New Constructions and Applications of Trapdoor DDH Groups*

Yannick Seurin

ANSSI, Paris, France
yannick.seurin@m4x.org

Abstract. Trapdoor Decisional Diffie-Hellman (TDDH) groups, introduced by Dent and Galbraith (ANTS 2006), are groups where the DDH problem is hard, unless one is in possession of a secret trapdoor which enables solving it efficiently. Despite their intuitively appealing properties, they have found up to now very few cryptographic applications. Moreover, among the two constructions of such groups proposed by Dent and Galbraith, only a single one based on hidden pairings remains unbroken. In this paper, we extend the set of trapdoor DDH groups by giving a construction based on composite residuosity. We also introduce a more restrictive variant of these groups that we name *static* trapdoor DDH groups, where the trapdoor only enables to solve the DDH problem with respect to a fixed pair (G, G^x) of group elements. We give two constructions for such groups whose security relies respectively on the RSA and the factoring assumptions. Then, we show that static trapdoor DDH groups yield elementary constructions of convertible undeniable signature schemes allowing delegatable verification. Using our constructions of static trapdoor DDH groups from the RSA or the factoring assumption, we obtain slightly simpler variants of the undeniable signature schemes of respectively Gennaro, Rabin, and Krawczyk (J. Cryptology, 2000) and Galbraith and Mao (CT-RSA 2003). These new schemes are conceptually more satisfying since they can strictly be viewed as instantiations, in an adequate group, of the original undeniable signature scheme of Chaum and van Antwerpen (CRYPTO '89).

Keywords: trapdoor DDH group, hidden pairing, signed quadratic residues, convertible undeniable signature scheme

1 Introduction

1.1 The CDH and DDH Problems

Given a group $\mathbb{G}$ and an element $G \in \mathbb{G}$ of large order, the Computational Diffie-Hellman (CDH) problem is to compute G^{xy}, given G^x and G^y for random integers x, y. The Decisional Diffie-Hellman (DDH) problem is to distinguish the two distributions (G^x, G^y, G^{xy}) and (G^x, G^y, G^z) for random and independent integers x, y, z. Usually, when considering the status of various groups with respect to the CDH and DDH problems, one of the following two cases arises: either the CDH and DDH problems are both presumably hard (this is the case for example for subgroups of large prime order of $\mathbb{Z}_p^*$, p prime), or the group is a so-called *gap group*: the CDH problem is (presumably) hard while the DDH problem is universally easy (*i.e.* easy given only the description of the group law, which seems to be the minimal publicly available information to obtain useful applications). The latter case typically arises in certain elliptic curve groups equipped with bilinear pairings [MOV93, FMR99], and has given rise to many important applications in cryptography [Jou00, BF01, BLS04].

1.2 Trapdoor DDH Groups

Trapdoor DDH groups (*TDDH groups* for short), introduced by Dent and Galbraith [DG06], lie somewhere between the above two cases. These are groups where the DDH problem is hard, except if one possesses a trapdoor for solving it efficiently. Dent and Galbraith gave two candidates for such groups based on the concept of *hidden pairings*, one in elliptic curves over the ring $\mathbb{Z}_N$, where N is hard to factor, and the other one based on Frey's idea of disguising an elliptic curve [Fre98]. Subsequently, the

* An abridged version appears at PKC 2013. This is the full version.

second proposal was broken by Morales [Mor08]. Since the DDH problem is the basis of so many cryptosystems [Bon98], the concept of trapdoor DDH groups is very attractive. Indeed, it should enable to control more precisely who is able to solve the DDH problem in a system. This may help in situations where there is a conflict between security, which requires a group where the DDH problem is hard, and some interesting additional functionalities that could be achieved thanks to an algorithm for solving the DDH problem. One example that comes to mind is threshold ElGamal encryption. In threshold ElGamal encryption [DF89], given a secret/public key pair $(x, X = G^x)$, each decryption server is given a share x_i of the secret key, to which is associated a "partial" public key G^{x_i}. In order to decrypt a ciphertext $(R, Y) = (G^r, MX^r)$, each server participating to decryption must compute a decryption share $S_i = R^{x_i}$. Hence, checking whether a decryption share from a server is correct or not amounts to deciding whether (X_i, R, S_i) is a DDH tuple or not. Yet IND-CPA-security of ElGamal encryption is equivalent to the hardness of the DDH problem in the underlying group $\mathbb{G}$ [TY98]. Hence, there seems to be no other choice than using a group where the DDH problem is hard, thereby condemning other participants to be unable to distinguish correct decryption shares from incorrect ones. We do not claim that TDDH groups are the best way to solve this problem (this can be more easily achieved by having each server provide a non-interactive zero-knowledge proof that his decryption share is correctly computed), and this example only serves to argue that sometimes, one may want that only some authorized party be able to solve the DDH problem. Despite these considerations, TDDH groups have found up to now very few cryptographic applications. In their original paper, Dent and Galbraith gave only one example, namely an identification scheme. To the best of our knowledge, the only previous paper proposing a non-trivial application of TDDH groups (namely the construction of statistically hiding sets, a variant of zero-knowledge sets) is due to Prabhakaran and Xue [PX09].

1.3 Contributions of this Work

The contributions of this paper can be summarized as follows. First, at a conceptual level, we refine the definition of TDDH groups of Dent and Galbraith by requiring that the CDH problem remain hard even given the trapdoor for solving the DDH problem. This was not made explicit in the formalization by Dent and Galbraith, yet we think that this is probably a key feature for many interesting applications, such as undeniable signatures for example. We also broaden the set of constructions of trapdoor DDH groups. We propose a new construction based on composite residuosity in $\mathbb{Z}^*_{N^2}$ (similar considerations have been made by [BCP03], albeit not in the formalism of TDDH groups), and identify under which hardness assumptions this group satisfies our definition. A drawback of this construction is that it lacks what we call *perfect soundness*, meaning that the algorithm solving the DDH problem with the trapdoor can sometimes err and declare valid a non-DH tuple.

Then, we introduce a variant of trapdoor DDH groups that we name *static* trapdoor DDH groups. Their definition is very similar to the one of trapdoor DDH groups, except that the trapdoor for solving the DDH problem is now dedicated to a specific pair of group elements (G, G^x), hence the name *static*. We then show that such groups can be easily constructed from the RSA and the factoring problems. This concept abstracts some of the ideas underlying the work of Hofheinz and Kiltz [HK09], who showed that the Strong Diffie-Hellman (SDH) problem (*i.e.* solving the CDH problem given access to a static DDH oracle) is hard in the so-called group of signed quadratic residues under the factoring assumption.

Finally, we describe a very natural application of (static or not) TDDH groups to convertible undeniable signature schemes. Namely, the construction we propose is exactly the original undeniable signature scheme proposed by Chaum and van Antwerpen [CvA89] (for which deciding the validity of a signature is equivalent to solving the DDH problem), but in a TDDH group rather than simply a group where the DDH problem is hard. The trapdoor for solving the DDH problem can then be used to universally convert or delegate verification of signatures. Once instantiated with our proposals of static TDDH groups based on the RSA or the factoring problems, we obtain schemes similar to previous RSA-based undeniable signature schemes due to Gennaro, Rabin, and Krawczyk [GRK00] and Galbraith and

Mao [GM03]. However, these new schemes are conceptually simpler and easier to analyze. Moreover, since they are strict instantiations of the Chaum and van Antwerpen scheme, their confirmation and disavowal protocols can use classical proofs of equality or inequality of discrete logarithms, which are simpler and more efficient than what was proposed previously for the schemes of [GRK00, GM03].

1.4 Open Problems

Two key features of TDDH groups are perfect soundness (the property that the algorithm for solving the DDH problem with the trapdoor perfectly distinguishes DH tuples from non-DH tuples), and the possibility to securely hash into the group (see discussion in Section 2.3). However, none of the two candidates for TDDH groups (the hidden pairing based proposal of [DG06], and our proposal in Section 3.2) fulfills both requirements. We think that providing a plausible candidate possessing both properties is the key to enable powerful applications of TDDH groups.[1] A related open problem is whether there exists a (plausible construction of a) TDDH group with publicly known (ideally prime) order, since they are usually simpler to use in cryptography.

1.5 Organization

In Section 2 we give some basic definitions and introduce some of the tools we will need in the remainder of the paper. In Section 3, we define trapdoor DDH groups, and give a construction based on composite residuosity. In Section 4, we introduce static trapdoor DDH groups, and give two constructions based on respectively the RSA and the factoring assumptions. Finally, in Section 5, we show how to obtain convertible undeniable signature schemes from static TDDH groups, and discuss their instantiation with the constructions described previously.

2 Preliminaries

2.1 Notation and Definitions

The set of integers i such that $a \leq i \leq b$ will be denoted $[a; b]$. The security parameter will be denoted k. A function f of the security parameter is said *negligible* if for any $c > 0$, $f(k) \leq 1/k^c$ for sufficiently large k. When S is a non-empty finite set, we write $s \leftarrow_{\$} S$ to mean that a value is sampled uniformly at random from S and assigned to s. By $z \leftarrow \mathcal{A}^{\mathcal{O}_1, \mathcal{O}_2, \cdots}(x, y, \ldots)$ we denote the operation of running the (possibly probabilistic) algorithm $\mathcal{A}$ on inputs $x, y, \ldots$ with access to oracles $\mathcal{O}_1, \mathcal{O}_2, \ldots$ (possibly none), and letting z be the output. PPT will stand for probabilistic polynomial-time. Given two Interactive Turing Machines $\mathcal{P}$ and $\mathcal{V}$, we denote $w \leftarrow \langle \mathcal{P}(x), \mathcal{V}(y) \rangle(z)$ to mean that the output of the interaction of $\mathcal{P}$ with private input x and $\mathcal{V}$ with private input y on common input z is w.

Given an integer N, the multiplicative group of integers modulo N is denoted $\mathbb{Z}_N^*$. This group has order $\phi(N)$ where $\phi(\cdot)$ is the Euler function and exponent $\lambda(N)$ where $\lambda(\cdot)$ is the Carmichael function. We denote $\mathbb{J}_N$ the subgroup of $\mathbb{Z}_N^*$ of all elements $x \in \mathbb{Z}_N^*$ with Jacobi symbol $\left(\frac{x}{N}\right) = 1$. This subgroup has index 2 and order $\phi(N)/2$ in $\mathbb{Z}_N^*$. Moreover it is efficiently recognizable even without the factorization of N since the Jacobi symbol is efficiently computable given only N. We also denote $\mathbb{QR}_N$ the subgroup of quadratic residues of $\mathbb{Z}_N^*$. This subgroup is widely believed not to be efficiently recognizable when N is composite and its factorization is unknown (this is the Quadratic Residuosity assumption). We call a prime number p such that $(p-1)/2$ is prime a *safe prime*.

