Group Encryption

Aggelos Kiayias[1], Yiannis Tsiounis[2], and Moti Yung[3]

[1] Computer Science and Engineering, University of Connecticut
Storrs, CT, USA. aggelos@cse.uconn.edu

[2] BQuotes, New York, NY, USA. yiannis@bquotes.com.

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

Abstract. We present group encryption, a new cryptographic primitive which is the encryption analogue of a group signature. It possesses similar verifiability, security and privacy properties, but whereas a group signature is useful whenever we need to conceal the source (signer) within a group of legitimate users, a group encryption is useful whenever we need to conceal a recipient (decryptor) within a group of legitimate receivers. We introduce and model the new primitive and present sufficient as well as necessary conditions for its generic implementation. We then develop an efficient novel number theoretic construction for group encryption of discrete logarithms whose complexity is independent of the group size. As part of achieving this we construct a new public-key encryption for discrete logarithms that satisfies CCA2-key-privacy and CCA2-security in the standard model (this gives the first Pailler-based system with the above two properties proven in the standard model).

Applications of group encryption include settings where a user wishes to hide her preferred trusted third party or even impose a hidden hierarchy of trusted parties while being required to assure well-formed ciphertexts, as well as oblivious storage settings where the set of retrievers need to be verifiable but the storage distribution should be oblivious to the server.

1 Introduction

Group signatures were introduced in [22] and further developed in a line of works, e.g., [23, 20, 17, 18, 11, 36, 4, 3, 14, 6, 33, 8, 16, 7, 34, 2, 43, 9, 35, 30]. In a nutshell a group signature allows a registered member of a PKI (a.k.a. a group of registered users) to issue a signature on behalf of the group so that the issuer's identity is assured to be valid but is hidden from the verifier. After its introduction, the primitive has found numerous applications.

In this work we introduce a novel cryptographic primitive that is the encryption analogue of a group signature; we call it *group encryption* (not to be confused with group-oriented cryptography as in [26, 12], which is essentially threshold cryptosystems). A group encryption scheme allows a sender to prepare a ciphertext and convince a verifier that it can be decrypted by a member of a given PKI group. As in group signature, in a group encryption there can be an opening authority that can, reveal the identity of the group member who is the recipient of the ciphertext when the appropriate circumstances are triggered. Note that group encryption provides “receiver anonymity” in the same way that group signature provides “sender anonymity.” Nevertheless, this primitive was never considered in the group-signature literature before, even though public-key encryption and signatures are typically dual primitives that have been developed in parallel in many other settings.

A Motivating Typical Scenario: In many protocols that attempt to maintain privacy/ anonymity and employ trusted parties, it has been often naturally advocated as a flexible service to allow a user to choose its recipient trustee (e.g., a trusted third party for conditionally opening the ciphertext) among a set of available authorized parties. However, the choice of a third party, while increasing flexibility, might also reveal some preference of the user, thus reducing privacy. Group encryption is motivated by such applications. As observed by Chaum [21] the fact that the trustee is hidden within a large set of trusted parties makes attempts to bribe officials harder, thus contributing to secrecy of individuals as well.

Let us investigate whether it is possible to implement the above typical scenario by employing existing primitives. The notion of key-privacy was introduced in [5] (also [31]) who showed that there exist encryption schemes where it is impossible for an adversary to distinguish what public-key has been used for the message encryption. If we attempt to use these encryption schemes, a user may make his own trustee’s public key (without even publishing this public key) and use that one for encryption, thus faking an encryption to a trustee. Note that this amounts to attacking the application, since this user’s encryption cannot be opened by any valid trustee. Key privacy for users who encrypt with their own key was given in [13], but this means that the user has to be his own trustee, which, again, is insufficient for the application above. Finally, the notion of verifiable encryption allows the sender to prove certain properties of the encrypted message (cf. e.g., [1, 12, 44, 19]). If we employ verifiable encryption for the above application, it only assures verifiability when the public key employed is known to the verifier. Knowledge of the public key employed, in turn, is an attack on the anonymity of the trustee in the above application.

Our Major Contributions. In this work, motivated by the above examples, we first contribute the definition, formalization and generic feasibility of group encryption. We then construct an efficient concrete implementation and investigate its related number theoretic properties.

– *Definition and Model.* The group encryption primitive (GE) involves a public-key encryption scheme with special properties, a group joining protocol (involving public-key certification) and a message space that may have a required structure. Besides correctness, there are three security properties that pertain to GE schemes. The first two of these properties, called *Security* and *Anonymity* protect the sender from a hostile environment that tries to either extract information about the message (security) or to extract information about who the recipient is (anonymity). We require both properties to have the strongest notion of immunity to attack, namely CCA2 [27, 41]. The third property, that we call

Soundness protects the verifier from a hostile environment in which the sender, the group manager and the recipients collude against him, so that he accepts a ciphertext (e.g., an encrypted record to be stored) that either does not have the required structure or cannot be decrypted by a registered group member.

– *Necessary and Sufficient Conditions and Generic Design.* We identify the necessary cryptographic components of a GE scheme that include: a digital signature with adaptive chosen message security, a public-key encryption scheme that satisfies both CCA2-key-privacy and CCA2-security, and zero-knowledge proofs for NP statements. Using such components we demonstrate how a generic GE scheme can be implemented and how, in turn, the scheme implies these components (where encryption is derived directly with a relatively tight reduction).

– *Efficient Design.* We design a GE scheme for the discrete logarithm relation, which is one of the most useful relations in cryptography. To this end we employ the modular design as a guide. However, in order to get an efficient scheme, we need to design, exploit and combine primitives that algebraically suit the primitive's structure so that the ciphertext and the interaction associated with it has size independent of the size of the group of potential receivers. Given the large multitude of strong security requirements the model possesses, we found the task of designing and proving the properties to be quite challenging.

– *Efficient Encryption of Discrete Logarithm with CCA2-Security and CCA2-key-privacy.* As our first step in the overall group encryption design, we point out that no existing public-key encryption scheme is suitable for designing a GE for discrete logarithm relations, since the compound set of the requirements that include verifiability, CCA2-security and CCA2-key-privacy for anonymity has not been achieved before and requires special attention. We then design a public-key encryption with CCA2 key-privacy suitable for CCA2 secure verifiable encryption of discrete-logarithms. The security of the scheme is based on the *Decisional Composite Residuosity* (DCR) assumption of [40] (and its design is motivated by earlier works of [24, 28, 19, 10]). We note that our encryption is the first Paillier-based scheme proven to satisfy key-privacy, a fact which may be of independent interest.

– *Algebraic Structure and Intractability Assumption.* A new intractability assumption is required for proving the key-privacy property of our encryption scheme: *Decisional Diffie Hellman assumption for the subgroup of square (quadratic) n-th residues* ($\mathsf{DDH}_{\mathsf{SQNR}}$). We explain why this is a natural variation of DDH over a cyclic subgroup of $\mathbb{Z}_{n^2}^*$ that has order without small prime divisors and moreover, to strengthen the claim of intractability, we prove that the DCR (which is needed for arguing the security of the scheme anyway) implies the computational Diffie Hellman (CDH) assumption in this subgroup. Note that we know of no arithmetic cyclic group without a partial discrete-log trapdoor, where CDH holds but where DDH does not and thus the assumption seems reasonable.

Applications of Group Encryption. The combination of CCA2 security of ciphertexts, CCA2 anonymity of receivers and verifiability is a strong one and supports some enhanced properties of known constructions as well as opens the door for new applications.

– **Anonymous Trusted Third Party Applications.** Many protocols such as Fair Encryption, Escrow Encryption, Group Signatures, Fair Exchange, etc. employ a trustee, namely a trusted third party who is off-line during the protocol and gets invoked in case something goes wrong. For these primitives it is expected and has been advocated that there will be a multitude of these trustees. In this case the identity of a chosen trustee may reveal certain aspects of the user, whereas the user prefers to retain her privacy. For example, imagine an "International Key Escrow" scenario where a user wants to deposit (decrypt) a key with her own national trusted representative (and needs to do this in a verifiable way). However, such a choice, if made public, may reveal the user's nationality (in violation of privacy). The new group encryption primitive enables the user to trust her own representative, but without revealing its identity, yet to assure others that indeed a designated trustee has been chosen (and not a "faked trustee"). We believe this enhanced privacy while allowing flexibility of choice of trustee is an important step forward in privacy primitives. In this new setting two models are possible for taking keys off escrow: In the first one, each trustee tries to retrieve all the keys from the available ciphertext repository, and will be successful only when the ciphertext is his to open. In the second model, there is an opening authority which can open the identity of the trustee (but not the encrypted key, due to separation of duties). The opening authority, in turn, directs the ciphertext to the chosen trustee to be decrypted. Our primitive supports both opening models. Another scenario that is similar to the above, is proxy voting where users deposit their votes encrypted under the public-key of a proxy of their choice. A proxy is a designated trustee in this case and each user may prefer (or even be required due to legislation) to hide her choice when depositing her vote. In this manner, the proxies can be called upon later, in the tallying phase, to recover the votes entrusted to them. Recall that, as motivated above when contrasting the notion of group encryption with mere key privacy or verifiable encryption, if any of the security properties of group encryption is missing, the application loses its effectiveness, and only the combination of provability (soundness), CCA2 security and CCA2 key privacy delivers the desired effect on the overall escrow system.

