. IEEE.

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

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

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

[26] A. Kiayias and M. Yung. Efficient secure group signatures with dynamic joins and keeping anonymity against group managers. In *Progress in Cryptology - Mycrypt 2005, First International Conference on Cryptology in Malaysia, Kuala Lumpur, Malaysia, September 28-30, 2005, Proceedings*, volume 3715 of *Lecture Notes in Computer Science*, pages 151–170. Springer, 2005.

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

[28] K. 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.

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

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

[31] C. Rackoff and D. R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In J. 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.

[32] A. 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.

[33] 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.

A Generalized Forking Lemma

Below we present a generalized lemma that is useful in proving the security of complex signature schemes (like the ones in the present paper) in the random oracle model. The lemma is a generalized variant of Pointcheval and Stern's "forking-lemma", cf. [30]. One of the dissimilarities to this previous work is the existence of oracle reprogramming queries that substitute signing queries. The lemma as stated below has no direct cryptographic interpretation and this is the main advantage of the formulation as we want to apply it in more complex settings compared to the digital signature setting of [30].

Lemma 30 *(Generalized Forking Lemma). Consider a probabilistic* PPT $\mathcal{P}$*, a* PPT *predicate* Q *and a hash-function* $\mathcal{H}$ *with range* $\{0,1\}^k$ *assumed to be a random oracle. The predicate* Q *satisfies the* *property $Q(x) = \top \implies (x = \langle \rho_1, c, \rho_2 \rangle) \wedge (c = \mathcal{H}(\rho_1))$. $\mathcal{R}$ is a process that given $\langle t, c \rangle$ it "reprograms" $\mathcal{H}$ so that $\mathcal{H}(t) = c$ provided that t was not queried to $\mathcal{H}$ before. $\mathcal{P}$ 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 $2/2^k$ (i.e., the min-entropy of t is at least $k - 1$).*

Assume now that $\mathcal{P}^{\mathcal{H},\mathcal{R}}(\mathsf{param})$ returns output x with a probability $\alpha > 2^{-k}$ such that x satisfies the following: (i) $Q(x) = \top$ and (ii) if $x = \langle \rho_1, c, \rho_2 \rangle$ it holds that $\langle \rho_1, c \rangle$ was not queried to $\mathcal{R}$. Then, there exists a PPT *$\mathcal{P}'$ so that if $y \leftarrow \mathcal{P}'(\mathsf{param})$ it holds with probability at least $\alpha^2/4q-(q \cdot s+1) \cdot 2^{-k+1}$ that: (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) $c \neq c'$, Here q is the number of $\mathcal{H}$-queries performed by $\mathcal{P}$, and s is the number of $\mathcal{R}$ queries. 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.

Proof. First assume that no queries to $\mathcal{R}$ are made whatsoever by $\mathcal{P}$. Let Ω be the probability space for the simulation of $\mathcal{P}$, i.e., each string in Ω fixes the coin tosses for $\mathcal{P}$ as well as all answers of the random oracle $\mathcal{H}$ to the queries posed by $\mathcal{P}$ (note that we only define $\mathcal{H}$ for the queries that are posed by $\mathcal{P}$).

Let $\mathsf{SuccP} \subseteq \Omega$ be the event that $\mathcal{P}$ simulated on $\omega \in \mathsf{SuccP}$ terminates outputting x such that $Q(x) = \top$. Let $\mathsf{Que}_i \subseteq \Omega$, for $i = 0, \ldots, q$ be the event that $\mathcal{P}$ produces some x such that $Q(x) = \top$ where $x = \langle \rho_1, c, \rho_2 \rangle$ and ρ_1 was the i-th query submitted to the oracle $\mathcal{H}$ by $\mathcal{P}$; if $i = 0$ then no query on ρ_1 was ever submitted to $\mathcal{H}$. We remark that $\mathbf{Prob}[\mathsf{SuccP} \cap \mathsf{Que}_0] \leq 2^{-k}$ and thus SuccP must overlap with some of the events $\mathsf{Que}_1, \ldots, \mathsf{Que}_q$.

Consider now the probability space Ω^2 and the following algorithm $\mathcal{P}'$ operating over this space: given $(\omega, \omega') \in \Omega^2$, $\mathcal{P}'$ simulates $\mathcal{P}$ over ω; when $\mathcal{P}$ terminates and as long as some event Que_i happens with $i > 0$ then $\mathcal{P}'$ replays $\mathcal{P}$ from the point of the i-th query using the appropriate suffix from coins of the string ω' (note that ω' will also redefine the random oracle $\mathcal{H}$).

We define the event $\mathsf{SuccP}' \subseteq \Omega^2$ to be the event that both simulations of $\mathcal{P}'$ terminate successfully, i.e., $\mathcal{P}$ simulated on ω terminates in a SuccP event and at the same time it holds that Que_i for some $i > 0$; in addition SuccP and Que_i hold also true for the second simulation that follows $\mathsf{merge}_i(\omega, \omega')$ (where this function merges the prefix of ω and the suffix of ω' just at the point that the value of the i-th query to $\mathcal{H}$ is requested by $\mathcal{P}$; merge is also defined for $i = 0$).