In all the following, given a group $\mathbb{G}$, we use the notation $[\mathbb{G}]$ to denote a description of the group, *i.e.* an efficient algorithm for computing the group operation. This notation always implies that $\mathbb{G}$ is efficiently recognizable. We assume that it is always possible to derive from the description of the group

[1] Our examples of *static* TDDH groups do fulfill both requirement, however non-static TDDH groups would allow more flexibility in cryptographic applications.

a negligibly close upper bound on the order $|\mathbb{G}|$ of the group (in some cases the exact order may be efficiently computable), and we use the notation $|\mathbb{G}|^+$ to denote this upper bound.[2] Given an element $G \in \mathbb{G}$, we denote $\mathtt{ord}(G)$ its order, $\langle G \rangle$ the group generated by G, $\mathtt{Dlog}_G(X)$ the discrete logarithm in base G of an element $X \in \langle G \rangle$, and $\mathtt{CDH}_G(X,Y) = G^{\mathtt{Dlog}_G(X)\mathtt{Dlog}_G(Y)}$. We also denote $\mathcal{DH}_G \subset \langle G \rangle^3$ the set of Diffie-Hellman (DH) tuples with respect to G:

$$\mathcal{DH}_G = \{(G^x, G^y, G^{xy}), x, y \in [0; \mathtt{ord}(G) - 1]\} \quad .$$

A group generator $\mathtt{Gen}$ is a PPT algorithm which on input a security parameter 1^k, outputs a tuple $([\mathbb{G}], G, \gamma)$ where $[\mathbb{G}]$ is the description of a group $\mathbb{G}$, $G \in \mathbb{G}$ is an element of order $2^{\Theta(k)}$, and γ is some arbitrary side information. We say that the CDH problem is hard for $\mathtt{Gen}$ if for any PPT adversary $\mathcal{A}$, the following probability is negligible:

$$\Pr\left[([\mathbb{G}], G, \gamma) \leftarrow \mathtt{Gen}(1^k), (X,Y) \leftarrow_\$ \langle G \rangle^2, Z \leftarrow \mathcal{A}([\mathbb{G}], G, \gamma; X, Y) : Z = \mathtt{CDH}_G(X,Y)\right] \quad .$$

We say that the DDH problem is hard for $\mathtt{Gen}$ if for any PPT adversary $\mathcal{A}$, the following advantage is negligible:

$$\Big|\Pr\left[([\mathbb{G}], G, \gamma) \leftarrow \mathtt{Gen}(1^k), (X,Y) \leftarrow_\$ \langle G \rangle^2, Z \leftarrow \mathtt{CDH}_G(X,Y) : 1 \leftarrow \mathcal{A}([\mathbb{G}], G, \gamma; X, Y, Z)\right] -$$
$$\Pr\left[([\mathbb{G}], G, \gamma) \leftarrow \mathtt{Gen}(1^k), (X,Y,Z) \leftarrow_\$ \langle G \rangle^3 : 1 \leftarrow \mathcal{A}([\mathbb{G}], G, \gamma; X, Y, Z)\right]\Big| \quad .$$

2.2 Proofs of Equality and Inequality of Discrete Logarithms

Protocols for proving, given $(G, X, Y, Z) \in \mathbb{G}$, the equality of discrete logarithms (EDL) $\mathtt{Dlog}_G(X) = \mathtt{Dlog}_Y(Z)$ or the inequality of discrete logarithms (IDL) constitute (among many other applications) the heart of respectively the confirmation and disavowal protocols for many undeniable signature schemes, and have therefore been the subject of many works. They vary depending on the exact kind of zero-knowledge property one wants to achieve. The basic honest-verifier zero-knowledge (HVZK) proof of EDL is due to Chaum and Pedersen [CP92], while the simplest HVZK proof of IDL is due to Camenish and Shoup [CS03a]. These protocols are usually described for ambient groups $\mathbb{G}$ with publicly known prime order, in which case recognizing $\langle G \rangle$ is trivial, so that these protocols are actually proofs that a tuple $(X, Y, Z) \in \mathbb{G}^3$ is in $\mathcal{DH}_G$ or not. They can be adapted to the case where the order of the ambient group is composite and secret using well-known techniques [Gir90, Gir91], with the caveat that if $\langle G \rangle$ is not efficiently recognizable, the verifier must be promised that $X, Y, Z \in \langle G \rangle$ since these proofs do not in general ensure membership of X, Y, Z in $\langle G \rangle$ with negligible soundness.[3] Stated differently, if $\mathbb{G}'$ is a cyclic and efficiently recognizable subgroup of $\mathbb{G}$ (*e.g.* $\mathbb{G} = \mathbb{Z}_N^*$ and $\mathbb{G}' = \mathbb{J}_N$ when $\mathbb{J}_N$ is cyclic), these protocols are actually proofs that a tuple $(X, Y, Z) \in \mathbb{G}'$ is a DH tuple with respect to G or not, assuming that the verifier is guaranteed that G is indeed a generator of $\mathbb{G}'$ (which may not be efficiently checkable). The HVZK protocols for EDL and IDL are described in Appendix A. They can be strengthen to achieve various notions of zero-knowledge (against cheating verifiers) using known techniques [GSV98, CDM00, Dam00, Gen04] that we do not discuss in this paper.

The HVZK proofs of EDL and IDL can be made non-interactive in the Random Oracle Model using the Fiat-Shamir transformation [FS86], *i.e.* by having the prover compute the challenge (first message from the verifier) by itself by applying a hash function to the commitment (first message from the prover). Note that these proofs then become universally convincing.

[2] *E.g.* when $\mathbb{G} = \mathbb{Z}_p^*$ for some prime number p, $|\mathbb{G}|^+ = p - 1$, while when $\mathbb{G} = \mathbb{Z}_N^*$, where the factorization of N is secret, $|\mathbb{G}|^+ = N$.

[3] The soundness of the Schnorr protocol [Sch91], seen as a proof of membership in $\langle G \rangle$, is $1/\ell$, where ℓ is the smallest prime factor of the order of the ambient group $\mathbb{G}$.

2.3 Hashing into Groups

For many applications (and in particular for undeniable signatures based on the Chaum and van Antwerpen scheme [CvA89]), it is required to securely hash into the subgroup $\langle G \rangle$ specified by the group generator $\mathtt{Gen}$. Assuming the existence of good hash functions $\mathcal{H}$ from $\{0,1\}^*$ into the ambient group $\mathbb{G}$ (which in turn can quite often be securely constructed from hash functions $\mathcal{H}_k : \{0,1\}^* \to \{0,1\}^k$), there might or might not exist good constructions based on $\mathcal{H}$ for this. Whether a construction is good or not can be analyzed in the indifferentiability framework of Maurer *et al.* [MRH04, BCI+10], modeling $\mathcal{H}_k$ as a random oracle. In general, when $\langle G \rangle$ is an efficiently recognizable and sufficiently dense subset of $\mathbb{G}$, defining $\mathcal{H}'(x) = \mathcal{H}(x\|i)$ for the smallest $i \geq 0$ (encoded on sufficiently many bits) such that $\mathcal{H}(x\|i) \in \langle G \rangle$ can be shown to be indifferentiable from a random oracle from $\{0,1\}^*$ into $\langle G \rangle$. More efficient constructions may exist depending on the specific case (see Sections 4.2 and 4.3). However, when the subgroup $\langle G \rangle$ is not (known to be) efficiently recognizable, there is in general no secure way to hash into it. In particular, the construction $\mathcal{H}'(x) = G^{\mathcal{H}_k(x)}$ will almost surely ruin the security of any scheme based on the discrete logarithm and related problems since this construction reveals the discrete logarithm in base G of its outputs.[4] In the sequel, we discuss, when they exist, good constructions of hash functions into each TDDH group we consider.

3 Trapdoor DDH Groups

We start by defining trapdoor DDH groups. Our definition is a refinement of the one of Dent and Galbraith [DG06] in that we explicitly require that the CDH problem remain hard even given the trapdoor τ enabling to solve the DDH problem.

3.1 Definition

Definition 1. *A* trapdoor DDH group $\mathcal{TDDH}$ *is a pair of algorithms* $(\mathtt{Gen}, \mathtt{Solve})$ *with the following properties. The trapdoor DDH group generator algorithm* $\mathtt{Gen}$ *is a PPT algorithm which takes as input a security parameter* 1^k *and outputs a tuple* $([\mathbb{G}], G, \tau)$ *where* $[\mathbb{G}]$ *is the description of a group* $\mathbb{G}$*,* $G \in \mathbb{G}$ *is a group element of order* $2^{\Theta(k)}$*, and* τ *is a trapdoor information, such that:*

i) hardness of DDH without the trapdoor*: the DDH problem is hard for the group generator* $\mathtt{Gen}'$ *which outputs only* $([\mathbb{G}], G)$*;*
ii) hardness of CDH with the trapdoor*: the CDH problem is hard for* $\mathtt{Gen}$*.*

$\mathtt{Solve}$ *is a deterministic polynomial-time algorithm which takes as input* $([\mathbb{G}], G, \tau)$ *and a tuple* $(X, Y, Z) \in \mathbb{G}^3$*, either accepts (outputs 1) or rejects (outputs 0), and satisfies the following:*

iii) completeness*: for all* $([\mathbb{G}], G, \tau)$ *possibly output by* $\mathtt{Gen}$*,* $\mathtt{Solve}$ *always accepts on input a DH tuple* $(X, Y, Z) \in \mathcal{DH}_G$*;*
iv) soundness*: for any PPT adversary* $\mathcal{A}$*, the following probability is negligible:*

$$\Pr\left[([\mathbb{G}], G, \tau) \leftarrow \mathtt{Gen}(1^k), (X, Y) \leftarrow_{\$} \langle G \rangle^2, Z \leftarrow \mathcal{A}([\mathbb{G}], G; X, Y) : 1 \leftarrow \mathtt{Solve}([\mathbb{G}], G, \tau; X, Y, Z) \wedge (X, Y, Z) \notin \mathcal{DH}_G\right] .$$

We say that $\mathcal{TDDH}$ *has perfect soundness when* $\mathtt{Solve}$ *always rejects on input a non-DH tuple* (X, Y, Z)*, so that the above probability is zero.*

[4] When the discrete logarithm is hard in $\langle G \rangle$, it can be shown that this construction is not indifferentiable from a random oracle from $\{0,1\}^*$ to $\langle G \rangle$.

Note that the soundness condition implies in particular that $\texttt{Solve}$, on input a *uniformly random* tuple $(X, Y, Z) \in \mathbb{G}^3$, accepts only with negligible probability. We silently assumed in the above definition that $\texttt{Solve}$ is always run with a correctly generated trapdoor. This is safe for all examples presented below since there is an efficient way, given $([\mathbb{G}], G, \tau)$, to check whether the trapdoor is correct. We assume that $\texttt{Solve}$ outputs a special symbol $\perp$ when this is not the case. We recall the original proposal of a TDDH group based on hidden pairings by Dent and Galbraith [DG06] in Appendix B.

3.2 A TDDH Group Based on Composite Residuosity

In this section, we describe a TDDH group $\mathcal{TDDH}_{\texttt{BCP}}$ based on the group of quadratic residues modulo N^2, where N is an RSA modulus. This group was first considered by Bresson, Catalano, and Pointcheval [BCP03], who noticed that when the factorization of N is publicly available, this constitutes a *gap group*, *i.e.* a group where the CDH problem is hard and the DDH problem is easy. Here, we show that it constitutes in fact a TDDH group when the factorization of N is kept secret and used as the trapdoor.

We first recall some basic facts about the group of quadratic residues modulo N^2, where N is an RSA modulus. Let p, q be two safe primes where $p = 2p' + 1$ and $q = 2q' + 1$ (p' and q' primes), and $N = pq$. The group $\mathbb{QR}_{N^2}$ of quadratic residues modulo N^2 is a cyclic group of order $m = Np'q'$. We define the notion of *partial discrete logarithm*.

Definition 2 (Partial Discrete Logarithm). *Given a generator G of $\mathbb{QR}_{N^2}$, the partial discrete logarithm of a group element $X \in \mathbb{QR}_{N^2}$ is defined as* $\texttt{PDlog}_G(X) = \texttt{Dlog}_G(X) \bmod N$.