– **Ad-Hoc Access Structure Group Signature**. We may implement the opening authority in group encryption as a multitude of trustees and use it to encrypt a signing credential. In this way we can build a group signature where signers can organize the set of trustees to open their signature by acting on it in a predetermined order following an ad-hoc structure that is only partially revealed to the verifier (e.g., a tree or other graph). This can be achieved by cascading the group encryption primitive so that a sequence of hops (identity discoveries and transfers) will be required to recover the identity of the signer in the signature opening step. This notion generalizes "hierarchical group signatures" a primitive introduced in [43] where the trustee access structure was determined as a fixed tree. This application demonstrates the power of our primitive in organizing hidden structures of decrypting parties with CCA2 hiding and securing properties.

– **Secure Oblivious Retriever Storage**. In the area of ubiquitous computing, secure and anonymous credentials may move between computing elements (computer, mobile unit, embedded device, etc.). A user may want to pass a credential secretly and anonymously between devices (either between her own devices, or devices of her peers, all belonging to the same group). Asynchronous transfer that does not require all devices to be present at the same time requires a storage server (similar to a mail server). We can employ group encryption in implementing such a storage server safely, where it is guaranteed that (1) the server only stores valid credentials (i.e., well formed ones that can be delivered to a legitimate retriever and avoid being tricked into storing garbage); (2) the credentials are encrypted and thus the server (or anyone who may compromise it) cannot employ them; and (3) the identity of retrievers of credentials is hidden (even under active attacks, i.e. CCA2 security conditions are needed). A device reading the storage can recover its credentials by scanning the storage sequentially and being successful in decrypting the credentials directed to it (with or without the aid of an opening authority).

We note that group encryption is naturally related to the notion of "custodian-hiding verifiable encryption" that was investigated in [38, 37] and may apply in similar application scenarios. From the construction point of view, the focus of the present work is in attaining constant complexity in the group size as opposed to linear that was the case in this previous work.

2 Group Encryption: Model and Definitions

The parties involved in a GE scheme are the sender, the verifier, a group manager (GM) that manages the group of receivers and an opening authority (OA) that is capable of discovering the identity of the receiver. Formally, a GE scheme that is verifiable for a public-relation $\mathcal{R}$ is a collection of procedures and protocols that are denoted as: $\mathtt{SETUP}, \mathtt{JOIN}, \langle \mathcal{G}_{\mathsf{r}}, \mathcal{R}, \mathsf{sample}_{\mathcal{R}} \rangle, \mathtt{ENC}, \mathtt{DEC}, \mathtt{OPEN}, \langle \mathcal{P}, \mathcal{V}, \mathsf{recon} \rangle$ The functionality of the above procedures is as follows: the $\mathtt{SETUP}$ is a set of intialization procedures for the system, one for the GM, one for the OA and one to produce public-parameters (denoted by $\mathtt{SETUP}_{\mathsf{GM}}, \mathtt{SETUP}_{\mathsf{OA}}, \mathtt{SETUP}_{\mathsf{init}}$ respectively). Using their respective setup procedures, the GM and the OA will produce their public/secret-key pairs $\langle \mathsf{pk}_{\mathsf{GM}}, \mathsf{sk}_{\mathsf{GM}} \rangle$ and $\langle \mathsf{pk}_{\mathsf{OA}}, \mathsf{sk}_{\mathsf{OA}} \rangle$; $\mathtt{JOIN} = \langle \mathsf{J}_{\mathsf{user}}, \mathsf{J}_{\mathsf{GM}} \rangle$ is a protocol between a prospective group member and the GM. After an execution of a $\mathtt{JOIN}$ protocol the group member will output his public/secret-key pair $(\mathsf{pk}, \mathsf{sk})$; the new member's public-key pk along with a certificate cert will be published in the public directory $\mathsf{database}$ by the GM. We will denote by $\mathcal{L}_{pk}^{\mathsf{param}}$ the language of all valid public-keys where param is a public parameter produced by the $\mathtt{SETUP}_{\mathsf{init}}$ procedure.

To employ GE in a transaction, it is assumed that the sender (call her Alice) has obtained a pair (x, w) that is sampled according to the procedure $\mathsf{sample}_{\mathcal{R}}(\mathsf{pk}_{\mathcal{R}}, \mathsf{sk}_{\mathcal{R}})$, where $\mathsf{pk}_{\mathcal{R}}, \mathsf{sk}_{\mathcal{R}}$ are produced by the generation procedure $\mathcal{G}_{\mathsf{r}}(1^{\nu})$ that samples the public/secret parameters for the relation $\mathcal{R}$. We remark that the secret-parameter $\mathsf{sk}_{\mathcal{R}}$ may be empty depending on the relation (e.g., in the case of discrete logarithm the relation is typically publicly samplable, hence $\mathsf{sk}_{\mathcal{R}}$ is empty – but this is not be the case in general). The polynomial-time testing procedure $\mathcal{R}(x, w)$ returns true iff (x, w) belongs to the relation based on the public-parameter $\mathsf{pk}_{\mathcal{R}}$. We note that given the relation $\mathcal{R}(\cdot,\cdot)$ it will be useful that it is hard to extract a "witness" w given an instance x; however this need not be included in the formal requirements for a GE scheme. Note that if verifiability is not desired from the GE, the relation $\mathcal{R}$ will be set to be the trivial relation that includes any string of a fixed size as a witness (and in such case x will be simply equal to $1^{|w|}$).

Alice possessing the pair (x, w), she wishes to encrypt w for her chosen receiver, call him Bob. She obtains Bob's certified public-key $\langle \mathsf{pk}, \mathsf{cert}\rangle$ from $\mathsf{database}$, and employing the public-keys $\mathsf{pk}_{\mathsf{GM}}$ and $\mathsf{pk}_{\mathsf{OA}}$ she encrypts w as $\mathtt{ENC}(\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}, w, L)$ to obtain the ciphertext ψ with a certain label L (L is a public string bound to the ciphertext that may contain some transaction related data or be empty; we call it the "context" of ψ). Alice will give x, ψ, L to the verifier. Subsequently, Alice and the verifier will engage in the proof of knowledge $\langle \mathcal{P}, \mathcal{V}\rangle$ that will ensure the following regarding the ciphertext ψ and label L: there exists a group member whose key is registered in the database (i.e., Bob in this case) that is capable of decrypting ψ in context L and obtaining a value w' for which it holds that if $w \leftarrow \mathsf{recon}(w')$ we have that $(x, w) \in \mathcal{R}$. Note that, for $\mathcal{P}, \mathcal{V}$, the input to the verifier will be the values $\mathsf{param}, \mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}_{\mathcal{R}}, x, \psi, L$, whereas the prover (Alice) will have as additional input the values $\mathsf{pk}, \mathsf{cert}, w$ as well as the coin tosses used for the formation of ψ. The function $\mathsf{recon}(\cdot)$ reconstructs a witness based on the decryption of ψ and may be the identity function.

In the remaining of the section we give four definitions, correctness and the three security related properties of GE, security, anonymity, and soundness. For simulating two-party protocols we use the following notation: $\langle \mathsf{output}_A \mid \mathsf{output}_B\rangle \leftarrow \langle A(\mathsf{input}_A),\ B(\mathsf{input}_B)\rangle(\mathsf{common\_input})$.

Definition 1. (Correctness) *A GE scheme is correct if the following "correctness game" returns 1 with overwhelming probability.*

1. $\mathsf{param} \leftarrow \mathtt{SETUP}_{\mathsf{init}}(1^\nu)$; $\langle \mathsf{pk}_{\mathcal{R}}, \mathsf{sk}_{\mathcal{R}}\rangle \leftarrow \mathcal{G}_{\mathsf{r}}(1^\nu)$; $(x, w) \leftarrow \mathsf{sample}_{\mathcal{R}}(\mathsf{pk}_{\mathcal{R}}, \mathsf{sk}_{\mathcal{R}})$.
2. $\langle \mathsf{pk}_{\mathsf{GM}}, \mathsf{sk}_{\mathsf{GM}}\rangle \leftarrow \mathtt{SETUP}_{\mathsf{GM}}(\mathsf{param})$; $\langle \mathsf{pk}_{\mathsf{OA}}, \mathsf{sk}_{\mathsf{OA}}\rangle \leftarrow \mathtt{SETUP}_{\mathsf{OA}}(\mathsf{param})$;
3. $\langle \mathsf{pk}, \mathsf{sk}, \mathsf{cert} \mid \mathsf{pk}, \mathsf{cert}\rangle \leftarrow \langle \mathsf{J}_{\mathsf{user}}, \mathsf{J}_{\mathsf{GM}}(\mathsf{sk}_{\mathsf{GM}})\rangle(\mathsf{pk}_{\mathsf{GM}})$. If $\mathsf{pk} \notin \mathcal{L}^{\mathsf{param}}_{pk}$ then abort;
4. $\psi \leftarrow \mathtt{ENC}(\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}, \mathsf{cert}, w, L)$.
5. $\mathsf{out}_1 \leftarrow w \stackrel{?}{=} \mathsf{recon}(\mathtt{DEC}(\mathsf{sk}, \psi, L))$.
6. $\mathsf{out}_2 \leftarrow \mathsf{pk} \stackrel{?}{=} \mathtt{OPEN}(\mathsf{sk}_{\mathsf{OA}}, [\psi]_{\mathsf{oa}}, L)$.
7. $\langle \mathsf{done} \mid \mathsf{out}_3\rangle \leftarrow \langle \mathcal{P}(w, \psi, coins_\psi), \mathcal{V}\rangle(\mathsf{param}, \mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}_{\mathcal{R}}, x, \psi, L)$.
8. if $(\mathsf{out}_1 = \mathsf{out}_2 = \mathsf{out}_3 = \mathsf{true})$ return 1.