Consider $\mathsf{Que}'_{i,j} \subseteq \Omega^2$ to be the event that $\mathcal{P}$ if simulated on ω terminates so that the event Que_i is true and $\mathcal{P}$ if simulated on $\mathsf{merge}_i(\omega, \omega')$ then the event Que_j is true. The events $\mathsf{Que}'_{i,j}$ for i, j constitute a partition of Ω^2. Based on this we obtain that

$$\mathbf{Prob}[\mathsf{SuccP}'] = \sum_{i,j} \mathbf{Prob}[\mathsf{SuccP}' \mid \mathsf{Que}'_{i,j}]\mathbf{Prob}[\mathsf{Que}'_{i,j}]$$

Let $\alpha_{i,j} = \mathbf{Prob}[\mathsf{SuccP}' \mid \mathsf{Que}'_{i,j}]$ and $\beta_{i,j} = \mathbf{Prob}[\mathsf{Que}'_{i,j}]$ It is easy to see that by definition it holds that $\alpha_{i,j} = 0$ whenever $i \neq j$ or when $i = 0$ and thus the above equation gets simplified to $\mathbf{Prob}[\mathsf{SuccP}'] = \sum_{i=1}^{q} \alpha_{i,i}\beta_{i,i}$.

Next we proceed to bound the probability $\alpha_{i,i}$ i.e., the probability of the event SuccP' in the conditional space $\mathsf{Que}'_{i,i}$. This conditional space contains the set of all coin tosses (ω, ω') that make $\mathcal{P}$ terminate the i-th query in both simulations. Let $\alpha_i = \mathbf{Prob}[\mathsf{SuccP} \mid \mathsf{Que}_{i,i'}]$. Note that $\alpha_i = \mathbf{Prob}[\mathsf{SuccP} \mid \mathsf{Que}_i]$ (since SuccP depends only on the first simulation).

Let $\Omega_{<i}$ be the prefix set of Ω up to the i-th query to $\mathcal{H}$, we define $\Omega_{<i} = \{[\omega]_{<i} \mid \omega \in \Omega\}$. For any $\hat{\omega} \in \Omega_{<i}$ we define $\mathsf{Que}_j^{\hat{\omega},i} = \{\omega' \in \Omega \mid \mathsf{merge}_i(\omega, \omega') \in \mathsf{Que}_j\}$ where ω is the bitstring that is 0 everywhere except $[\omega]_{<i} = \hat{\omega}$; below we will use this as a convention, i.e., $\hat{\omega}$ will stand for $[\omega]_{<i}$.

Now we define the event $\mathsf{SSuccP}_i \subseteq \Omega$: $\mathsf{SSuccP}_i \subseteq \mathsf{SuccP} \cap \mathsf{Que}_i$ such that $\omega \in \mathsf{SSsuccP}_i$ if and only if

$$\frac{\#\{\omega' \mid \mathsf{merge}_i(\omega,\omega') \in \mathsf{SuccP} \cap \mathsf{Que}_i\}}{\#\mathsf{Que}_i^{\hat{\omega},i}} \geq \frac{\alpha_i}{2}$$

Next we prove that $\mathbf{Prob}[\mathsf{SSuccP}_i \mid \mathsf{Que}_i] \geq \alpha_i/2$. Indeed assume for the sake of contradiction that the opposite holds true, i.e., $\#\mathsf{SSuccP}_i < s_i/2$ where $s_i = \#(\mathsf{SuccP} \cap \mathsf{Que}_i)$. Based on this we have that $\#\overline{\mathsf{SSuccP}_i} > s_i/2$. Also let $q_i = \#\mathsf{Que}_i$.

Now observe that for each $\omega \in \overline{\mathsf{SSuccP}_i}$ it holds that less than $\alpha_i \cdot q_i^{\hat{\omega}}/(2\#\Omega_{<i})$ where $q_i^{\hat{\omega}} = \#\mathsf{Que}_i^{\hat{\omega},i}$ elements of $\mathsf{SuccP} \cap \mathsf{Que}_i$ share the same "head" with ω. The set of all possible heads is $\Omega_{<i}$. It follows that the set of all elements of $\mathsf{SuccP} \cap \mathsf{Que}_i$ that shares the same head with some element of $\overline{\mathsf{SSuccP}_i}$ has cardinality less than $\alpha_i \cdot \sum_{\hat{w} \in \Omega_{<i}} q_i^{\hat{\omega}}/(2\#\Omega_{<i})$. Observe that trivially $\overline{\mathsf{SSuccP}_i}$ is a subset of this set and as a result $\#\overline{\mathsf{SSuccP}_i} \leq \alpha_i \cdot \sum_{\hat{w} \in \Omega_{<i}} q_i^{\hat{\omega}}/(2\#\Omega_{<i})$. From this we obtain that it must be $s_i/2 < s_i/q_i \cdot \sum_{\hat{w} \in \Omega_{<i}} q_i^{\hat{\omega}}/(2\#\Omega_{<i})$ which implies $\sum_{\hat{w} \in \Omega_{<i}} q_i^{\hat{\omega}} > q_i \cdot \#\Omega_{<i}$ which is easily seen to be a contradiction. It follows that $\mathbf{Prob}[\mathsf{SSuccP}_i \mid \mathsf{Que}_i] \geq \alpha_i/2$.