Computing the partial discrete logarithm is believed to be hard without the factorization of N.[5] However, it can be efficiently computed given the prime factors of N (or simply $\lambda(N)$) as follows [Pai99]:

1. input: N, $\lambda(N)$, generator G of $\mathbb{QR}_{N^2}$ and $X \in \mathbb{QR}_{N^2}$; output: $\texttt{PDlog}_G(X)$
2. for integers $u \in [0; N^2 - 1]$ such that $u = 1 \bmod N$, define the function (having integer values) $\mathcal{L}(u) = (u - 1)/N$
3. return
$$\frac{\mathcal{L}(X^{\lambda(N)} \bmod N^2)}{\mathcal{L}(G^{\lambda(N)} \bmod N^2)} \bmod N \ .$$

We now formally describe the TDDH group $\mathcal{TDDH}_{\texttt{BCP}}$. On input the security parameter 1^k, $\texttt{Gen}_{\texttt{BCP}}$ selects two k-bit safe primes $p = 2p' + 1$ and $q = 2q' + 1$, sets $N = pq$, selects a random generator G of $\mathbb{QR}_{N^2}$, and outputs $([\mathbb{Z}^*_{N^2}], G, \tau = (p, q))$. The $\texttt{Solve}_{\texttt{BCP}}$ algorithm works as follows: on input a tuple $(X, Y, Z) \in (\mathbb{Z}^*_{N^2})^3$ (as well as the trapdoor $\tau = (p, q)$), it checks whether $X, Y, Z \in \mathbb{QR}_{N^2}$, computes $x' = \texttt{PDlog}_G(X)$, $y' = \texttt{PDlog}_G(Y)$, and $z' = \texttt{PDlog}_G(Z)$ as described above, and checks whether $z' = x'y' \bmod N$. It accepts if this holds and rejects otherwise. The security of this TDDH group relies on a "partial" version of the CDH problem, defined as follows.

Definition 3 (Partial CDH Problem). *We say that the* Partial CDH problem *is hard if for any PPT algorithm $\mathcal{A}$, the following probability is negligible:*

$$\Pr[([\mathbb{Z}^*_{N^2}], G, \tau) \leftarrow \texttt{Gen}_{\texttt{BCP}}(1^k), (X, Y) \leftarrow_{\$} \langle G \rangle^2, Z \leftarrow \mathcal{A}([\mathbb{Z}^*_{N^2}], G; X, Y) : \texttt{Dlog}_G(Z) \equiv \texttt{Dlog}_G(X)\texttt{Dlog}_G(Y) \bmod N] \ .$$

Theorem 1. *Assuming that the DDH problem (without the factorization of N), the CDH problem (with the factorization of N), and the Partial CDH problem (without the factorization of N) are hard for $\mathbb{QR}_{N^2}$, $\mathcal{TDDH}_{\texttt{BCP}}$ is a trapdoor DDH group.*

[5] As noted by Paillier [Pai99] and in [BCP03], the Partial Discrete Logarithm problem can be shown equivalent to the Composite Residuosity Class problem in the particular case considered here.

Proof. We prove that properties *i*) to *iv*) of Definition 1 are satisfied. Properties *i*) and *ii*) follow directly from the assumptions that respectively the DDH (without the factorization of N) and the CDH (with the factorization of N) problems are hard in $\mathbb{QR}_{N^2}$. Property *iii*) is straightforward to verify by definition of $\texttt{Solve}_{\texttt{BCP}}$. Finally, property *iv*) follows from the hardness of the Partial CDH problem. □

Note that this TDDH group does not have perfect soundness. In particular, on input a random tuple $(X, Y, Z) \in (\mathbb{QR}_{N^2})^3$, there is a negligible probability that $\texttt{Solve}_{\texttt{BCP}}$ accepts and yet $(X, Y, Z) \notin \mathcal{DH}_G$ (this probability can easily be seen to be $\mathcal{O}(1/N)$ [BCP03]). Moreover, given the trapdoor $\tau = (p, q)$, and two random elements $(X, Y) \in (\mathbb{QR}_{N^2})^2$, it is easy to generate Z such that $(X, Y, Z) \notin \mathcal{DH}_G$ and yet $\texttt{Solve}_{\texttt{BCP}}$ accepts on input (X, Y, Z): simply compute $x' = \texttt{PDlog}_G(X)$ and $y' = \texttt{PDlog}_G(Y)$ and output $G^{x'y' \bmod N}$. Alternatively, given two random elements $(X, Y) \in (\mathbb{QR}_{N^2})^2$ and $Z = \texttt{CDH}_G(X, Y)$, it is easy to compute $Z' \neq Z$ such that $\texttt{Solve}_{\texttt{BCP}}$ accepts on input (X, Y, Z'): simply compute $Z' = ZU^N$ for some random $U \in \mathbb{QR}_{N^2}$. This may be of concern in some applications, especially for undeniable signature schemes where $\texttt{Solve}$ is typically used to check the validity of signatures (see Section 5).[6]

Hashing into $\mathbb{QR}_{N^2}$. The Quadratic Residuosity assumption states that no efficient algorithm can recognize elements of $\mathbb{QR}_{N^2}$. Hence, it seems hard to securely hash into $\mathbb{QR}_{N^2}$. In particular, using a hash function $\mathcal{H} : \{0,1\}^* \to \mathbb{Z}_{N^2}$ and squaring its output is inadequate in many settings since this reveals a square root of the output of the resulting hash function $\mathcal{H}' = \mathcal{H}^2 \bmod N^2$. However, it might be possible to use the group of signed quadratic residues $\mathbb{QR}^+_{N^2} = \mathbb{J}_{N^2}/\{-1, 1\}$ as in Section 4.3 in order to obtain an efficiently recognizable group with similar trapdoor DDH properties as $\mathbb{QR}_{N^2}$. Since the lack of perfect soundness of this TDDH group restricts the range of its applications, we do not pursue this possibility further.

4 Static Trapdoor DDH Groups

In this section, we define and construct *static* trapdoor DDH groups. They are similar to trapdoor DDH groups as defined in Section 3, except that the trapdoor only allows to solve the DDH problem with respect to a specific pair of group elements (G, G^x).

4.1 Definition

Definition 4. *A* static trapdoor DDH group $\mathcal{STDDH}$ *is a tuple of algorithms* $(\texttt{Gen}, \texttt{Samp}, \texttt{Solve})$ *with the following properties. The static trapdoor DDH group generator algorithm* $\texttt{Gen}$ *is a PPT algorithm which takes as input a security parameter* 1^k *and outputs a tuple* $([\mathbb{G}], G, \tau)$ *where* $[\mathbb{G}]$ *is the description of a group* $\mathbb{G}$*,* $G \in \mathbb{G}$ *is a group element of order* $2^{\Theta(k)}$*, and* τ *is a (master) trapdoor information, such that:*

i) hardness of DDH without the trapdoor*: the DDH problem is hard for the group generator* $\texttt{Gen}'$ *which outputs only* $([\mathbb{G}], G)$*.*

$\texttt{Samp}$ *is a PPT algorithm which on input* $([\mathbb{G}], G, \tau)$*, samples uniformly at random a group element* $X \leftarrow_\$ \langle G \rangle$*, and outputs*[7] (X, x, τ_x) *where* $x = \texttt{Dlog}_G(X)$ *and* τ_x *is a (static) trapdoor information, such that:*

ii) hardness of CDH with the static trapdoor*: for any PPT algorithm* $\mathcal{A}$*, the following probability is negligible:*

$$\Pr\left[([\mathbb{G}], G, \tau) \leftarrow \texttt{Gen}(1^k), (X, x, \tau_x) \leftarrow \texttt{Samp}([\mathbb{G}], G, \tau), Y \leftarrow_\$ \langle G \rangle, \right.$$
$$\left. Z \leftarrow \mathcal{A}([\mathbb{G}], G; X, Y; \tau_x) : Z = \texttt{CDH}_G(X, Y)\right] .$$

[6] We note however that imperfect soundness is not a problem for the identification scheme outlined in [DG06].

[7] We stress that in typical applications, x is retained by an authorized user and is never made available to the adversary.

$\texttt{Solve}$ *is a deterministic polynomial-time algorithm which takes as input* $([\mathbb{G}], G)$*, a tuple* $(X, Y, Z) \in \langle G \rangle \times \mathbb{G}^2$*, and the trapdoor* τ_x *for* X*, either accepts (outputs 1) or rejects (outputs 0), and satisfies the following:*

iii) completeness*: for all* $([\mathbb{G}], G, \tau)$ *and* (X, x, τ_x) *possibly output by* $\texttt{Gen}$ *and* $\texttt{Samp}$*, and any* $(Y, Z) \in \mathbb{G}^2$*,* $\texttt{Solve}$ *always accepts when* $(X, Y, Z) \in \mathcal{DH}_G$*;*

iv) soundness*: for any PPT adversary* $\mathcal{A}$*, the following probability is negligible:*

$$\Pr\left[([\mathbb{G}], G, \tau) \leftarrow \texttt{Gen}(1^k), (X, x, \tau_x) \leftarrow \texttt{Samp}([\mathbb{G}], G, \tau), Y \leftarrow_\$ \langle G \rangle, \right.$$
$$\left. Z \leftarrow \mathcal{A}([\mathbb{G}], G; X, Y) : 1 \leftarrow \texttt{Solve}([\mathbb{G}], G; X, Y, Z; \tau_x) \wedge (X, Y, Z) \notin \mathcal{DH}_G \right] .$$

We say that $\mathcal{STDDH}$ *has perfect soundness when* $\texttt{Solve}$ *always rejects on input a non-DH tuple* (X, Y, Z)*, so that the above probability is zero.*

Again, we silently assumed that $\texttt{Solve}$ is always run with the correct trapdoor τ_x because in all examples below this can be checked efficiently. In the remainder of this section, we propose two constructions of static TDDH groups based respectively on the RSA problem and the factoring problem.

4.2 A Construction Based on the RSA Problem

We first show how a static TDDH group can be obtained from the RSA problem. Let $N = pq$ be an RSA modulus. When $(p-1)/2$ and $(q-1)/2$ are coprime, then the subgroup $\mathbb{J}_N$ of $\mathbb{Z}_N^*$ is cyclic. Moreover, when p and q are distinct safe primes, the DDH problem is widely believed to be hard in $\mathbb{J}_N$ [Bon98]. We define the static TDDH group $\mathcal{STDDH}_{\texttt{RSA}}$ as follows. On input 1^k, the group generator $\texttt{Gen}_{\texttt{RSA}}$ selects two k-bit safe primes $p = 2p'+1$ and $q = 2q'+1$, defines $N = pq$ and $m = (p-1)(q-1)/2 = 2p'q'$, selects a generator G of $\mathbb{J}_N$, and outputs $([\mathbb{J}_N], G, \tau = m)$. The $\texttt{Samp}_{\texttt{RSA}}$ algorithm, on input $([\mathbb{J}_N], G, m)$, draws a random $x \leftarrow_\$ \mathbb{Z}_m^*$, computes $X = G^x$, $\tau_x = 1/x \bmod m$, and outputs (X, x, τ_x) (note that we slightly deviate from Definition 4 here since X is not uniformly random in $\langle G \rangle$, but the statistical distance is negligible). Algorithm $\texttt{Solve}_{\texttt{RSA}}$, on input $([\mathbb{J}_N], G; X, Y, Z; \tau_x)$, first checks that $X, Y, Z \in \mathbb{J}_N$, that the trapdoor is correct by verifying whether $X^{\tau_x} = G$ (it outputs $\perp$ if this does not hold), and outputs 1 iff $Z^{\tau_x} = Y$.