As shown above the opening procedure $\mathtt{OPEN}$ may not operate on the ciphertext ψ but on a substring of the ciphertext ψ that is denoted by $[\psi]_{\mathsf{oa}}$; we make the distinction explicit as it is relevant in terms of chosen ciphertext security.

There are three "security notions" for GE schemes: security, anonymity and soundness (that includes verifiability). Security and anonymity are properties that protect Alice (the prover) against a system that acts against her.

Formulation of the Security Property. In our definitions we use a number of traditional oracles that express the nature of the interaction of the adversary and the system. Accordingly, we employ oracles that are stateless (those that maintain no state across queries) and those that are stateful. Next, we introduce the decryption oracle, the challenge procedures and the prover simulator oracle.

$\mathsf{DEC}(\mathsf{sk}, \cdot)$: This is a decryption oracle for the GE decryption function $\mathtt{DEC}$. The value sk is a secret-key that will be clarified from the context. If ψ is some "forbidden" ciphertext with label L that the oracle must reject we will write $\mathsf{DEC}^{\neg\langle\psi,L\rangle}(\mathsf{sk}, \cdot)$.

$\mathsf{CH}^b_{\mathsf{ror}}(1^\nu, \mathsf{pk}, w, L)$: This a real-or-random challenge procedure for the GE encryption scheme. It returns two values denoted as $\langle\psi, coins_\psi\rangle$ so that if $b = 1$ then $\psi \leftarrow \mathtt{ENC}(\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}, \mathsf{cert}, w, L)$, whereas if $b = 0$, $\psi \leftarrow \mathtt{ENC}(\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}, \mathsf{cert}, w', L)$ where w' is a plaintext sampled at random from the space of all possible plaintexts of length 1^ν for the encryption function (it is assumed at least two plaintexts exist). In either case $coins_\psi$ are the random coin tosses that are used for the computation of ψ.

$\mathsf{PROVE}^b_{\mathcal{P},\mathcal{P}'}(\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}, \mathsf{cert}, \mathsf{pk}_{\mathcal{R}}, x, w, \psi, L, coins_\psi)$: this is an oracle that if $b = 1$, it simulates an execution of the prover procedure of $\mathcal{P}$ of the GE scheme (i.e., Alice), on $\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}, \mathsf{cert}$, $\mathsf{pk}_{\mathcal{R}}, x, w, \psi, L, coins_\psi$. On the other hand, if $b = 0$, it simulates the protocol $\mathcal{P}'$ that takes the same input as $\mathcal{P}$ with the exception of the values of w and $coins_\psi$ (the design of $\mathcal{P}'$ is part of proving the security property).

Based on the above three procedures we are ready to give the security definition, which is reminiscent of a real-or-random attack on the underlying encryption scheme. In the game below the adversary controls the GM and OA and all group members except the member that Alice chooses as her recipient, i.e., Bob. In fact, the adversary is the entity that introduces Bob into the group and issues a certificate for his public-key. Moreover, the adversary has CCA2 access to Bob's secret-key. The adversary also selects some public relation $\mathcal{R}$ based on $\mathsf{pk}_{\mathcal{R}}$ as well as a pair (x, w). Subsequently a coin is tossed and the adversary either receives the encryption of w and engages with Alice in the proof of ciphertext validity or the adversary receives an encryption of a random plaintext and engages in a simulated proof of validity. A GE would satisfy security if the adversary is unable to tell the difference. More formally (note that $\mathsf{negl}(\nu)$ is a function that for any c, is less than ν^{-c} for sufficiently large ν):

Definition 2. *A GE scheme satisfies* security *if there exists a protocol $\mathcal{P}'$ s.t. the "security game" below when instantiated by any PPT $\mathcal{A}$, returns 1 with probability less or equal to $1/2 + \mathsf{negl}(\nu)$.*

1. $\mathsf{param} \leftarrow \mathtt{SETUP}_{\mathsf{init}}(1^\nu)$; $\langle \mathsf{aux}, \mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}\rangle \leftarrow \mathcal{A}(\mathsf{param})$;
2. $\langle \mathsf{pk}, \mathsf{sk}, \mathsf{cert} \mid \mathsf{aux}\rangle \leftarrow \langle \mathsf{J}_{\mathsf{user}}, \mathcal{A}(\mathsf{aux})\rangle(\mathsf{pk}_{\mathsf{GM}})$;
3. $\langle \mathsf{aux}, x, w, L, \mathsf{pk}_{\mathcal{R}}\rangle \leftarrow \mathcal{A}^{\mathsf{DEC}(\mathsf{sk},\cdot)}(\mathsf{aux})$; *if* $(x, w) \notin \mathcal{R}$ *then abort;*

4. $b \xleftarrow{r} \{0,1\}$; $\langle \psi, coins_\psi \rangle \leftarrow \mathsf{CH}^b_{\mathsf{ror}}(1^\nu, \mathsf{pk}, w, L)$;
5. $b^* \leftarrow \mathcal{A}^{\mathsf{PROVE}^b_{\mathcal{P},\mathcal{P}'}(\mathsf{pk}_{\mathsf{GM}},\mathsf{pk}_{\mathsf{OA}},\mathsf{pk},\mathsf{cert},\mathsf{pk}_{\mathcal{R}},x,w,\psi,L,coins_\psi),\mathsf{DEC}^{\neg\langle\psi,L\rangle}(\mathsf{sk},\cdot)}(\mathsf{aux},\psi)$
6. *if* $b = b^*$ *return 1 else 0.*

Formulation of the Anonymity Property. In the anonymity attack the adversary controls the system except the opening authority. Anonymity can be thought of as a CCA2 attack against the encryption system of the OA. The adversary registers the two possible recipients into the PKI database and provides the relation and the witness to Alice. Alice will encrypt the same witness always as provided by the adversary but will use the key of one of the two recipients at random. The adversary, who has CCA2 decryption access to both recipients as well as the OA, will have to guess which one of the two is Alice's choice. We define the following procedures:

$\mathsf{CH}^b_{\mathsf{anon}}(\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}_0, \mathsf{pk}_1, w, L)$: The challenge procedure receives a plaintext w and two public-keys $\mathsf{pk}_0, \mathsf{pk}_1$, and returns two values, $\langle \psi, coins_\psi \rangle$ so that $\psi \leftarrow \mathtt{ENC}(\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}_b, \mathsf{cert}_b, w, L)$ and $coins_\psi$ are the random coin tosses that are used for the computation of ψ.

$\mathsf{USER}(\mathsf{pk}_{\mathsf{GM}})$: this is an oracle that simulates two instantiations of $\mathsf{J}_{\mathsf{user}}$, i.e., it is given $\mathsf{pk}_{\mathsf{GM}}$ and simulates two users that wish to become members of the group; the oracle has access to a string denoted by keys in which USER will write the output of the two $\mathsf{J}_{\mathsf{user}}$ instances.

$\mathsf{OPEN}(\mathsf{sk}_{\mathsf{OA}}, \cdot)$: this is an oracle that simulates the $\mathtt{OPEN}$ operation of the opening authority; recall that $\mathtt{OPEN}$ may not operate on the whole ciphertext ψ but rather on substring of it that will be denoted by $[\psi]_{\mathsf{oa}}$.

Definition 3. *A* GE *scheme satisfies* anonymity *if the following game instantiated for any PPT $\mathcal{A}$, it returns 1 with probability less or equal* $1/2 + \mathsf{negl}(\nu)$.

1. $\mathsf{param} \leftarrow \mathtt{SETUP}_{\mathsf{init}}(1^\nu)$; $\langle \mathsf{pk}_{\mathsf{OA}}, \mathsf{sk}_{\mathsf{OA}} \rangle \leftarrow \mathtt{SETUP}_{\mathsf{OA}}(\mathsf{param})$;
2. $\langle \mathsf{pk}_{\mathsf{GM}}, \mathsf{sk}_{\mathsf{GM}} \rangle \leftarrow \mathtt{SETUP}_{\mathsf{GM}}(\mathsf{param})$; $\mathsf{aux} \leftarrow \mathcal{A}^{\mathsf{USER}(\mathsf{pk}_{\mathsf{GM}}),\mathsf{OPEN}(\mathsf{sk}_{\mathsf{OA}},\cdot)}(\mathsf{sk}_{\mathsf{GM}})$;
3. *if* $\mathsf{keys} \neq \langle \mathsf{pk}_0, \mathsf{sk}_0, \mathsf{cert}_0, \mathsf{pk}_1, \mathsf{sk}_1, \mathsf{cert}_1 \rangle$ *then abort;*
4. $\langle \mathsf{aux}, x, w, L, \mathsf{pk}_{\mathcal{R}} \rangle \leftarrow \mathcal{A}^{\mathsf{OPEN}(\mathsf{sk}_{\mathsf{OA}},\cdot),\mathsf{DEC}(\mathsf{sk}_0,\cdot),\mathsf{DEC}(\mathsf{sk}_1,\cdot)}(\mathsf{aux})$;
5. *if* $(x, w) \notin \mathcal{R}$ *then abort;* $b \xleftarrow{r} \{0,1\}$;
6. $\langle \psi, coins_\psi \rangle \leftarrow \mathsf{CH}^b_{\mathsf{anon}}(\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}_0, \mathsf{pk}_1, w, L)$;
7. $t_b \leftarrow \langle \mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}_{\mathcal{R}}, \mathsf{pk}_b, \mathsf{cert}_b, x, w, \psi, L, coins_\psi \rangle$;
8. $b^* \leftarrow \mathcal{A}^{\mathcal{P}(t_b),\mathsf{OPEN}^{\neg\langle[\psi]_{\mathsf{oa}},L\rangle}(\mathsf{sk}_{\mathsf{OA}},\cdot),\mathsf{DEC}^{\neg\langle\psi,L\rangle}(\mathsf{sk}_0,\cdot),\mathsf{DEC}^{\neg\langle\psi,L\rangle}(\mathsf{sk}_1,\cdot)}(\mathsf{aux},\psi)$;
9. *if* $b = b^*$ *return 1 else 0;*

This completes the security definition as far as Alice is concerned. From the point of view of the verifier, the goal of a malicious environment in which the verifier operates is to provide him with a ciphertext that encrypts a witness for a public relation that does not open to a witness even if all the group members apply their decryption function to it. Immunity to this attack, which we call soundness, guarantees that at least one group key will open to a valid witness.