From the above result it can be easily seen that $\alpha_{i,i} \geq \alpha_i^2/4$ and as a result

$$\mathbf{Prob}[\mathsf{SuccP}'] \geq \frac{1}{4}\sum_{i=1}^{q} \alpha_i^2 \beta_{i,i}$$

Next we focus on the $\beta_{i,i}$ probability and its relation to $\beta_i = \mathbf{Prob}[\mathsf{Que}_i]$. Let $\mathsf{Head}_i(\hat{\omega}) \subseteq \Omega$ be the event that for a given ω it holds that $\omega_{<i} = \hat{\omega}$; clearly $\{\mathsf{Head}_i(\hat{\omega})\}_{\hat{\omega} \in \Omega_{<i}}$ is a partition of Ω. Now observe that $\mathsf{Que}_i = \sum_{\hat{\omega} \in \Omega_{<i}} \mathsf{Que}_i \cap \mathsf{Head}_i(\hat{\omega})$ and $\mathsf{Que}'_{i,i} = \#\Omega_{<i} \cdot \sum_{\hat{\omega} \in \Omega_{<i}} (\mathsf{Que}_i \cap \mathsf{Head}_i(\hat{\omega}))^2$. It follows that:

$$\beta_{i,i} = \frac{\#\Omega_{<i} \cdot \sum_{\hat{\omega} \in \Omega_{<i}} (\mathsf{Que}_i \cap \mathsf{Head}_i(\hat{\omega}))^2}{(\#\Omega)^2}$$

Next we use the inequality: $\sum_{\lambda \in \Lambda} \alpha_\lambda^2 \geq (\sum_{\lambda \in \Lambda} \alpha_\lambda)^2/\#\Lambda$ to obtain from the above that $\beta_{i,i} \geq \beta_i^2$. Applying this inequality to the bound for the probability of SuccP' we have:

$$\mathbf{Prob}[\mathsf{SuccP}'] \geq \frac{1}{4}\sum_{i=1}^{q} (\alpha_i \beta_i)^2 \geq \frac{1}{4q}(\sum_{i=1}^{q} \alpha_i \beta_i)^2$$

Recall now that $\sum_{i=0}^{q} \alpha_i \beta_i = \mathbf{Prob}[\mathsf{SuccP}]$ and as a result

$$\mathbf{Prob}[\mathsf{SuccP}'] \geq \frac{1}{4q}(\mathbf{Prob}[\mathsf{SuccP}] - \alpha_0 \mathbf{Prob}[\mathsf{Que}_0])^2$$

To complete the proof observe that $\alpha_0 \leq 2^{-k}$ and as a result

$$\mathbf{Prob}[\mathsf{SuccP}'] \geq \frac{1}{4q}(\mathbf{Prob}[\mathsf{SuccP}] - 2^{-k})^2$$

Now define RewS to be the event of Ω^2 such that SuccP happens and $c \neq c'$. It is easy to verify that $\mathbf{Prob}[\mathsf{RewS}] \geq \mathbf{Prob}[\mathsf{SuccP}]^2/4q - 2^{-k+1}$.

Next we consider the setting where $\mathcal{R}$ queries are allowed. It holds now that the procedure $\mathcal{P}$ is not considered to be successful if it outputs a reprogrammed string ρ_1, c, ρ_2. This does not affect the analysis above. However there is the possibility of jamming the the oracle $\mathcal{R}$ by reprogramming a query that has been asked before. The probability of jamming can be bounded as follows: each time a query is made a fresh t value is selected that has min-entropy $k-1$. This means that the probability that t has been asked already is $q_\ell \cdot 2^{-k+1}$ where q_ℓ is the number of queries that have been asked to $\mathcal{H}$ oracle till the ℓ-th query to $\mathcal{R}$. It follows that the probability of jamming is at most $2^{-k+1}\sum_{\ell=1}^{s} q_\ell$. In the worse case we will have $q_\ell = q$ for all $\ell = 1, \ldots, s$ and we will obtain $2^{-k+1} q \cdot s$ as the upper bound. Based on all the above we conclude that $\mathbf{Prob}[\mathsf{RewS}] \geq (\mathbf{Prob}[\mathsf{SuccP}])^2/4q - (q \cdot s + 1) \cdot 2^{-k+1}$.

$\square$