Definition 5. *We say that the RSA problem is hard for* $\mathbb{J}_N$ *if for any PPT adversary* $\mathcal{A}$*, the following probability is negligible:*

$$\Pr\left[([\mathbb{J}_N], G, m) \leftarrow \texttt{Gen}_{\texttt{RSA}}(1^k), e \leftarrow_\$ \mathbb{Z}_m^*, Y \leftarrow_\$ \mathbb{J}_N, Z \leftarrow \mathcal{A}([\mathbb{J}_N], Y, e) : Z^e = Y\right] .$$

Theorem 2. *Assuming that the DDH problem and the RSA problem are hard in* $\mathbb{J}_N$ *(for* N *the product of two distinct safe primes),* $\mathcal{STDDH}_{\texttt{RSA}}$ *is a static TDDH group with perfect soundness.*

Proof. We show that properties *i*) to *iv*) of Definition 4 hold. Property *i*) holds by assumption that DDH is hard for $\mathbb{J}_N$. We now prove property *ii*). Assume that there is an adversary $\mathcal{A}$ breaking property *ii*). We construct a reduction $\mathcal{R}$ that solves the RSA problem as follows. The reduction is given the product $N = pq$ of two safe primes, a random e coprime with $m = (p-1)(q-1)/2$, and a random challenge $Y \in \mathbb{J}_N$ of which it must compute the e-th root. The reduction draws a random $X \leftarrow_\$ \mathbb{J}_N$. With overwhelming probability, X is a generator of $\mathbb{J}_N$ since p and q are safe primes. The reduction defines $G = X^e$, and runs $\mathcal{A}$ on input $([\mathbb{J}_N], G; X, Y; e)$. The statistical distance between inputs (G, X, Y) in the simulated experiment and in the real CDH experiment defining property *ii*) is negligible (the difference coming from cases where X does not generate $\mathbb{J}_N$). Moreover, e is the correct trapdoor for X since $G = X^e$ implies $e = 1/x \bmod m$, where $x = \texttt{Dlog}_G(X)$. Hence, $\mathcal{A}$ returns the correct value $Z = \texttt{CDH}_G(X, Y)$ with probability negligibly close to its advantage, in which case $Z = Y^x$, which implies $Z^e = Y$, so that Z is indeed the e-th root of Y. The running time of $\mathcal{R}$ is similar to the one of $\mathcal{A}$ and its success probability is negligibly close to the one of $\mathcal{A}$. Property *iii*) is clear, and $\mathcal{STDDH}_{\texttt{RSA}}$ has perfect soundness since by definition of $\texttt{Samp}_{\texttt{RSA}}$, x is coprime to m so that $Z^{\tau_x} = Y \Leftrightarrow Z^{x\tau_x} = Y^x \Leftrightarrow Z = Y^x$. □

Hashing into $\mathbb{J}_N$. Hashing into $\mathbb{J}_N$ can be done easily as follows. Let $\mathcal{H}$ be a hash function into $\mathbb{Z}_N$ (which can be easily constructed from a hash function into $\{0,1\}^k$). Let $a \in \mathbb{Z}_N^*$ be a fixed integer such that $\left(\frac{a}{N}\right) = -1$. Neglecting the cases where $\mathcal{H}(x) \notin \mathbb{Z}_N^*$, define $\mathcal{H}'(x) = \mathcal{H}(x)$ if $\left(\frac{\mathcal{H}(x)}{N}\right) = 1$ and $\mathcal{H}'(x) = a\mathcal{H}(x) \bmod N$ if $\left(\frac{\mathcal{H}(x)}{N}\right) = -1$. When $\mathcal{H}$ is modeled as a random oracle, this construction can easily be shown indifferentiable from a random oracle into $\mathbb{J}_N$.

4.3 A Construction Based on Signed Quadratic Residues

In this section, we describe a static TDDH group based on signed quadratic residues, whose usefulness for cryptography was first noticed by Hofheinz and Kiltz [HK09]. This can be seen as a variant of $\mathcal{STDDH}_{\texttt{RSA}}$ described above, whose security relies on the factoring problem rather than the RSA problem. We first give some definitions.

Definition 6. *Let N be an odd positive integer such that $-1 \in \mathbb{J}_N$. We denote $\mathbb{J}_N^+$ the quotient group $\mathbb{J}_N/\{-1,1\}$. We identify $\mathbb{J}_N^+$ with the set $\mathbb{J}_N \cap [1;(N-1)/2]$ equipped with the group operation $\circ$ defined as $a \circ b = |ab \bmod N|$, where $|x \bmod N|$ is defined as the absolute value of $x \bmod N$ when representing elements of $\mathbb{Z}_N$ as integers in $[-(N-1)/2;(N-1)/2]$.*

To be completely rigorous, the mapping which to an element $\{-x,x\} \in \mathbb{J}_N^+$ associates $|x|$ is a group isomorphism between $\mathbb{J}_N^+$ and $(\mathbb{J}_N \cap [1,(N-1)/2], \circ)$.

Let $N = pq$ be a Blum integer (*i.e.* p and q are two primes such that $p \equiv q \equiv 3 \bmod 4$). Then $-1 \in \mathbb{J}_N$ so that we can define $\mathbb{J}_N^+$, which in this particular case is named the group of signed quadratic residues and denoted $\mathbb{QR}_N^+$.[8] Its order is $\phi(N)/4 = (p-1)(q-1)/4$. The most interesting points to notice about this group is that it is efficiently recognizable (since it is isomorphic to $\mathbb{J}_N \cap [1;(N-1)/2]$), and that the squaring operation is one-to-one so that any $x \in \mathbb{QR}_N^+$ has a unique square root in $\mathbb{QR}_N^+$ (more precisely, for any $x \in \mathbb{QR}_N^+$, either x or $-x \bmod N$ is a quadratic residue mod N, and exactly one corresponding square root is in $\mathbb{QR}_N^+$). Moreover, when $(p-1)/2$ and $(q-1)/2$ are coprime, then $\mathbb{J}_N$ is cyclic and so is $\mathbb{QR}_N^+$. See [HK09] for proofs of these basic facts.

In the following, we restrict ourselves for simplicity to the special case where N is the product of two distinct safe primes. This implies that N is a Blum integer, and that $(p-1)/2$ and $(q-1)/2$ are coprime so that $\mathbb{QR}_N^+$ is cyclic. Moreover, a uniformly random element of $\mathbb{QR}_N^+$ is a generator with overwhelming probability since the number of generators of $\mathbb{QR}_N^+$ is $\phi((p-1)(q-1)/4) = (p-3)(q-3)/4$.

Let G be a generator of $\mathbb{QR}_N^+$, and denote $m = |\mathbb{QR}_N^+| = (p-1)(q-1)/4$. Let $x \in [0;m-1]$ and $X = G^x$. To build a trapdoor enabling to solve the static DDH problem for (G,X), we use the following idea: the trapdoor will be $t = 2x \pm m$ (computed over $\mathbb{Z}$), *i.e.* the value $2x$ masked with the group order m. Since computing the group order m is as hard as factoring N, t does not reveal x. Now, given a group element $Y = G^y \in \mathbb{G}$, t enables computing $Y^t = G^{2xy} = \texttt{CDH}_G(X,Y)^2$. This enables testing whether an element Z is a correct solution to the static CDH problem (in other words to solve the static DDH problem) by simply checking whether $Z^2 = Y^t$. However, as we will see, the static CDH problem remains as hard as computing square roots in $\mathbb{QR}_N^+$, which in turn is equivalent to factoring N. For what follows, we will also make the assumption that the DDH problem is hard in $\mathbb{QR}_N^+$. The DDH problem in $\mathbb{QR}_N^+$ can easily shown to be equivalent to the DDH problem in $\mathbb{J}_N$, which as already pointed out is widely believed to be hard when N is the product of two distinct safe primes [Bon98].

We now formally define the static TDDH group $\mathcal{STDDH}_{\texttt{SQR}}$. For ease of exposition, given an odd integer m, we define the function ξ from $[0;m-1]$ to $\{1,3,5,\ldots,2m-3,2m-1\}$ as:

$$\begin{cases} \xi(x) = 2x + m \text{ if } x \in [0;(m-1)/2] \\ \xi(x) = 2x - m \text{ if } x \in [(m+1)/2;m-1] \end{cases}.$$

[8] We warn that $\mathbb{QR}_N^+$ is *not* equal to $\mathbb{QR}_N/\{-1,1\}$ for the good reason that $-1 \notin \mathbb{QR}_N$ when N is a Blum integer.

$\xi(x)$ is the unique odd integer $t \in [1; 2m-1]$ such that $t = 2x \pm m$.

On input the security parameter 1^k, $\mathtt{Gen}_{\mathtt{SQR}}$ selects two k-bit safe primes $p = 2p'+1$ and $q = 2q'+1$, sets $N = pq$, $m = p'q'$, selects a generator G of the group of signed quadratic residues $\mathbb{QR}_N^+$, and outputs $([\mathbb{QR}_N^+], G, m)$. Algorithm $\mathtt{Samp}_{\mathtt{SQR}}$, on input $([\mathbb{QR}_N^+], G, m)$, selects a random $x \in [0; m-1]$, sets $X = G^x$, $\tau_x = \xi(x)$, and outputs (X, x, τ_x). The algorithm $\mathtt{Solve}_{\mathtt{SQR}}$, on input $([\mathbb{QR}_N^+], G; X, Y, Z; \tau_x)$, first checks that the trapdoor is correct by verifying whether $G^{\tau_x} = X^2$ (it outputs $\perp$ if this does not hold), and outputs 1 iff $Y^{\tau_x} = Z^2$. We now formally prove that this constitutes a static TDDH group under appropriate assumptions (the proof of property *ii*) is reminiscent of the one of Theorem 3.2 in [HK09]).

Theorem 3. *Under the factoring assumption (for the product of safe primes) and the DDH assumption for $\mathbb{QR}_N^+$, $\mathcal{STDDH}_{\mathtt{SQR}}$ is a static TDDH group with perfect soundness.*

Proof. We show that properties *i*) to *iv*) of Definition 4 hold. Property *i*) holds by assumption that DDH is hard in $\mathbb{QR}_N^+$. We now show that property *ii*) holds under the factoring assumption (for the product of two safe primes). For this, we assume that there exists an adversary $\mathcal{A}$ solving the static CDH problem given the static trapdoor, and derive a reduction $\mathcal{R}$ that solves the factoring problem. The reduction is given a challenge $N = pq$ which is the product of two safe primes. Denote $(\mathbb{Z}_N^*)^+ = \mathbb{Z}_N^* \cap [1; (N-1)/2]$. The reduction chooses a uniformly random $u \leftarrow_{\$} (\mathbb{Z}_N^*)^+ \setminus \mathbb{QR}_N^+$, and sets $H = |u^2 \bmod N|$. Note that $H \in \mathbb{QR}_N^+$, and generates $\mathbb{QR}_N^+$ with overwhelming probability. It also chooses random integers $a, b \in [0; (N-1)/4]$ and sets $G = H^2$, $X = HG^a$, and $Y = HG^b$. Finally, it computes $t = 2a+1$ and runs $\mathcal{A}$ on input $([\mathbb{QR}_N^+], G; X, Y; t)$. The statistical distance between inputs (G, X, Y) in a real CDH experiment and in the simulated experiment is negligible (the difference coming from cases where H does not generate $\mathbb{QR}_N^+$ and the statistical distance between uniformly drawing in $[0; (N-1)/4]$ and $[0; m-1]$). Moreover, by definition of X and Y, we have that the discrete logarithms $x = \mathtt{Dlog}_G(X)$ and $y = \mathtt{Dlog}_G(Y)$ of these two elements are implicitly set as $x = a + 1/2 \bmod m$ and $y = b + 1/2 \bmod m$, where $m = |\mathbb{QR}_N^+| = (p-1)(q-1)/4$. Since m is odd, $1/2 = (m+1)/2 \bmod m$, so that when $a \in [0; (m-3)/2]$, we have $x = a + (m+1)/2$ (over the integers), whereas when $a \in [(m-1)/2, m-1]$, we have $x = a - (m-1)/2$ (over the integers). In both cases, we see that $t = 2a+1$ is the unique integer in $[1; 2m-1]$ such that $t = 2x \pm m$, so that we always have $t = \xi(x)$. Note that we neglected the case where $a \in [m+1; (N-1)/4]$ but this happens only with negligible probability.