Formulation of the Soundness Property. A soundness attack proceeds as follows: the adversary will create adaptively the group of recipients communicating with the GM. In this attack game, the adversary wins if, while playing the role of Alice, she convinces the verifier that a ciphertext is valid with respect to a public-relation $\mathcal{R}$ of the adversary's choice, but it holds that either (1) if the opening authority applies $\mathsf{sk}_{\mathsf{OA}}$ to the ciphertext the result is a value that is not equal to a public-key of any group member, or (2) the revealed key satisfies $\mathsf{pk} \notin \mathcal{L}_{pk}^{\mathsf{param}}$. To formalize soundness we introduce the following group registration oracle:

$\mathsf{REG}(\mathsf{sik}, \cdot)$: this is an oracle that simulates J_{GM}, i.e., it is given $\mathsf{sk}_{\mathsf{GM}}$ and registers users in the group; the oracle has access to a string $\mathsf{database}$ that stores the public-keys and their certificates.

Definition 4. *A* GE *scheme satisfies* soundness *if the following "soundness game", when instantiated with any PPT adversary $\mathcal{A}$, the probability it returns 1 is negligible.*

1. $\mathsf{param} \leftarrow \mathtt{SETUP}_{\mathsf{init}}(1^{\nu})$; $\langle \mathsf{pk}_{\mathsf{OA}}, \mathsf{sk}_{\mathsf{OA}} \rangle \leftarrow \mathtt{SETUP}_{\mathsf{OA}}(\mathsf{param})$;
2. $\langle \mathsf{pk}_{\mathsf{GM}}, \mathsf{sk}_{\mathsf{GM}} \rangle \leftarrow \mathtt{SETUP}_{\mathsf{GM}}(\mathsf{param})$;
3. $\langle \mathsf{pk}_{\mathcal{R}}, x, \psi, L, \mathsf{aux} \rangle \leftarrow \mathcal{A}^{\mathsf{REG}(\mathsf{sk}_{\mathsf{GM}}, \cdot)}(\mathsf{param}, \mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{sk}_{\mathsf{OA}})$;
4. $\langle \mathsf{aux}, \mathsf{out} \rangle \leftarrow \langle \mathcal{A}(\mathsf{aux}), \mathcal{V} \rangle(\mathsf{param}, \mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}_{\mathcal{R}}, x, \psi, L)$;
5. $\mathsf{pk} \leftarrow \mathtt{OPEN}(\mathsf{sk}_{\mathsf{OA}}, [\psi]_{\mathsf{oa}}, L)$;
6. if $\mathsf{pk} \notin \mathsf{database}$ or $\mathsf{pk} \notin \mathcal{L}_{pk}^{\mathsf{param}}$ or $\psi \notin \mathcal{L}_{ciphertext}^{x, L, \mathsf{pk}_{\mathcal{R}}, \mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}}$ then ret. 1 else 0;

Note that $\mathcal{L}_{ciphertext}^{x, L, \mathsf{pk}_{\mathcal{R}}, \mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}} = \{\mathtt{ENC}(\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}, \mathsf{cert}, w, L) \mid w : (x, w) \in \mathcal{R}, \langle \mathsf{pk}, \mathsf{cert} \rangle \in \mathsf{Valid}\}$. This means that the soundness adversary wins if the key obtained by OA after opening is either not in the database, or is invalid, or the ciphertext ψ is not a valid ciphertext under pk encrypting a witness for x under $\mathcal{R}$.

A GE scheme should satisfy correctness, security, anonymity and soundness. Note that: (1) By defining the oracles USER and REG one can allow concurrent attacks or force sequential execution of the group registration process. (2) CPA variants of the security and anonymity definition w.r.t. either group members or the OA can be obtained by dropping the corresponding DEC oracles. (3) Soundness and security assume a trusted setup; extension to malicious setup can be done by enforcing trustworthy initialization by standard methods (e.g. threshold cryptography or ZK proofs).

3 Necessary and Sufficient Conditions for GE schemes

Given that a GE scheme is a complex primitive it would be helpful to break down its construction to more basic primitives and provide a general methodology for constructing GE schemes. The necessary components for building a GE scheme will be the following:

1. Adaptively Chosen Message Secure Digital Signature. It will be used to generate the public-key certificates by the GM during the `JOIN` procedure.

2. Public-key Encryption with CCA2 Security and Key-Privacy. We will employ an encryption scheme $\langle \mathcal{G}_{\mathsf{e}}, \mathcal{E}, \mathcal{D} \rangle$ that satisfies (1) CCA2-security and (2)

CCA2-Key-privacy. We note that in public-key encryption with key-privacy the key-generation has two components, one called $\mathcal{Z}_{\mathsf{e}}$ that produces public-parameters shared by all key-holders and the key-generation $\mathcal{G}_{\mathsf{e}}$ that given the public-parameter of the system produces a public/secret-key pair. Note that using $\mathcal{Z}_{\mathsf{e}}$ is mandatory since some agreement between the receivers is necessary for key-privacy (at minimum all users should employ public-keys of the same length).

3. Proofs of Knowledge. Such protocols in the zero-knowledge setting satisfy three properties: completeness, soundness with knowledge extraction and zero-knowledge. These proofs exist for any NP language assuming one-way functions by reduction, e.g., to the graph 3-colorability proof of knowledge [29]. In certain settings, zero-knowledge proofs can be constructed more efficiently by starting with a honest-verifier zero-knowledge (HVZK) proof of language membership protocol (i.e., a protocol that requires no knowledge extraction and it is only zero-knowledge against honest verifiers) and then coupling such protocol with an extractable commitment scheme (to achieve knowledge extraction) and with an equivocal commitment (to enforce zero-knowledge against dishonest verifiers, cf. [25]).

Modular Design of GE schemes. Consider an arbitrary relation $\mathcal{R}$ that has an associated paramter generation procedure $\mathcal{G}_{\mathsf{r}}$ and a witness sampler $\mathsf{sample}_{\mathcal{R}}$. In the modular construction we will employ: (1) a digital signature scheme $\langle \mathcal{G}_{\mathsf{s}}, \mathcal{S}, \mathcal{V}_{\mathsf{s}} \rangle$ that is adaptively chosen message secure; (2) a public-key encryption scheme $\langle \mathcal{Z}_{\mathsf{e}}, \mathcal{G}_{\mathsf{e}}, \mathcal{E}, \mathcal{D} \rangle$ that satisfies CCA2 security and Key-privacy; (3) two zero-knowledge proofs of language membership (defined below); to facilitate knowledge extraction we will employ also an extractable commitment scheme $\langle \mathcal{Z}_{\mathsf{c},1}, \mathcal{C}_1, \mathcal{T}_1 \rangle$. Without loss of generality we will assume that all employed primitives operate over bitstrings. The construction of a GE scheme $\langle \mathtt{SETUP}, \mathtt{JOIN}, \langle \mathcal{G}_{\mathsf{r}}, \mathcal{R}, \mathsf{sample}_{\mathcal{R}} \rangle, \mathtt{ENC}, \mathtt{DEC}, \mathtt{OPEN}, \langle \mathcal{P}, \mathcal{V} \rangle, \mathsf{recon} \rangle$ is as follows:

$\mathtt{SETUP}$. The $\mathtt{SETUP}_{\mathsf{init}}$ procedure will select the parameters param by performing a sequential execution of $\mathcal{Z}_{\mathsf{e}}, \mathcal{Z}_{\mathsf{c},1}$. The $\mathtt{SETUP}_{\mathsf{GM}}$ procedure will be the signature-setup $\mathcal{G}_{\mathsf{s}}$ and the $\mathtt{SETUP}_{\mathsf{OA}}$ will be the encryption-setup $\mathcal{G}_{\mathsf{e}}$.

$\mathtt{JOIN}$. Each prospective user will execute $\mathcal{G}_{\mathsf{e}}$ to obtain pk, sk and then engage in a protocol $\langle \mathcal{P}_{pk}, \mathcal{V}_{pk} \rangle$ which is proof of language membership with the GM for the language $\mathcal{L}_{pk}^{\mathsf{param}} = \{\mathsf{pk} \mid \exists \mathsf{sk}, \rho : \langle \mathsf{pk}, \mathsf{sk} \rangle \leftarrow \mathcal{G}_{\mathsf{e}}(\mathsf{param}; \rho)\}$. The GM will respond with the signature $\mathsf{cert} \leftarrow \mathcal{S}(\mathsf{sk}_{\mathsf{GM}}, \mathsf{pk})$.

$\mathtt{ENC}$. The procedure $\mathtt{ENC}$, given a witness w for a value x such that $(x, w) \in \mathcal{R}$ and a label L, it will return the pair $\psi =_{\mathsf{df}} \langle \psi_1, \psi_2, \psi_3, \psi_4 \rangle$ where $\psi_1 \leftarrow \mathcal{E}(\mathsf{pk}, w, L_1)$, $\psi_2 \leftarrow \mathcal{E}(\mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}, L_2)$, $\psi_3 \leftarrow \mathcal{C}_1(\mathsf{cpk}, \mathsf{pk})$ $\psi_4 \leftarrow \mathcal{C}_1(\mathsf{cpk}, \mathsf{cert})$ where $L_1 = \psi_2 || \psi_3 || \psi_4 || L$ and $L_2 = \psi_3 || \psi_4 || L$.

$\mathtt{DEC}$. Given sk, a ciphertext $\langle \psi_1, \psi_2, \psi_3, \psi_4 \rangle$ and a label L, it will return $\mathcal{D}(\mathsf{sk}, \psi_1, \psi_2 || \psi_3 || \psi_4 || L)$.

$\mathtt{OPEN}$. Given $\mathsf{sk}_{\mathsf{OA}}$, a ciphertext $\langle \psi_2, \psi_3, \psi_4 \rangle =_{\mathsf{df}} [\psi]_{\mathsf{oa}}$ and a label L it will return $\mathcal{D}(\mathsf{sk}_{\mathsf{OA}}, \psi_2, \psi_3 || \psi_4 || L)$.

Finally, the protocol $\langle \mathcal{P}, \mathcal{V} \rangle$ is a zero-knowledge proof of language membership for the language:

$$\{\langle \mathsf{param}, \mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}_{\mathcal{R}}, x, \psi_1, \psi_2, \psi_3, \psi_4, L\rangle \mid \exists\, (coins_{\psi_1}, coins_{\psi_2},$$

$$coins_{\psi_3}, coins_{\psi_4}, \mathsf{pk}, \mathsf{cert}, w) :$$

$$\wedge(\mathcal{C}_1(\mathsf{cpk}, \mathsf{pk}; coins_{\psi_3}) = \psi_3) \wedge (\mathcal{C}_1(\mathsf{cpk}, \mathsf{cert}; coins_{\psi_4}) = \psi_4) \wedge (\mathcal{V}_{\mathsf{s}}(\mathsf{pk}, \mathsf{cert}) = \mathsf{true})$$

$$\wedge(\mathcal{E}(\mathsf{pk}, w, (\psi_2||\psi_3||\psi_4||L); coins_{\psi_1}) = \psi_1)$$

$$\wedge(\mathcal{E}(\mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}, (\psi_3||\psi_4||L); coins_{\psi_2}) = \psi_2) \wedge ((x, w) \in \mathcal{R})$$

Note that the reconstruction procedure recon will be set to simply the identity function.

Theorem 1. *The* GE *scheme above satisfies (i)* Correctness*, given that all involved primitives are correct and $\langle \mathcal{P}_{pk}, \mathcal{V}_{pk} \rangle, \langle \mathcal{P}, \mathcal{V} \rangle$ satisfy completeness. (ii)* Anonymity*, given that the encryption scheme for users satisfies CCA2-key-privacy, the encryption scheme for OA satisfies CCA2-security, the commitment scheme $\mathcal{C}_1$ is hiding and $\langle \mathcal{P}_{pk}, \mathcal{V}_{pk} \rangle$ and $\langle \mathcal{P}, \mathcal{V} \rangle$ are zero-knowledge. (iii)* Security*, given that the employed encryption scheme for users satisfies CCA2-security, the commitment scheme $\mathcal{C}_1$ is hiding and $\langle \mathcal{P}_{pk}, \mathcal{V}_{pk} \rangle$, $\langle \mathcal{P}, \mathcal{V} \rangle$ are zero-knowledge. (iv)* Soundness*, given that the employed digital signature scheme satisfies adaptive chosen message security, the commitment scheme $\mathcal{C}_1$ is binding and extractable and $\langle \mathcal{P}_{pk}, \mathcal{V}_{pk} \rangle$ and $\langle \mathcal{P}, \mathcal{V} \rangle$ satisfy soundness.*

Necessity of the basic primitives. We consider the reverse of the above results: the existence of GE would imply public-key encryption that is CCA2 secure and private as well as digital signature and zero-knowledge proofs for any NP-language. More details are given in the full version [32].

4 Efficient GE of Discrete-Logarithms

In this section we will consider the discrete-logarithm relation $\langle \mathcal{G}_{\mathsf{dl}}, \mathcal{R}_{\mathsf{dl}}, \mathsf{sample}_{\mathsf{dl}} \rangle$: $\mathcal{G}_{\mathsf{dl}}$ given 1^ν samples a description of a cyclic group of ν-bits order and a generator γ of that group; $\mathcal{R}$ contains pairs of the form (x, w) where $x = \gamma^w$; note that $\mathsf{pk}_{\mathcal{R}} = \langle \mathrm{desc}(G), \gamma \rangle$ and $\mathsf{sk}_{\mathcal{R}}$ is empty. Finally $\mathsf{sample}_{\mathsf{dl}}$ on input $\mathsf{pk}_{\mathcal{R}}$ selects a witness w and returns the pair $(x = \gamma^w, w)$. In this section we will present a GE scheme for the above relation. Note that the results of this section can be easily extended to other relations based on discrete-logs such as a commitment to w.

Design of a public-key encryption for discrete-logarithms with key-privacy and security. One of the hurdles in designing a GE for discrete-logarithms is finding a suitable encryption scheme for the group members. In this section we will present a public-key encryption scheme that is suitable for verifiable encryption of discrete-logarithms while it satisfies CCA2-key-privacy and CCA2-security. The scheme is related to previous public-key encryption schemes of [24, 40, 28, 19, 10] and it is the first Paillier-based public-key encryption that is proven to satisfy key-privacy and security against chosen ciphertext attacks. Below we give a detailed description of our public-key encryption $\langle \mathcal{Z}_{\mathsf{e}}, \mathcal{G}_{\mathsf{e}}, \mathcal{E}, \mathcal{D}\rangle$ and of the accompanying intractability assumptions that ensure its properties.

Public-parameters. The parameter selection function $\mathcal{Z}_{\mathsf{e}}$, given 1^ν selects a composite modulus $n = pq$ so that n is a ν-bit number, $p = 2p' + 1, q = 2q' + 1$ and p, p', q, q' are all prime numbers with p, q of equal size at least $\lfloor \nu/2 \rfloor + 1$. Then it samples $g \leftarrow \mathbb{Z}^*_{n^2}$ and computes $g_1 \leftarrow g^{2n} (\mathrm{mod} n^2)$. Observe that $\langle g_1 \rangle$ with very high probability is a subgroup of order $p'q'$ within $\mathbb{Z}^*_{n^2}$. In such case $\langle g_1 \rangle$ is a group that contains all square n-th residues of $\mathbb{Z}^*_{n^2}$ and we will call this group $\mathcal{X}_{n^2}$. We note further that all elements of $\mathbb{Z}^*_{n^2}$ can be written in a unique way in the form $g_1^r (1+n)^v (-1)^\alpha (p_2 p - q_2 q)^\beta$ where $r \in [p'q'], v \in [n], \alpha, \beta \in \{0, 1\}$ (in this decomposition, p_2, q_2 are integers that satisfy $p_2 p^2 \equiv_{q^2} 1, q_2 q^2 \equiv_{p^2} 1$). We will denote by $\mathcal{Q}_{n^2}$ the subgroup of quadratic residues modulo n^2 which can be easily seen to contain all elements of the form $g_1^r (1+n)^v$ with $r \in \mathbb{Z}_{p'q'}$ and $v \in \mathbb{Z}_n$ and has order $np'q'$ (precisely one fourth of $\mathbb{Z}^*_{n^2}$ and is generated by $g_1(1+n)$). Note that we will use the notation $h =_{\mathsf{df}} 1+n$. Finally, a second value g_2 is selected as follows: w is sampled at random from $[\frac{n}{4}] =_{\mathsf{df}} \{0, \ldots, \lfloor \frac{n}{4} \rfloor\}$ and we set $g_2 \leftarrow g_1^w$. A random member $\mathcal{H}$ of a universal one-way hash function family UOWHF is selected [39]; the range of $\mathcal{H}$ is assumed to be $[0, 2^{\nu/2-2})$. The global parameters of the cryptosystem that will be shared by all recipients are equal to $\mathsf{param} = \langle n, g_1, g_2, \mathrm{desc}\mathcal{H}\rangle$, where $\mathrm{desc}\mathcal{H}$ is the description of $\mathcal{H}$.