By the previous analysis, we have that $\mathcal{A}$ returns the correct value $Z = \mathtt{CDH}_G(X, Y)$ with probability negligibly close to its advantage. In such a case, we have:

$$Z = G^{xy} = G^{(a+1/2)(b+1/2) \bmod m} = H^{2ab+a+b+1/2 \bmod m},$$

and consequently, defining $V = ZG^{-2ab-a-b} = H^{1/2 \bmod m}$, $\mathcal{R}$ has obtained a square root V of H. Since this square root is in $\mathbb{QR}_N^+$, it is necessarily different from $u \in (\mathbb{Z}_N^*)^+ \setminus \mathbb{QR}_N^+$ and $-u \in [-(N-1)/2; 1]$. Hence, $\gcd(u - V, N)$ yields a factor of N. The running time of $\mathcal{R}$ is similar to the one of $\mathcal{A}$ and its success probability is negligibly close to the one of $\mathcal{A}$. Finally, property *iii*) is clear, and $\mathcal{STDDH}_{\mathtt{SQR}}$ has perfect soundness (*i.e.* $\mathtt{Solve}_{\mathtt{SQR}}$ accepts iff it is run on input $(X, Y, Z) \in \mathcal{DH}_G$) since the squaring operation is one-to-one, so that given $X = G^x$ and $Y = G^y$, $Z = G^{xy}$ iff $Z^2 = G^{2xy} = Y^t$. □

Hashing into $\mathbb{QR}_N^+$. Hashing into $\mathbb{QR}_N^+$ can be done similarly to the method described for $\mathbb{J}_N$. Let $\mathcal{H} : \{0,1\}^* \to [0; (N-1)/2]$ be a hash function, and $a \in \mathbb{Z}_N^* \cap [1; (N-1/2)]$ be a fixed integer such that $\left(\frac{a}{N}\right) = -1$. Neglecting the cases where $\mathcal{H}(x) \notin \mathbb{Z}_N^*$, define:

$$\mathcal{H}'(x) = \begin{cases} \mathcal{H}(x) & \text{if } \left(\frac{\mathcal{H}(x)}{N}\right) = 1 \\ |a\mathcal{H}(x) \bmod N| & \text{if } \left(\frac{\mathcal{H}(x)}{N}\right) = -1 \end{cases}$$

Modeling $\mathcal{H}$ as a random oracle, then $\mathcal{H}'$ can be shown indifferentiable from a random oracle into $\mathbb{QR}_N^+$. Choosing two safe primes such that $p \equiv 3 \bmod 8$ and $q \equiv 7 \bmod 8$, then one can always choose $a = 2$.

4.4 Relation to the Strong Diffie-Hellman Problem

We note that in a static TDDH group with perfect soundness, the Strong Diffie-Hellman (SDH) problem [ABR01] is always hard.[9] The SDH problem is to compute $\mathtt{CDH}_G(X, Y)$ given $X, Y \in \langle G \rangle$, and being granted access to a static DDH oracle which on input $(Y', Z') \in \mathbb{G}^2$ outputs 1 iff $(X, Y', Z') \in \mathcal{DH}_G$. Clearly, an adversary $\mathcal{A}$ breaking the SDH problem can be turned into an adversary $\mathcal{B}$ breaking property *ii*) of the static TDDH group ($\mathcal{B}$ can answer queries of $\mathcal{A}$ to the static DDH oracle thanks to the trapdoor τ_x it is given as input). Applying this observation to $\mathcal{STDDH}_{\mathtt{SQR}}$, we recover Theorem 3.2 of [HK09] which states that SDH is hard in $\mathbb{QR}_N^+$ under the factoring assumption. Hence, the concept of static TDDH group allows to cast the result of [HK09] in a more general framework. In particular, Theorem 2 directly implies that under the RSA assumption, the SDH problem is hard in $\mathbb{J}_N$, which complements the result of [HK09].[10] As an immediate consequence of the results of [ABR01, CS03b], we obtain that Hybrid ElGamal encryption over $\mathbb{J}_N$ is IND-CCA2-secure in the ROM under the RSA assumption.

5 Convertible Undeniable Signatures

5.1 Background on Undeniable Signatures

In this section, we show how TDDH groups can be used to build simple and natural undeniable signature schemes with attractive properties such as universal convertibility and delegation. Undeniable signatures, introduced by Chaum and van Antwerpen [CvA89], are signatures that cannot be universally verified: confirmation (or disavowal) of a signature requires the cooperation of the signer (however a signer cannot deny the validity of a correct signature, hence the name undeniable). Later, Boyar *et al.* [BCDP90] proposed the refined notion of *convertible* undeniable signature (CUS) scheme, where a mechanism allows the signer to selectively or globally transform undeniable signatures into self-authenticating signatures. The particular scheme proposed in [BCDP90] was later broken in [MPH96]. Subsequently, schemes based on usual signatures such as ElGamal [DP96], Schnorr [MS97], and RSA [GRK00, GMP02, GM03] were proposed.

We first recall the basic Chaum and van Antwerpen undeniable signature scheme [CvA89] (in its Full Domain Hash version [OP01, OKH05]). Let $\mathbb{G}$ be a group, $\mathbb{G}'$ be a cyclic and efficiently recognizable subgroup of $\mathbb{G}$, G be a (certified) generator of $\mathbb{G}'$, and $\boldsymbol{H} : \{0,1\}^* \to \mathbb{G}'$ be a hash function (modeled as a random oracle in security proofs). Assume the DDH problem is hard for $\mathbb{G}'$. The secret and public keys of a user are $x \in \mathbb{Z}_{|\mathbb{G}'|^+}$ and $X = G^x$ respectively. To sign a message $\mu \in \{0,1\}^*$, the signer computes $M = \boldsymbol{H}(\mu) \in \mathbb{G}'$, and $S = M^x$. The signature is S. A signature S on μ is valid iff $(X, \boldsymbol{H}(\mu), S)$ is a valid DH tuple (with respect to G). Since we assumed that the DDH problem is hard, checking the validity of a signature cannot be done without knowledge of x.[11] Hence, the signer must cooperate with the verifier in order to confirm or disavow a purported signature. The confirmation protocol is a proof that $(X, \boldsymbol{H}(\mu), S) \in \mathcal{DH}_G$ (*i.e.* a proof of EDL since G is guaranteed to be a generator of $\mathbb{G}'$), whereas the disavowal protocol is a proof that $(X, \boldsymbol{H}(\mu), S) \notin \mathcal{DH}_G$ (*i.e.* a proof of IDL). The security of this scheme (depending on which type of EDL and IDL proofs are used) has been studied in [OP01, KH05, OKH05, KF08].

The idea to allow efficient universal conversion of signatures is simply to use a Chaum and van Antwerpen undeniable signature with a (static or not) TDDH group, and to use the trapdoor to delegate the ability to verify undeniable signatures and to universally convert them. In the following, we describe the construction using static TDDH groups since the instantiations using constructions of Sections 4.2 and 4.3 are particularly interesting.

[9] More precisely, the SDH problem is hard for the group generator which only outputs $([\mathbb{G}], G)$.

[10] Note that, by inspection of the proof of property *ii*), this result holds in fact for all RSA moduli N such that $\mathbb{J}_N$ is cyclic, not only the product of safe primes.

[11] When the DDH problem is easy in $\mathbb{G}'$, signatures can be universally verified. For example, using bilinear groups (where a pairing can be used to solve the DDH problem), one obtains the Boneh-Lynn-Shacham signature scheme [BLS04].

5.2 Construction of a CUS scheme from a Static TDDH Group

Let $\mathcal{STDDH} = (\mathtt{Gen}, \mathtt{Samp}, \mathtt{Solve})$ be a static TDDH group with perfect soundness. For this part, we assume that $\mathtt{Gen}$ outputs a tuple $([\mathbb{G}], G, \tau)$ such that $\mathbb{G}$ is cyclic and efficiently recognizable, and G is a generator of $\mathbb{G}$. This assumption is satisfied by $\mathcal{STDDH}_{\mathtt{RSA}}$ and $\mathcal{STDDH}_{\mathtt{SQR}}$. Note that there is not necessarily an efficient way to check that G is indeed a generator; we come back on this issue later. We construct a CUS scheme $\mathcal{CUS}$ as follows (see Appendix C for a more formal description). To construct his public/secret key pair, the signer runs $\mathtt{Gen}(1^k)$ to obtain $([\mathbb{G}], G, \tau)$ and then $\mathtt{Samp}([\mathbb{G}], G, \tau)$ to obtain (X, x, τ_x). It also selects a hash function $\boldsymbol{H} : \{0,1\}^* \to \mathbb{G}$. The public key of the signer is $\mathtt{pk} = ([\mathbb{G}], G, X, \boldsymbol{H})$ and its secret key is $\mathtt{sk} = (x, \tau_x)$. To sign a message $\mu \in \{0,1\}^*$, the signer computes $M = \boldsymbol{H}(\mu)$, and $S = M^x$. The signature is S. The signer can confirm or disavow a signature by running a proof of EDL or IDL respectively with the verifier. To individually convert a signature, the signer produces a NIZK proof of EDL (using an independent hash function $\boldsymbol{H}_{\mathrm{FS}}$ to apply the Fiat-Shamir transform). To universally convert signatures, the signer releases τ_x as universal receipt. A signature S for message μ can then be verified by running $\mathtt{Solve}([\mathbb{G}], G; X, \boldsymbol{H}(\mu), S; \tau_x)$.

Informally, the two main security properties of a CUS scheme (beside soundness of the confirmation and disavowal protocols) are (see Appendix C.2 for details):

- *security against existential forgery under chosen-message attacks (EF-CMA-security)*: any PPT attacker, given the receipt for universal verification τ_x, and with access to a signing oracle, can forge a new signature with only negligible probability (note that access to confirmation or disavowal oracles is unnecessary here since the adversary is given the universal receipt τ_x for checking signatures);
- *invisibility under chosen-message attacks (INV-CMA-security)*: any PPT adversary can distinguish a valid signature for a message of its choice from a string sampled uniformly at random from the signature space with only negligible probability. The adversary is granted access to the signing oracle, the confirmation and disavowal protocols, and the individual signature conversion oracle (with the restriction that they cannot be queried on the challenge message).

We stress that formalizing the invisibility notion is quite subtle (many variations appear in the literature [CvHP91, DP96, CM00, GM03]), and that the exact property that is achieved is dependent on the nature of the confirmation and disavowal protocols [OKH05, KF08].