Key-Generation. The key-generation algorithm $\mathcal{G}_{\mathsf{e}}$ receives the parameters $\langle n, g_1, g_2, \mathrm{desc}\mathcal{H}\rangle$, samples $x_1, x_2, y_1, y_2 \leftarrow_R [\frac{n^2}{4}]$ and sets $\mathsf{pk} = \langle c, d, y\rangle$ where $c = g_1^{x_1} g_2^{x_2}$, $d = g_1^{y_1} g_2^{y_2}$ and $y = g_1^z$; the secret-key is $\mathsf{sk} = \langle x_1, x_2, y_1, y_2, z\rangle$. Note that below we may include the string param as part of the pk and sk strings to avoid repeating it, nevertheless it should be recalled in all cases that $n, g_1, g_2, \mathrm{desc}\mathcal{H}$ are global parameters that are available to all parties.

Encryption. The encryption function $\mathcal{E}$ operates as follows: given the pk, a message w and a label L it samples $r \leftarrow_R [\frac{n}{4}]$ and outputs the triple $\langle u_1, u_2, e, v\rangle$ computed as follows: $u_1 \leftarrow g_1^r \bmod n^2$, $u_2 \leftarrow g_2^r \bmod n^2$, $e \leftarrow y^r (1+n)^w \bmod n^2$, $v \leftarrow ||c^r d^{r\mathcal{H}(u_1,u_2,e,L)} \bmod n^2||$ where $||\cdot|| : \mathbb{Z}^*_{n^2} \to \mathbb{Z}^*_{n^2}$ is defined as follows $||x|| = x$ if $x \leq n^2/2$ and $||x|| = -x$ if $x > n^2/2$. We note that the "absolute value" function $||\cdot||$ is used to disallow the malleability of a ciphertext with respect to multiplication with -1 (cf. the decryption test below). To summarize, encryption works as follows:

$$r \leftarrow_R \left[\frac{n}{4}\right] \quad : \quad u_1 \leftarrow g_1^r \quad u_2 \leftarrow g_2^r \quad e \leftarrow y^r h^m \quad v \leftarrow ||c^r d^{r\mathcal{H}(u_1,u_2,e,L)}||$$

Decryption. The decryption function $\mathcal{D}$ given a ciphertext (u_1, u_2, e, v) and a label L it performs the following checks:

$$v \stackrel{?}{=} ||v|| \qquad \wedge \qquad v^2 \stackrel{?}{=} (u_1^{x_1} u_2^{x_2})^2 (u_1^{y_1} u_2^{y_2})^{2\mathcal{H}(u_1,u_2,e,L)}$$

if all tests pass it computes $m' = e^2 {u_1}^{-2z} - 1 (\bmod n^2)$ and returns $(m' \cdot 2^{-1} \bmod n)/n$, otherwise it returns $\perp$.

This completes the description of the cryptosystem. Observe that the cryptosystem is correct, i.e., encryption inverts decryption: indeed, assuming that $\langle u_1, u_2, e, v\rangle \leftarrow \mathcal{E}(\mathsf{pk}, w, L)$, we have that $m' = e^2 u_1^{-2z} - 1 \equiv_{n^2} h^{2w} - 1$ and due to the fact that $h^x \equiv_{n^2} 1 + xn$ for all $x \in \mathbb{Z}_n$ we have that $w' \equiv_{n^2} (2m \bmod n) \cdot n$. It follows that $(w' \cdot 2^{-1} \bmod n)/n = w$.

We will next argue about the security of the cryptosystem. We note that the above cryptosystem has a "double trapdoor" property: for each public-key, c, d, y, based on parameters $n, g_1, g_2, \mathrm{desc}\mathcal{H}$, one trapdoor is the discrete-logarithm of y base g_1, whereas the the other trapdoor is the factorization of n. Indeed given the factorization of n, one can easily decrypt any ciphertext $\langle u_1, u_2, e, v\rangle$ by computing $e^{p'q'} \equiv_{n^2} h^{p'q'm}$. Subsequently m can be computed easily similarly to the regular decryption function. In GE the global trapdoor will not be used and the factorization of n will be assumed unknown by all parties. The intractability assumption that will be employed is the following:

Definition 5. *The* Decisional Composite Residuosity DCR *assumption [40]: It is computationally hard to distinguish between: (i) tuples of the form* $(n, u^n \bmod n^2)$ *where* n *is a composite RSA modulus and* $u \leftarrow_R \mathbb{Z}^*_{n^2}$*, and (ii) tuples of the form* (n, v) *where* $v \leftarrow_R \mathbb{Z}^*_{n^2}$.

Next, we prove $\mathsf{IND\text{-}CCA2}$ security under the DCR.

Theorem 2. *The cryptosystem* $\langle \mathcal{Z}_{\mathsf{e}}, \mathcal{G}_{\mathsf{e}}, \mathcal{E}, \mathcal{D}\rangle$ *defined above satisfies CCA2 security under the* DCR *assumption and the target collision resistance of the employed* UOWH *family.*

Interestingly, it is not clear whether the DCR can be used for proving the key-privacy of the cryptosystem. To see why this is the case consider the following: Consider the CPA version of the cryptosystem using only a single generator over $\mathcal{X}_{n^2}$: in the CPA case the cryptosystem is similar to ElGamal, with ciphertexts pairs of the form $\langle g^r \bmod n^2, y^r h^m \bmod n^2\rangle$. Note that $\mathsf{IND\text{-}CPA}$ security can be easily shown under the DCR assumption. On the other hand, to show CPA-key-privacy one has to (essentially) establish the indistinguishability of the distributions $\langle g, y_0, y_1, g^r, y_0^r h^m\rangle$ and $\langle g, y_0, y_1, g^r, y_1^r h^m\rangle$. It is not apparent how to apply DCR to prove this indistinguishability; ultimately this is because the message m is the same in both of these distributions and its randomization (easily provided by DCR) appears to be immaterial to the indistinguishability of the two distributions. It should be noted that since the adversary is not interested in the h^m portion of the ciphertext he can easily cancel it out by raising everything to n. For this reason the power of DCR seems of little use in this case, and a Diffie-Hellman-like assumption in $\mathcal{X}_{n^2}$ would seem more appropriate.

Based on the above we employ the Decisional Diffie Hellman assumption over the group $\mathcal{X}_{n^2}$, denoted as $\mathsf{DDH}_{\mathsf{SQNR}}$. Regarding the relationship between Diffie Hellman type of problems and the DCR we show the following theorem:

Theorem 3. $\mathsf{DCR} \Longrightarrow \mathsf{CDH}_{\mathsf{SQNR}}$

Based on the above we formulate our key-privacy theorem for the cryptosystem:

Theorem 4. *The cryptosystem* $\langle \mathcal{Z}, \mathcal{G}_{\mathsf{e}}, \mathcal{E}, \mathcal{D} \rangle$ *defined above satisfies CCA2-key-privacy under the* $\mathsf{DDH}_{\mathsf{SQNR}}$ *assumption and the target collision resistance of the employed* UOWH *family.*

Proof of Public-Key Validity. We will employ the public-key encryption scheme above to build the public-key $\mathsf{database}$ of the GE scheme. When a user joins the group he will be allowed to generate a public-key and he will be required to show that the public-key is valid. For our new cryptosystem the language of valid public-keys is $\mathcal{L}_{pk}^{\mathsf{param}} = \{\langle c, d, y \rangle \mid c, d, y \in \mathcal{X}_{n^2}\}$ where $\mathsf{param} = \langle n, g_1, g_2, \mathcal{H} \rangle$. It follows that joining will require three instances of a proof of language membership to the subgroup $\mathcal{X}_{n^2}$ of $\mathbb{Z}^*_{n^2}$. The validity of an element y can be performed by executing the following steps where $k_0, k_1 \in \mathbb{N}$ are parameters that affect the soundness and zero-knowledge properties of the proof of language membership below:

1. [User:] Select $t \xleftarrow{\mathrm{r}} \{0,1\}^{k_0}$ and transmit $a \leftarrow g^t \bmod n^2$.
2. [GM:] Select $c \xleftarrow{\mathrm{r}} \{0,1\}^{k_1}$ and transmit c.
3. [User:] Compute $s \leftarrow t - cz \in \mathbb{Z}$ and transmit s.
4. [GM:] Verify $a^2 \equiv_{n^2} (g_1^2)^s y^{2c}$.

It is easy to verify that given any prover that produces a value y and then executes the proof above, it must be the case that $y^2 \in \mathcal{X}_{n^2}$ with probability $1 - 2^{-k_1}$. Note that this still allows for a slight misbehavior on the part of the user as he can multiply y with an element of order 2 inside $\mathbb{Z}^*_{n^2}$; while it is easy to add an additional step in the above proof to avoid this slight misbehavior we will not do so as we will show the security properties of our GE scheme without such guarantee.

Construction of GE of Discrete-logarithms. We proceed to the description of the GE scheme $\mathtt{SETUP}, \mathtt{JOIN}, \langle \mathcal{G}_{\mathsf{dl}}, \mathcal{R}_{\mathsf{dl}}, \mathsf{sample}_{\mathsf{dl}} \rangle, \mathtt{ENC}, \mathtt{DEC}, \mathtt{OPEN}, \langle \mathcal{P}, \mathcal{V}, \mathsf{recon} \rangle$. First recall that from the discrete-logarithm relation, $\mathcal{G}_{\mathsf{dl}}$ given 1^ν samples a description of a cyclic group of ν-bits order and a generator γ of that group; $\mathcal{R}_{\mathsf{dl}}$ contains pairs of the form (x, w) where $x = \gamma^w$. Finally $\mathsf{sample}_{\mathsf{dl}}$ on input $\mathsf{pk}_{\mathcal{R}} = \langle \mathrm{desc}(G), \gamma \rangle$ selects a witness w and returns the pair $(x = \gamma^w, w)$.
Parameter Selection. The procedure $\mathtt{SETUP}$ selects the following parameters:

- ◦ Integer values k_0, k_1.
- ◦ A safe composite n of ℓ_n bits and generators $g, \breve{g}, g_1, g_2$ of the group $\mathcal{X}_{n^2}$.
- ◦ The description of a hash function $\mathcal{H}$ drawn at random from a UOWH family.
- ◦ A prime number Q of the form $\lambda \cdot n^2 + 1$ and F, H generators of the order n^2 subgroup of $\mathbb{Z}^*_Q$.
- ◦ A safe composite $\hat{n}$ of ℓ_N bits and two generators $\hat{g}, \hat{y}$ of the group $\mathcal{X}_{\hat{n}^2}$.
- ◦ A sequence of integers $G, Y_1, Y_2, Y_3 \in \mathbb{N}$ of length ℓ_N.

We stress that the above parameters are part of the trusted setup of the system (also referred to as the common reference string, and no participant of the system, including the GM, OA, or any user will know any private information about these values).

$\mathtt{SETUP}_{\mathsf{OA}}$. The procedure selects $\mathsf{x}_1, \mathsf{x}_2, \mathsf{y}_1, \mathsf{y}_2, \mathsf{z} \leftarrow_R [\frac{n^2}{4}]$ and set $\mathsf{pk}_{\mathsf{OA}} = \langle \breve{y}, \breve{c}, \breve{d} \rangle = \langle g^{\mathsf{z}}, g^{\mathsf{x}_1} \breve{g}^{\mathsf{x}_2}, g^{\mathsf{y}_1} \breve{g}^{\mathsf{y}_2} \rangle$.

$\mathtt{SETUP}_{\mathsf{GM}}$. The GM will employ a digital signature $\langle \mathcal{G}_{\mathsf{s}}, \mathcal{S}, \mathcal{V}_{\mathsf{s}} \rangle$ that must satisfy adaptive chosen message security and be suitable for engaging in proofs of knowledge of signed messages when the signature is committed. In our design will employ the block signature of Camenisch and Lysyanskaya [15] as the underlying digital signature scheme (hence referred to as CL-signature). The choice of the digital signature is not unique to our design and other signature schemes can be employed as well. The key-generation procedure $\mathcal{G}_{\mathsf{s}}$ (that will be used by GM in $\mathtt{SETUP}_{\mathsf{GM}}$) samples a pair $\langle \mathsf{sk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{GM}} \rangle$ where $\mathsf{pk}_{\mathsf{GM}} = \langle A_0, A_{1,\mathsf{c}}, A_{1,\mathsf{d}}, A_{1,\mathsf{y}}, A_2, N \rangle$ with N a safe composite of ℓ_N bits and $A_0, A_{1,\mathsf{c}}, A_{1,\mathsf{d}}, A_{1,\mathsf{y}}, A_2 \in \mathbb{Z}_N^*$ are random quadratic residues in $\mathcal{Q}_N$. The signing key $\mathsf{sk}_{\mathsf{GM}}$ is the factorization of N. In addition to ℓ_N we have the parameters ℓ_m where $[0, 2^{\ell_m}) \times [0, 2^{\ell_m}) \times [0, 2^{\ell_m})$ will be the message space for the signature such that $n^2 < 2^{\ell_m}$ (this is because we want to use the signature to sign public-keys of the encryption scheme).

$\mathtt{JOIN}$. The prospective group member submits c, d, y as generated by the encryption system $\langle \mathcal{G}_{\mathsf{e}}, \mathcal{E}, \mathcal{D} \rangle$ given in the beginning of the section. In particular, recall that $\langle c, d, y \rangle$ is defined as $c \leftarrow g_1^{x_1} g_2^{x_2} \bmod n^2, d \leftarrow g_1^{y_1} g_2^{y_2} \bmod n^2$, $y \leftarrow g_1^z$ and $x_1, x_2, y_1, y_2, z \leftarrow_R [\frac{n^2}{4}]$. The secret key of the user is set to the values x_1, x_2, y_1, y_2, z. The user engages with the GM in a proof of membership for the validity of c, d, y. Upon acceptance the GM will use the signing procedure $\mathcal{S}$ for CL-signatures that is as follows: given the message $M = \langle c, d, y \rangle$, the GM will sample $R \leftarrow [0, 2^{\ell_N + \ell_m + \ell})$ where ℓ is a security parameter and a random prime $E > 2^{\ell_m + 1}$ of length $\ell_m + 2$ bits; then it will compute $A = (A_0 A_{1,\mathsf{c}}^c A_{1,\mathsf{d}}^d A_{1,\mathsf{y}}^y A_2^R)^{1/E} (\mathrm{mod} N)$ (recall that the factorization of N is the signing key). Finally the signature to M is the triple $\langle A, E, R \rangle$.

Finally, the GM will enter $\langle c, d, y \rangle$ into the public $\mathsf{database}$ followed by the signature. Note that the GM should not allow a user to enter into $\mathsf{database}$ a key $\langle c, d, y \rangle$ such that there is some $\langle c_i, d_i, y_i \rangle$ in the database already for which it holds that $c^2 = c_i^2$, or $d^2 = d_i^2$ or $y^2 = y_i^2$. Recall that the verification algorithm $\mathcal{V}_{\mathsf{s}}$ given a message $M = \langle c, d, y \rangle$ and a signature $\langle A, E, R \rangle$ on it, checks whether it holds that $A^E = A_0 A_{1,\mathsf{c}}^c A_{1,\mathsf{d}}^d A_{1,\mathsf{y}}^y A_2^R \bmod N$ and verifies all the range constraints on c, d, y, E, R as stated above.

$\mathtt{ENC}$, $\mathtt{DEC}$ and recon. Following our modular design methodology of section 3 the GE encryption function consists of the encryption of the witness w under a recipient's public-key $\langle c, d, y \rangle$ and a sequence of commitments to the public-key used and commitments to the certificate of this public-key. More specifically when Alice wants to encrypt her witness w for her public-value $x = \gamma^w$ under label L she computes the following:

1. Commitment to Certificate of Public-key. The commitment to the certificate of the public-key of the recipient that Alice selected is formed as follows: for the certificate $\langle A, E, R\rangle$ the following values are computed $\tilde{B} = G^{2u} \bmod N$, $\tilde{A} = Y_1^{2u} A \bmod N$, $\tilde{E} = Y_2^{2u} G^{2E} \bmod N$, $\tilde{R} = Y_3^{2u} G^{2R} \bmod N$.

2. Bridge Commitments. The "bridge commitments" will assist in the efficient proof of ciphertext validity. In particular Alice includes the commitments $\hat{E} = \hat{g}^E (l_1)^{\hat{n}} \bmod \hat{n}^2$, $\hat{R} = \hat{g}^R (l_2)^{\hat{n}} \bmod \hat{n}^2$ and $l_j \stackrel{\mathtt{r}}{\leftarrow} \mathbb{Z}_n$ for $j = 1, 2$. Moreover she includes the commitments $\tilde{y} = H_{\mathsf{y}}^{u'} F^y \bmod Q, \tilde{c} = H_{\mathsf{c}}^{u'} F^c \bmod Q, \tilde{d} = H_{\mathsf{d}}^{u'} F^d \bmod Q$.

3. Encryption of the recipient's public-key. Encryption of the public-key that Alice selected is formed as three ciphertexts: $\langle f_{\mathsf{c}}, \breve{f}_{\mathsf{c}}, \dot{f}_{\mathsf{c}}, \ddot{f}_{\mathsf{c}}\rangle$, $\langle f_{\mathsf{d}}, \breve{f}_{\mathsf{d}}, \dot{f}_{\mathsf{d}}, \ddot{f}_{\mathsf{d}}\rangle$, $\langle f_{\mathsf{y}}, \breve{f}_{\mathsf{y}}, \dot{f}_{\mathsf{y}}, \ddot{f}_{\mathsf{y}}\rangle$, where each is selected as $\langle g^{u_{\mathsf{a}}}, \breve{g}^{u_{\mathsf{a}}}, \breve{y}^{u_{\mathsf{a}}} a, \breve{c}^{u_{\mathsf{a}}} \breve{d}^{u_{\mathsf{a}} \mathcal{H}(L'_{\mathsf{a}})}\rangle$ where $u_{\mathsf{a}} \stackrel{\mathtt{r}}{\leftarrow} [\frac{n}{4}]$, $\mathsf{a} \in \{\mathsf{y}, \mathsf{c}, \mathsf{d}\}$, $a \in \{y, c, d\}$ and $L'_{\mathsf{a}} = \langle f_{\mathsf{a}}, \breve{f}_{\mathsf{a}}, \dot{f}_{\mathsf{a}}, \ddot{f}_{\mathsf{a}}, L\rangle$.

4. Encryption of the witness. The encryption of witness w is as follows: $\langle u_1, u_2, e, v\rangle \leftarrow \langle g_1^r, g_2^r, y^r h^w, ||c^r d^{r\mathcal{H}(u_1, u_2, e, L'_{\mathsf{c}}, L'_{\mathsf{d}}, L'_{\mathsf{y}})}||\rangle$.