Theorem 4. *When instantiated with a static TDDH group with perfect soundness, and when the confirmation and disavowal protocols are zero-knowledge, the CUS scheme described above is EF-CMA-secure and INV-CMA-secure in the ROM (for $\boldsymbol{H}$ and $\boldsymbol{H}_{\mathrm{FS}}$).*

Proof. We first show EF-CMA-security. Assume there is an adversary $\mathcal{A}$ breaking EF-CMA-security of the scheme. We build a reduction $\mathcal{R}$ that breaks property *ii*) of the static TDDH group $\mathcal{STDDH}$. Let q_h be an upper bound on the number of queries to $\boldsymbol{H}$ made by $\mathcal{A}$. The reduction is given as input $([\mathbb{G}], G; X, Y; \tau_x)$. It runs $\mathcal{A}$ on input $\mathtt{pk} = ([\mathbb{G}], G, X, \boldsymbol{H})$ and $\rho_u = \tau_x$. It simulates the random oracle $\boldsymbol{H}$ as usual in security proofs for Full Domain Hash, namely it guesses which oracle query will be used by $\mathcal{A}$ to forge a signature, and uses Y as answer to this query. Other answers to random oracle queries are computed as G^α for known values α, which enable the reduction to simulate the signing oracle by computing the signatures as X^α. When the guess for the forgery was right, the reduction obtains a signature S which is exactly $Y^x = \mathtt{CDH}_G(X, Y)$. The success probability of the reduction is therefore $\mathbf{Adv}^{\mathrm{ef-cma}}_{\mathcal{CUS},\mathcal{A}}(k)/q_h$. Details are standard and therefore omitted.

The proof for invisibility is exactly the same as the proof of Theorem 7 of [OKH05] (the sole difference is that here we have to simulate the individual conversion oracle but as usual this can be done without knowing $\mathtt{Dlog}_G(X)$ using the zero-knowledge simulator). □

Delegation. The ability to verify (confirm or disavow) and convert (either individually or universally) signatures can easily be delegated to a semi-trusted party by simply giving him the trapdoor

τ_x. Since the CDH problem remains hard even with the trapdoor, the third party cannot forge signatures on behalf of the signer. It can however prove in zero-knowledge whether a signature is valid or invalid (since it knows the witness τ_x for this). We avoid using the term *designated confirmer signatures* [Cha94] here since this usually refers to schemes (mostly following the "encryption of a signature" paradigm [Oka94, CM00]) where the signer can create designated confirmer undeniable signatures without having beforehand to transmit some secret information to the confirmer (in our case the trapdoor τ_x).

Instantiation with $\mathcal{STDDH}_{\mathtt{RSA}}$ and $\mathcal{STDDH}_{\mathtt{SQR}}$. The CUS scheme described above can be instantiated with the two static TDDH groups described in Sections 4.2 and 4.3. The schemes obtained this way are similar respectively to the scheme of Gennaro, Rabin, and Krawczyk [GRK00] and Galbraith and Mao [GM03], with important distinctions though. Both schemes work over $\mathbb{Z}_N^*$, but without explicitly restricting in which subgroup. As a consequence, they cannot be exactly seen as an instantiation of the Chaum and van Antwerpen scheme, and specific confirmation and disavowal protocols were therefore proposed for them (see also [GMP02]). On the contrary, our schemes are strict instantiations of the Chaum and van Antwerpen scheme, and in particular the confirmation and disavowal protocols can use zero-knowledge proofs of EDL and IDL derived from the HVZK protocols described in Appendix A. This is conceptually simpler and more efficient (especially for the disavowal protocol).

Certifying signers public keys. Correct key generation is of primary importance in factoring-based undeniable signatures, since a cheating signer may generate its secret/public key in a different way than the one expected by verifiers, which may enable him to confirm invalid signatures or disavow valid ones (see [GMP02]). Hence, the signer, when registering his public key, must prove to the certification authority (CA) that it was generated according to the specification of the static TDDH group generator. We now discuss this issue with respect to $\mathcal{STDDH}_{\mathtt{RSA}}$ and $\mathcal{STDDH}_{\mathtt{SQR}}$. For both schemes, the signer must first prove to the CA that its modulus N is the product of two safe primes. A zero-knowledge protocol for this was proposed by Camenish and Michels [CM99]. Though expensive, this protocol must be run only once at key registration time. Then, the signer must prove that G is indeed a generator of either $\mathbb{J}_N$ or $\mathbb{QR}_N^+$. The situation is slightly different in the two cases. Denote $N = pq$ with $p = 2p' + 1$ and $q = 2q' + 1$. When p and q are safe primes, then an integer $g \in \mathbb{Z}_N^*$ such that $g^2 \neq 1 \bmod N$ and $\gcd(g^2 - 1, N) = 1$ necessarily has order in $\{p'q', 2p'q'\}$ [GRK00, Lemma 1]. Hence, an ad-hoc solution for ensuring that the element G provided by the signer is a generator of the intended group is as follows. Restrict the scheme to moduli N such that $N \equiv 1 \bmod 8$ and fix $g_0 = 2$ so that $g_0 \in \mathbb{J}_N$. Since an element $g \in \mathbb{QR}_N^+$ generates $\mathbb{QR}_N^+$ exactly when g has multiplicative order modulo N in $\{p'q', 2p'q'\}$, we see by the previous remark that g_0 is always a generator of $\mathbb{QR}_N^+$. Hence, when using $\mathcal{STDDH}_{\mathtt{SQR}}$, we can impose to the signer to always use $G = g_0$. Things are a bit more complicated when using $\mathcal{STDDH}_{\mathtt{RSA}}$, since for an element $g \in \mathbb{Z}_N^*$ with order in $\{p'q', 2p'q'\}$ to generate $\mathbb{J}_N$, one has to check that it is a quadratic non-residue. What we propose for this is that the signer proves in zero-knowledge to the CA whether $g_0 \in \mathbb{QR}_N$ or not [GMR89]. If it is in $\mathbb{QR}_N$, then the signer tries with $g_0 + 1$, $g_0 + 2$, etc. until a quadratic non-residue in $\mathbb{J}_N$ is found. The signer then has to use $G = g_0 + i$ for the smallest $i \geq 0$ such that $g_0 + i \in \mathbb{J}_N \setminus \mathbb{QR}_N$.

As a matter of fact, there seems to be no reason to instantiate the CUS scheme with $\mathcal{STDDH}_{\mathtt{RSA}}$ rather than $\mathcal{STDDH}_{\mathtt{SQR}}$ since both schemes are almost identical, except that the key registration step is simpler for $\mathcal{STDDH}_{\mathtt{SQR}}$.

Acknowledgments

The author is thankful to anonymous reviewers of PKC 2013 for helpful comments and suggestions.

References

[ABR01] Michel Abdalla, Mihir Bellare, and Phillip Rogaway. The Oracle Diffie-Hellman Assumptions and an Analysis of DHIES. In David Naccache, editor, *Topics in Cryptology - CT-RSA 2001*, volume 2020 of *Lecture Notes in Computer Science*, pages 143–158. Springer, 2001.

[BCDP90] Joan Boyar, David Chaum, Ivan Damgård, and Torben P. Pedersen. Convertible Undeniable Signatures. In Alfred Menezes and Scott A. Vanstone, editors, *Advances in Cryptology - CRYPTO '90*, volume 537 of *Lecture Notes in Computer Science*, pages 189–205. Springer, 1990.

[BCI+10] Eric Brier, Jean-Sébastien Coron, Thomas Icart, David Madore, Hugues Randriam, and Mehdi Tibouchi. Efficient Indifferentiable Hashing into Ordinary Elliptic Curves. In Tal Rabin, editor, *Advances in Cryptology - CRYPTO 2010*, volume 6223 of *Lecture Notes in Computer Science*, pages 237–254. Springer, 2010.

[BCP03] Emmanuel Bresson, Dario Catalano, and David Pointcheval. A Simple Public-Key Cryptosystem with a Double Trapdoor Decryption Mechanism and Its Applications. In Chi-Sung Laih, editor, *Advances in Cryptology - ASIACRYPT 2003*, volume 2894 of *Lecture Notes in Computer Science*, pages 37–54. Springer, 2003.

[BF01] Dan Boneh and Matthew K. Franklin. Identity-Based Encryption from the Weil Pairing. In Joe Kilian, editor, *Advances in Cryptology - CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer, 2001.

[BLS04] Dan Boneh, Ben Lynn, and Hovav Shacham. Short Signatures from the Weil Pairing. *Journal of Cryptology*, 17(4):297–319, 2004.

[Bon98] Dan Boneh. The Decision Diffie-Hellman Problem. In Joe Buhler, editor, *Algorithmic Number Theory Symposium - ANTS '98*, volume 1423 of *Lecture Notes in Computer Science*, pages 48–63. Springer, 1998.

[CDM00] Ronald Cramer, Ivan Damgård, and Philip D. MacKenzie. Efficient Zero-Knowledge Proofs of Knowledge Without Intractability Assumptions. In Hideki Imai and Yuliang Zheng, editors, *Public Key Cryptography - PKC 2000*, volume 1751 of *Lecture Notes in Computer Science*, pages 354–373. Springer, 2000.

[Cha94] David Chaum. Designated Confirmer Signatures. In Alfredo De Santis, editor, *Advances in Cryptology - EUROCRYPT '94*, volume 950 of *Lecture Notes in Computer Science*, pages 86–91. Springer, 1994.

[CM99] Jan Camenisch and Markus Michels. Proving in Zero-Knowledge that a Number Is the Product of Two Safe Primes. In Jacques Stern, editor, *Advances in Cryptology - EUROCRYPT '99*, volume 1592 of *Lecture Notes in Computer Science*, pages 107–122. Springer, 1999.

[CM00] Jan Camenisch and Markus Michels. Confirmer Signature Schemes Secure against Adaptive Adversaries. In Bart Preneel, editor, *Advances in Cryptology - EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 243–258. Springer, 2000.

[CP92] David Chaum and Torben P. Pedersen. Wallet Databases with Observers. In Ernest F. Brickell, editor, *Advances in Cryptology - CRYPTO '92*, volume 740 of *Lecture Notes in Computer Science*, pages 89–105. Springer, 1992.

[CS03a] Jan Camenisch and Victor Shoup. Practical Verifiable Encryption and Decryption of Discrete Logarithms. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 126–144. Springer, 2003.

[CS03b] Ronald Cramer and Victor Shoup. Design and Analysis of Practical Public-Key Encryption Schemes Secure against Adaptive Chosen Ciphertext Attack. *SIAM Journal on Computing*, 33(1):167–226, 2003.

[CvA89] David Chaum and Hans van Antwerpen. Undeniable Signatures. In Gilles Brassard, editor, *Advances in Cryptology - CRYPTO '89*, volume 435 of *Lecture Notes in Computer Science*, pages 212–216. Springer, 1989.

[CvHP91] David Chaum, Eugène van Heijst, and Birgit Pfitzmann. Cryptographically Strong Undeniable Signatures, Unconditionally Secure for the Signer. In Joan Feigenbaum, editor, *Advances in Cryptology - CRYPTO '91*, volume 576 of *Lecture Notes in Computer Science*, pages 470–484. Springer, 1991.

[Dam00] Ivan Damgård. Efficient Concurrent Zero-Knowledge in the Auxiliary String Model. In Bart Preneel, editor, *Advances in Cryptology - EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 418–430. Springer, 2000.

[DF89] Yvo Desmedt and Yair Frankel. Threshold Cryptosystems. In Gilles Brassard, editor, *Advances in Cryptology - CRYPTO '89*, volume 435 of *Lecture Notes in Computer Science*, pages 307–315. Springer, 1989.

[DG06] Alexander W. Dent and Steven D. Galbraith. Hidden Pairings and Trapdoor DDH Groups. In Florian Hess, Sebastian Pauli, and Michael E. Pohst, editors, *Algorithmic Number Theory Symposium - ANTS 2006*, volume 4076 of *Lecture Notes in Computer Science*, pages 436–451. Springer, 2006.

[DP96] Ivan Damgård and Torben P. Pedersen. New Convertible Undeniable Signature Schemes. In Ueli M. Maurer, editor, *Advances in Cryptology - EUROCRYPT '96*, volume 1070 of *Lecture Notes in Computer Science*, pages 372–386. Springer, 1996.