`DEC` is the decryption process as defined in the beginning of the section for the new encryption scheme. recon is simply the identity function.
`OPEN`. The opening procedure applies to the three ciphertext excluding the witness ciphertext (item 4, above). In particular, it returns $\langle c, d, y\rangle = \langle f_{\mathsf{c}} \dot{f}_{\mathsf{c}}^{-\mathsf{z}}, \dot{f}_{\mathsf{d}} f_{\mathsf{d}}^{-\mathsf{z}}, \dot{f}_{\mathsf{y}} f_{\mathsf{y}}^{-\mathsf{z}}\rangle$ or $\perp$ depending on the outcome of the tests $f_{\mathsf{a}}^{\mathsf{x}_1+\mathsf{y}_1} \breve{f}_{\mathsf{a}}^{(\mathsf{x}_2+\mathsf{y}_2)\mathcal{H}(L')} \stackrel{?}{=} \ddot{f}_{\mathsf{a}}$ for $\mathsf{a} \in \{\mathsf{y}, \mathsf{c}, \mathsf{d}\}$. The owner of the public-key is identified by comparing $\langle c^2, d^2, y^2\rangle$ to all entries $\langle c_i^2, d_i^2, y_i^2\rangle$ that are inside the database $\mathsf{database}$.

The proof of validity $\langle \mathcal{P}, \mathcal{V}\rangle$. This protocol will be constructed as an AND composition of four sub-protocols that due to lack of space presented in the full version [32]. These protocols belong to a class of efficient proofs for discrete log relations that are very common in the design of cryptographic primitives and their concrete and efficient instantiation has become quite standard in the literature. An exception perhaps is protocol # 2 which is a more complex protocol and is related to the "double-decker" proof of knowledge for discrete-logarithms [42, 20]. This protocol is the least efficient as it requires parallel repetition for decreasing the knowledge-error. Still, we stress that the overall communication is independent of the size of the group and well within practical limits.

Based on the above, the theorem below follows as a corollary of theorem 1:

Theorem 5. *The* GE *scheme for discrete-logarithms defined above satisfies (i)* Correctness*; (ii)* Anonymity *and (iii)* Security*, under the* $\mathsf{DDH}_{\mathsf{SQNR}}$*,* DDH *over* $\mathcal{Q}_N$*,* DCR *and the collision resistance of the* UOWH *family; (iv)* Soundness*, under the Strong-RSA and the DLOG assumptions.*

References

1. N. Asokan, V. Shoup, and M. Waidner. Optimistic fair exchange of digital signatures (extended abstract). In *EUROCRYPT*, pages 591–606, 1998.

2. G. Ateniese, J. Camenisch, S. Hohenberger, and B. de Medeiros. Practical group signatures without random oracles. Cryptology ePrint Archive, Report 2005/385, 2005. `http://eprint.iacr.org/`.
3. 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.
4. G. Ateniese and G. Tsudik. Some open issues and new directions in group signatures. In M. Franklin, editor, *Financial cryptography: Third International Conference, FC '99, Anguilla, British West Indies, February 22–25, 1999: proceedings*, volume 1648 of *Lecture Notes in Computer Science*, pages 196–211. Springer-Verlag, 1999.
5. M. Bellare, A. Boldyreva, A. Desai, and D. Pointcheval. Key-privacy in public-key encryption. In C. Boyd, editor, *ASIACRYPT*, volume 2248 of *Lecture Notes in Computer Science*, pages 566–582. Springer, 2001.
6. M. Bellare, D. Micciancio, and B. Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In E. Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, Warsaw, Poland, 2003. Springer.
7. M. Bellare, H. Shi, and C. Zhang. Foundations of group signatures: The case of dynamic groups. In A. Menezes, editor, *CT-RSA*, volume 3376 of *Lecture Notes in Computer Science*, pages 136–153. Springer, 2005.
8. D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In M. Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 41–55. Springer, 2004.
9. X. Boyen and B. Waters. Compact group signatures without random oracles. In S. Vaudenay, editor, *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 427–444. Springer, 2006.
10. E. Bresson, D. Catalano, and D. Pointcheval. A simple public-key cryptosystem with a double trapdoor decryption mechanism and its applications. In C. S. Laih, editor, *Proc. of Asiacrypt '03*, volume 2894 of *LNCS*, pages 37–54, Taipei, TW, November-December 2003. IACR, Springer-Verlag.
11. 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.
12. J. Camenisch and I. Damgård. Verifiable encryption, group encryption, and their applications to separable group signatures and signature sharing schemes. In T. Okamoto, editor, *ASIACRYPT*, volume 1976 of *Lecture Notes in Computer Science*, pages 331–345. Springer, 2000.
13. J. Camenisch and A. Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In B. Pfitzmann, editor, *EUROCRYPT*, volume 2045 of *Lecture Notes in Computer Science*, pages 93–118. Springer, 2001.
14. 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.
15. J. Camenisch and A. Lysyanskaya. A signature scheme with efficient protocols. In *International Conference on Security in Communication Networks – SCN*, volume 2576 of *Lecture Notes in Computer Science*, pages 268–289. Springer Verlag, 2002.

16. J. Camenisch and A. Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In M. K. Franklin, editor, *CRYPTO*, volume 3152 of *Lecture Notes in Computer Science*, pages 56–72. Springer, 2004.
17. 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.
18. 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.
19. J. Camenisch and V. Shoup. Practical verifiable encryption and decryption of discrete logarithms. In *CRYPTO 2003*. Springer-Verlag, 2003.
20. J. Camenisch and M. Stadler. Efficient group signature schemes for large groups. In B. S. K. Jr., editor, *Advances in Cryptology – CRYPTO ' 1997*, volume 1294 of *Lecture Notes in Computer Science*, pages 410–424. International Association for Cryptologic Research, Springer, 1997.
21. D. Chaum. Private communication, 2006.
22. D. Chaum and E. van Heyst. Group signatures. In D. W. Davies, editor, *Advances in Cryptology, EUROCRYPT 1991 (Lecture Notes in Computer Science 547)*, pages 257–265. Springer-Verlag, April 1991. Brighton, U.K.
23. 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.
24. R. Cramer and V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In H. Krawczyk, editor, *CRYPTO 1998*, pages 13–25. Springer-Verlag, 1998. Lecture Notes in Computer Science No. 1462.
25. I. Damgård. Efficient concurrent zero-knowledge in the auxiliary string model. In *EUROCRYPT*, pages 418–430, 2000.
26. Y. Desmedt. Society and group oriented cryptography: A new concept. In C. Pomerance, editor, *CRYPTO*, volume 293 of *Lecture Notes in Computer Science*, pages 120–127. Springer, 1987.
27. 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, 6–8May 1991.
28. R. Gennaro and Y. Lindell. A framework for password-based authenticated key exchange. In E. Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, Warsaw, Poland, 2003. Springer.
29. O. Goldreich. *The Foundations of Cryptography - Volume 1*. Cambridge University Press, 2004.
30. J. Groth. Simulation-sound nizk proofs for a practical language and constant size group signatures. In X. Lai and K. Chen, editors, *ASIACRYPT*, volume 4284 of *Lecture Notes in Computer Science*, pages 444–459. Springer, 2006.
31. S. Halevi. Sufficient condition for key privacy. Cryptology ePrint Archive, Report 2005/005, 2005. `http://eprint.iacr.org/`.
32. A. Kiayias, Y. Tsiounis, and M. Yung. Group encryption. Cryptology ePrint Archive, Report 2007/015, 2007. `http://eprint.iacr.org/`.

33. 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.
34. A. Kiayias and M. Yung. Group signatures with efficient concurrent join. In R. Cramer, editor, *EUROCRYPT*, volume 3494 of *Lecture Notes in Computer Science*, pages 198–214. Springer, 2005.
35. A. Kiayias and M. Yung. Secure scalable group signature with dynamic joins and separable authorities. *Int. J. Security and Networks*, 1(1/2):24–45, 2006.
36. 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.
37. J. K. Liu, P. P. Tsang, D. S. Wong, and R. W. Zhu. Universal custodian-hiding verifiable encryption for discrete logarithms. In D. Won and S. Kim, editors, *ICISC*, volume 3935 of *Lecture Notes in Computer Science*, pages 389–409. Springer, 2005.
38. J. K. Liu, V. K. Wei, and D. S. Wong. Custodian-hiding verifiable encryption. In C. H. Lim and M. Yung, editors, *WISA*, volume 3325 of *Lecture Notes in Computer Science*, pages 51–64. Springer, 2004.
39. M. Naor and M. Yung. Universal one-way hash functions and their cryptographic applications. In *STOC*, pages 33–43. ACM, 1989.
40. P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in cryptology—EUROCRYPT 1999*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–238, 1999.
41. 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.
42. M. Stadler. Publicly verifiable secret sharing. In U. Maurer, editor, *Advances in Cryptology – EUROCRYPT ' 96*, volume 1070 of *Lecture Notes in Computer Science*, pages 190–199. International Association for Cryptologic Research, Springer, 1996.
43. M. Trolin and D. Wikström. Hierarchical group signatures. In L. Caires, G. F. Italiano, L. Monteiro, C. Palamidessi, and M. Yung, editors, *ICALP*, volume 3580 of *Lecture Notes in Computer Science*, pages 446–458. Springer, 2005.
44. A. Young and M. Yung. A pvss as hard as discrete log and shareholder separability. In K. Kim, editor, *Public Key Cryptography*, volume 1992 of *Lecture Notes in Computer Science*, pages 287–299. Springer, 2001.