[FMR99] Gerhard Frey, Michael Müller, and Hans-Georg Rück. The Tate pairing and the discrete logarithm applied to elliptic curve cryptosystems. *IEEE Transactions on Information Theory*, 45(5):1717–1719, 1999.

[Fre98] Gerhard Frey. How to disguise an elliptic curve (Weil descent). Elliptic Curve Cryptography - ECC '98, 1998. Available at http://cacr.uwaterloo.ca/conferences/1998/ecc98/frey.ps.

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

[Gen04] Rosario Gennaro. Multi-trapdoor Commitments and Their Applications to Proofs of Knowledge Secure Under Concurrent Man-in-the-Middle Attacks. In Matthew K. Franklin, editor, *Advances in Cryptology - CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 220–236. Springer, 2004.

[Gir90] Marc Girault. An Identity-based Identification Scheme Based on Discrete Logarithms Modulo a Composite Number. In Ivan Damgård, editor, *Advances in Cryptology - EUROCRYPT '90*, volume 473 of *Lecture Notes in Computer Science*, pages 481–486. Springer, 1990.

[Gir91] Marc Girault. Self-Certified Public Keys. In Donald W. Davies, editor, *Advances in Cryptology - EUROCRYPT '91*, volume 547 of *Lecture Notes in Computer Science*, pages 490–497. Springer, 1991.

[GM03] Steven D. Galbraith and Wenbo Mao. Invisibility and Anonymity of Undeniable and Confirmer Signatures. In Marc Joye, editor, *Topics in Cryptology - CT-RSA 2003*, volume 2612 of *Lecture Notes in Computer Science*, pages 80–97. Springer, 2003.

[GMP02] Steven D. Galbraith, Wenbo Mao, and Kenneth G. Paterson. RSA-Based Undeniable Signatures for General Moduli. In Bart Preneel, editor, *Topics in Cryptology - CT-RSA 2002*, volume 2271 of *Lecture Notes in Computer Science*, pages 200–217. Springer, 2002.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The Knowledge Complexity of Interactive Proof Systems. *SIAM Journal on Computing*, 18(1):186–208, 1989.

[GRK00] Rosario Gennaro, Tal Rabin, and Hugo Krawczyk. RSA-Based Undeniable Signatures. *Journal of Cryptology*, 13(4):397–416, 2000.

[GSV98] Oded Goldreich, Amit Sahai, and Salil P. Vadhan. Honest-Verifier Statistical Zero-Knowledge Equals General Statistical Zero-Knowledge. In Jeffrey Scott Vitter, editor, *Symposium on the Theory of Computing - STOC '98*, pages 399–408. ACM, 1998.

[HK09] Dennis Hofheinz and Eike Kiltz. The Group of Signed Quadratic Residues and Applications. In Shai Halevi, editor, *Advances in Cryptology - CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 637–653. Springer, 2009.

[Jou00] Antoine Joux. A One Round Protocol for Tripartite Diffie-Hellman. In Wieb Bosma, editor, *Algorithmic Number Theory Symposium - ANTS 2000*, volume 1838 of *Lecture Notes in Computer Science*, pages 385–394. Springer, 2000.

[KF08] Kaoru Kurosawa and Jun Furukawa. Universally Composable Undeniable Signature. In Ivan Damgård, editor, *International Colloquium on Automata, Languages and Programming - ICALP 2008, Track C: Security and Cryptography Foundations*, volume 5126 of *Lecture Notes in Computer Science*, pages 524–535. Springer, 2008.

[KH05] Kaoru Kurosawa and Swee-Huay Heng. 3-Move Undeniable Signature Scheme. In Ronald Cramer, editor, *Advances in Cryptology - EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 181–197. Springer, 2005.

[Mor08] David J. Mireles Morales. An attack on disguised elliptic curves. *Journal of Mathematical Cryptology*, 2(1):1–8, 2008. Available at http://eprint.iacr.org/2006/469.pdf.

[MOV93] Alfred Menezes, Tatsuaki Okamoto, and Scott A. Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Transactions on Information Theory*, 39(5):1639–1646, 1993.

[MPH96] Markus Michels, Holger Petersen, and Patrick Horster. Breaking and Repairing a Convertible Undeniable Signature Scheme. In Li Gong and Jacques Stearn, editors, *ACM Conference on Computer and Communications Security - CCS '96*, pages 148–152. ACM, 1996.

[MRH04] Ueli M. Maurer, Renato Renner, and Clemens Holenstein. Indifferentiability, Impossibility Results on Reductions, and Applications to the Random Oracle Methodology. In Moni Naor, editor, *Theory of Cryptography Conference- TCC 2004*, volume 2951 of *Lecture Notes in Computer Science*, pages 21–39. Springer, 2004.

[MS97] Markus Michels and Markus Stadler. Efficient Convertible Undeniable Signature Schemes. In *Selected Areas in Cryptography - SAC '97*, pages 231–244, 1997.

[Oka94] Tatsuaki Okamoto. Designated Confirmer Signatures and Public-Key Encryption are Equivalent. In Yvo Desmedt, editor, *Advances in Cryptology - CRYPTO '94*, volume 839 of *Lecture Notes in Computer Science*, pages 61–74. Springer, 1994.

[OKH05] Wakaha Ogata, Kaoru Kurosawa, and Swee-Huay Heng. The Security of the FDH Variant of Chaum's Undeniable Signature Scheme. In Serge Vaudenay, editor, *Public Key Cryptography - PKC 2005*, volume 3386 of *Lecture Notes in Computer Science*, pages 328–345. Springer, 2005.

[OP01] Tatsuaki Okamoto and David Pointcheval. The Gap-Problems: A New Class of Problems for the Security of Cryptographic Schemes. In Kwangjo Kim, editor, *Public Key Cryptography - PKC 2001*, volume 1992 of *Lecture Notes in Computer Science*, pages 104–118. Springer, 2001.

[Pai99] Pascal Paillier. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In Jacques Stern, editor, *Advances in Cryptology - EUROCRYPT '99*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–238. Springer, 1999.

[PX09] Manoj Prabhakaran and Rui Xue. Statistically Hiding Sets. In Marc Fischlin, editor, *Topics in Cryptology - CT-RSA 2009*, volume 5473 of *Lecture Notes in Computer Science*, pages 100–116. Springer, 2009.

[Sch91] Claus-Peter Schnorr. Efficient Signature Generation by Smart Cards. *Journal of Cryptology*, 4(3):161–174, 1991.

[TY98] Yiannis Tsiounis and Moti Yung. On the Security of ElGamal Based Encryption. In Hideki Imai and Yuliang Zheng, editors, *Public Key Cryptography - PKC '98*, volume 1431 of *Lecture Notes in Computer Science*, pages 117–134. Springer, 1998.

A HVZK Proof of EDL and IDL in Groups of Unknown Order

In all the following, we let $\mathbb{G}$ denote the ambient group, $\mathbb{G}'$ be a cyclic and efficiently recognizable subgroup of $\mathbb{G}$, and G be a certified generator of $\mathbb{G}'$. We denote $k_G = \lceil \log |\mathbb{G}'|^+ \rceil$, and we interpret strings in $\{0,1\}^\ell$ as integers in $[0; 2^\ell - 1]$.

A.1 Proof of EDL

Common input: Security parameter k, $([\mathbb{G}], G, X, Y, Z)$ where $X, Y, Z \in \mathbb{G}'$
Prover private input: $x = \texttt{Dlog}_G(X)$

- Prover: draw a random $r \leftarrow \{0,1\}^{k_G+2k}$, compute $A = G^r$ and $B = Y^r$, and send (A, B) to the verifier
- Verifier: upon reception of (A, B), check that $A, B \in \mathbb{G}'$, draw a random $c \leftarrow_\$ \{0,1\}^k$, and send c to the prover
- Prover: check that $c \in [0; 2^k - 1]$, compute $s = r + cx$ and send s to the verifier
- Verifier: Check that $G^s = AX^c$ and $Y^s = BZ^c$, and accept iff both equalities hold

It can be checked that this proof system is sound, complete, and statistically HVZK.

A.2 Proof of IDL

Common input: Security parameter k, $([\mathbb{G}], G, X, Y, Z)$ where $X, Y, Z \in \mathbb{G}'$
Prover private input: $x = \texttt{Dlog}_G(X)$

- Prover: draw randomly $t \leftarrow \{0,1\}^k$, $r \leftarrow \{0,1\}^{k_G+3k}$, and $r' \leftarrow \{0,1\}^{k_G+2k}$, compute $W = (Y^x/Z)^t$, $A = G^r/X^{r'}$, and $B = Y^r/Z^{r'}$, and send (W, A, B) to the verifier
- Verifier: upon reception of (W, A, B), check that $W, A, B \in \mathbb{G}'$ and $W \neq 1$, draw a random $c \leftarrow_\$ \{0,1\}^k$, and send c to the prover
- Prover: check that $c \in [0; 2^k - 1]$, compute $s = r + ctx$ and $s' = r' + ct$, and send (s, s') to the verifier
- Verifier: Check that $G^s/X^{s'} = A$ and $Y^s/Z^{s'} = BW^c$, and accept iff both equalities hold

It can be checked that this proof system is sound, complete, and statistically HVZK.

B A TDDH Group Based on (Hidden) Pairings

This construction was proposed by Dent and Galbraith in their original paper [DG06]. Let $N = p_1 p_2$ be the product of two primes such that $p_1 \equiv p_2 \equiv 3 \bmod 4$, and such that there are two large primes r_1 and r_2 such that $r_1 | (p_1 + 1)$ and $r_2 | (p_2 + 1)$. Let $E : y^2 = x^3 + x$ be an elliptic curve over the ring $\mathbb{Z}_N$. Then $|E(\mathbb{Z}_N)| = (p_1 + 1)(p_2 + 1)$. Let $P = (x_P, y_P) \in E(\mathbb{Z}_N)$ be a point of order $r_1 r_2$. Then the group generator outputs $([E(\mathbb{Z}_N)], P, \tau)$ where the trapdoor is $\tau = (p_1, p_2, r_1, r_2)$. By the Chinese Remainder Theorem, a tuple $(X, Y, Z) \in E(\mathbb{Z}_N)^3$ is a DDH tuple iff the elements reduce modulo p_1 and p_2 to valid DDH tuples in $E(\mathbb{F}_{p_1})$ and $E(\mathbb{F}_{p_2})$ respectively. Hence, to solve the DDH problem given τ, algorithm `Solve` solves the DDH problem in each group $E(\mathbb{F}_{p_i})$ using the Weil or Tate pairing [MOV93, FMR99]. The resulting TDDH group has perfect soundness, and we conjecture that this TDDH group satisfies Definition 1 (namely that CDH remains hard given the trapdoor). To the best of our knowledge, there is no known secure way to hash into $\langle P \rangle$.

C Convertible Undeniable Signatures

C.1 Syntactic Definition

A Convertible Undeniable Signature (CUS for short) scheme $\mathcal{CUS}$ is specified by the following set of algorithms and protocols:

- $\texttt{KeyGen}(1^k)$: A randomized algorithm which, on input the security parameter 1^k, outputs a public/secret key pair $(\texttt{pk}, \texttt{sk})$ for the signer.
- $\texttt{USign}(\texttt{pk}, \texttt{sk}, \mu)$: A potentially randomized algorithm which takes as input a public/secret key pair $(\texttt{pk}, \texttt{sk})$ and a message $\mu \in \{0,1\}^*$, and returns an undeniable signature σ. Given a public/secret key pair $(\texttt{pk}, \texttt{sk})$ and a message μ, we say that a signature σ is a *valid* signature for m under $(\texttt{pk}, \texttt{sk})$ if σ is in the support of $\texttt{USign}(\texttt{pk}, \texttt{sk}, \mu)$, and *invalid* otherwise. When USign is deterministic, this boils down to $\texttt{USign}(\texttt{pk}, \texttt{sk}, \mu) = \sigma$. We implicitly assume that there is an efficient algorithm which on input $(\texttt{pk}, \texttt{sk}, \mu, \sigma)$ decides whether σ is a valid signature for m under $(\texttt{pk}, \texttt{sk})$ (this is always the case when USign is deterministic).
- $\Pi_{\text{con}} = (\mathcal{P}_{\text{con}}, \mathcal{V}_{\text{con}})$: The confirmation protocol which is run between the signer (with private input sk) and a verifier, on common input $(\texttt{pk}, \mu, \sigma)$. At the end of the protocol, the verifier outputs either valid (meaning that it considers the signature as valid) or $\perp$ (meaning that it considers the validity of the signature as undetermined).
- $\Pi_{\text{dis}} = (\mathcal{P}_{\text{dis}}, \mathcal{V}_{\text{dis}})$: The disavowal protocol which is run between the signer (with private input sk) and a verifier, on common input $(\texttt{pk}, \mu, \sigma)$. At the end of the protocol, the verifier outputs either invalid (meaning that it considers the signature as invalid) or $\perp$ (meaning that it considers the validity of the signature as undetermined).
- $\texttt{IConvert}(\texttt{pk}, \texttt{sk}, \mu, \sigma)$: A potentially randomized algorithm which on input pk, sk, a message μ and a signature σ, either outputs $\perp$ if the signature is invalid, or an individual receipt ρ_i (enabling to universally verify the signature) if σ is valid.
- $\texttt{IVer}(\texttt{pk}, \mu, \sigma, \rho_i)$: A deterministic algorithm which on input pk, a message/signature pair (μ, σ), and a individual receipt ρ_i, either accepts (outputs 1) or rejects (outputs 0).
- $\texttt{UConvert}(\texttt{pk}, \texttt{sk})$: A potentially randomized algorithm which on input a public/secret key pair $(\texttt{pk}, \texttt{sk})$, outputs a universal receipt ρ_u enabling to universally verify signatures created under $(\texttt{pk}, \texttt{sk})$.
- $\texttt{UVer}(\texttt{pk}, \rho_u, \mu, \sigma)$: A deterministic algorithm which on input pk, a universal receipt ρ_u and a message/signature pair (μ, σ), either accepts (outputs 1) or rejects (outputs 0).

Given a public/secret key pair $(\texttt{pk}, \texttt{sk})$, we define the oracle $\texttt{Check}_{(\texttt{pk},\texttt{sk})}$ as follows: it takes as input a message μ and a signature σ. If σ is a valid signature for μ under pk, then it implements $\mathcal{P}_{\text{con}}$ with private input sk and common input $(\texttt{pk}, \mu, \sigma)$, and otherwise it implements $\mathcal{P}_{\text{dis}}$ (with the same inputs).

The scheme should satisfy the following correctness properties. For all $(\texttt{pk}, \texttt{sk})$ possibly output by KeyGen, all messages $\mu \in \{0,1\}^*$, all valid signatures σ possibly output by $\texttt{USign}(\texttt{pk}, \texttt{sk}, \mu)$, and all invalid signatures σ' for μ, the following holds with probability 1:

- completeness of Π_{con}: $\texttt{valid} \leftarrow \langle \mathcal{P}_{\text{con}}(\texttt{sk}), \mathcal{V}_{\text{con}} \rangle (\texttt{pk}, \mu, \sigma)$
- completeness of Π_{dis}: $\texttt{invalid} \leftarrow \langle \mathcal{P}_{\text{dis}}(\texttt{sk}), \mathcal{V}_{\text{dis}} \rangle (\texttt{pk}, \mu, \sigma')$
- $1 \leftarrow \texttt{IVer}(\texttt{pk}, \mu, \sigma, \texttt{IConvert}(\texttt{pk}, \texttt{sk}, \mu, \sigma))$
- $1 \leftarrow \texttt{UVer}(\texttt{pk}, \texttt{UConvert}(\texttt{pk}, \texttt{sk}), \mu, \sigma)$

C.2 Security Definitions

The security goals for a CUS scheme are as follows. We assume that a public key unambiguously defines a finite signature space $\texttt{SigSp}(\texttt{pk})$.

– *soundness* of Π_{con} and Π_{dis}: informally, a cheating signer shall not be able to prove an invalid signature valid with Π_{con}, or an valid signature invalid with Π_{dis}.
– *unforgeability*: Security against existential forgery under chosen-message attacks (EF-CMA-security) is defined as follows. Let $\mathcal{CUS}$ be a CUS scheme and $\mathcal{A}$ be an adversary. We define the EF-CMA advantage of $\mathcal{A}$ as:

$$\mathbf{Adv}^{\text{ef-cma}}_{\mathcal{CUS},\mathcal{A}}(k) = \Pr\left[\mathbf{Exp}^{\text{ef-cma}}_{\mathcal{CUS},\mathcal{A}}(k) = 1\right] \ ,$$

where the experiment is defined as:

Experiment $\mathbf{Exp}^{\text{ef-cma}}_{\mathcal{CUS},\mathcal{A}}(k)$:
$(\mathtt{sk}, \mathtt{pk}) \leftarrow \mathtt{KeyGen}(1^k)$
$\rho_u \leftarrow \mathtt{UConvert}(\mathtt{pk}, \mathtt{sk})$
$(\mu^*, \sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}}(\mathtt{pk}, \rho_u)$
if σ^* is valid for μ^* under $(\mathtt{pk}, \mathtt{sk})$ output 1 else output 0

and the oracle $\mathcal{O}$ is defined as $\mathtt{USign}(\mathtt{pk}, \mathtt{sk}, \cdot)$. The adversary is not allowed to return (μ^*, σ^*) such that σ^* was obtained by querying μ^* to $\mathcal{O}$.
A CUS scheme is said to be EF-CMA-secure if the advantage $\mathbf{Adv}^{\text{ef-cma}}_{\mathcal{CUS},\mathcal{A}}(k)$ is negligible for all PPT adversaries $\mathcal{A}$.
– *invisibility*: Invisibility under chosen message attacks (INV-CMA-security) is defined as follows. Let $\mathcal{CUS}$ be a CUS scheme and $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be a two-stage adversary. We define the INV-CMA advantage of $\mathcal{A}$ as:

$$\mathbf{Adv}^{\text{inv-cma}}_{\mathcal{CUS},\mathcal{A}}(k) = \left|\Pr\left[\mathbf{Exp}^{\text{inv-cma}}_{\mathcal{CUS},\mathcal{A}}(k) = 1\right] - \frac{1}{2}\right| \ ,$$

where the experiment is defined as:

Experiment $\mathbf{Exp}^{\text{inv-cma}}_{\mathcal{CUS},\mathcal{A}}(k)$:
$(\mathtt{sk}, \mathtt{pk}) \leftarrow \mathtt{KeyGen}(1^k)$
$(\mu^*, \mathtt{state}) \leftarrow \mathcal{A}_1^{\mathcal{O}}(\mathtt{pk})$
$b \leftarrow_{\$} \{0,1\}$
if $b = 0$ set $\sigma^* = \mathtt{USign}(\mathtt{pk}, \mathtt{sk}, \mu^*)$ else set $\sigma^* \leftarrow_{\$} \mathtt{SigSP}(\mathtt{pk})$
$b' \leftarrow \mathcal{A}_2^{\mathcal{O}}(\mathtt{pk}, \mathtt{state}, \sigma^*)$
if $b = b'$ output 1 else output 0

and $\mathcal{O}$ is the set of oracles $\{\mathtt{USign}(\mathtt{pk}, \mathtt{sk}, \cdot), \mathtt{Check}_{(\mathtt{pk},\mathtt{sk})}(\cdot,\cdot), \mathtt{IConvert}(\mathtt{pk}, \mathtt{sk}, \cdot, \cdot)\}$. The adversary is not allowed to query these oracles with input μ^*.
A CUS scheme is said to be INV-CMA-secure if the advantage $\mathbf{Adv}^{\text{inv-cma}}_{\mathcal{CUS},\mathcal{A}}(k)$ is negligible for all PPT adversaries $\mathcal{A}$.

C.3 Construction from a Static TDDH Group

Let $\mathcal{STDDH} = (\mathtt{Gen}, \mathtt{Samp}, \mathtt{Solve})$ be a static TDDH group with perfect soundness. For this part, we assume that $\mathtt{Gen}$ outputs a tuple $([\mathbb{G}], G, \tau)$ such that $\mathbb{G}$ is cyclic and efficiently recognizable, and G is a generator of $\mathbb{G}$. We construct a CUS scheme $\mathcal{CUS}$ as follows.

– $\mathtt{KeyGen}(1^k)$: run the static TDDH group generator $\mathtt{Gen}(1^k)$ to obtain $([\mathbb{G}], G, \tau)$, and $\mathtt{Samp}([\mathbb{G}], G, \tau)$ to obtain (X, x, τ_x). Select a hash function $\boldsymbol{H} : \{0,1\}^* \to \mathbb{G}$. The public key of the signer is $\mathtt{pk} = ([\mathbb{G}], G, X, \boldsymbol{H})$ and its secret key is $\mathtt{sk} = (x, \tau_x)$.
– $\mathtt{USign}(\mathtt{pk}, \mathtt{sk}, \mu)$: To sign a message $\mu \in \{0,1\}^*$, the signer computes $M = \boldsymbol{H}(\mu)$, and $S = M^x$. The signature is S.
– Π_{con}: run a zero-knowledge variant of the protocol for proving EDL of Appendix A to prove that $(X, \boldsymbol{H}(\mu), S) \in \mathcal{DH}_G$

- Π_{dis}: run a zero-knowledge variant of the protocol for proving IDL of Appendix A to prove that $(X, \boldsymbol{H}(\mu), S) \notin \mathcal{DH}_G$
- $\texttt{IConvert}(\texttt{pk}, \texttt{sk}, \mu, \sigma)$: Given a public key $\texttt{pk} = ([\mathbb{G}], G, X, \boldsymbol{H})$, a secret key $\texttt{sk} = (x, \tau_x)$, a message μ and a signature $\sigma = S$, check whether $\boldsymbol{H}(\mu)^x = S$, and output $\perp$ if this does not hold. Otherwise compute and output a NIZK proof ρ_i that $(X, \boldsymbol{H}(\mu), S) \in \mathcal{DH}_G$ (using a hash function $\boldsymbol{H}_{\text{FS}}$ for the Fiat-Shamir transform)
- $\texttt{IVer}(\texttt{pk}, \mu, \sigma, \rho_i)$: check that ρ_i is a valid NIZK proof that $(X, \boldsymbol{H}(\mu), S) \in \mathcal{DH}_G$
- $\texttt{UConvert}(\texttt{pk}, \texttt{sk})$: to universally convert undeniable signatures, the signer outputs the trapdoor τ_x as the universal receipt ρ_u.
- $\texttt{UVer}(\texttt{pk}, \rho_u, \mu, \sigma)$: To verify a signature $\sigma = S$ on a message $\mu \in \{0,1\}^*$ with the public key $\texttt{pk} = ([\mathbb{G}], G, X, \boldsymbol{H})$ and the universal receipt $\rho_u = \tau_x$, compute $M = \boldsymbol{H}(\mu)$ and run algorithm $\texttt{Solve}([\mathbb{G}], G; X, M, S; \tau_x)$, and accept the signature as valid iff $\texttt{Solve}$ accepts.