Trapdoor DDH groups from pairings and isogenies

Péter Kutas[1] , Christophe Petit[1,2], Javier Silva[3⋆]

[1] University of Birmingham, Birmingham, UK.
[2] Université libre de Bruxelles, Belgium.
[3] Universitat Pompeu Fabra, Barcelona, Spain.
p.kutas@bham.ac.uk, christophe.f.petit@gmail.com,
javier.silva@upf.edu

Abstract. Trapdoor DDH groups are an appealing cryptographic primitive introduced by Dent–Galbraith (ANTS 2006), where DDH instances are hard to solve unless provided with additional information (i.e., a trapdoor). In this paper, we introduce a new trapdoor DDH group construction using pairings and isogenies of supersingular elliptic curves, and present two instantiations of it. The construction solves all shortcomings of previous constructions as identified by Seurin (RSA 2013). We also present partial attacks on a previous construction due to Dent–Galbraith, and we provide a formal security definition of the related notion of "trapdoor pairings".

Keywords: elliptic curve cryptography, pairings, isogenies, trapdoor DDH.

1 Introduction

The hardness of computing discrete logarithms and related problems (including the computational and decisional Diffie–Hellman problems in various groups) has supported the security of numerous cryptographic protocols for more than 40 years. While the decisional Diffie–Hellman (DDH) problem can be solved by solving a discrete logarithm problem, the converse is not known to be true. There are instances of groups equipped with bilinear pairings, where the discrete logarithm problem is believed to be hard but the decisional Diffie–Hellman problem can be solved efficiently.

Trapdoor DDH groups are a cryptographic primitive introduced by Dent–Galbraith in 2006 [17]. Formally, a trapdoor DDH group involves *two* descriptions of a single group. With either description of the group, the usual group operations, including inversion, can be computed efficiently, and solving the discrete logarithm problem and computational Diffie–Hellman problem must be hard. Crucially, the *decisional* Diffie–Hellman problem must also be hard to

⋆ This author was supported by a PhD grant from the Spanish government, co-financed by the ESF (Ayudas para contratos predoctorales para la formación de doctores 2016). This work was partially done while visiting the University of Birmingham.

solve when provided only with the first description of the group, and easy with the second description. The second description can then be used as a trapdoor in a cryptographic protocol, conferring to its owner the power to solve DDH instances.

To the best of our knowledge, there are only two constructions of trapdoor DDH groups in the literature. Dent–Galbraith [17] use supersingular elliptic curves with equations $y^2 = x^3 + x$ defined over RSA rings $\mathbb{Z}_N$. Another construction by Dent–Galbraith was broken in [31]. Seurin [37] uses the group of quadratic residues modulo N^2 where again N is an RSA modulus.

Two more constructions based on the RSA and factoring assumptions are provided by Seurin [37], but these are *static* trapdoor DDH group constructions, where the trapdoor can only solve DDH challenges involving a fixed pair of group elements g, g^x.

Trapdoor DDH groups have been used by Dent–Galbraith to build an identification scheme [17], and by Prabhakaran–Xue to build statistically hiding sets [35]. Seurin further constructs convertible undeniable signature schemes with delegatable verification from *static* trapdoor DDH groups [37]. In his paper, Seurin identifies several features that existing constructions (including his) are lacking, and which could be key to enable "powerful applications of trapdoor DDH groups" [37, Section 1.4].

Our contributions. We provide a new construction of trapdoor DDH groups which has all the features identified by Seurin. Our construction uses a *random* supersingular curve with a large prime as the group order, and an isogeny between this curve and a curve with a known distortion map as a trapdoor. Security relies on the hardness of solving the Decisional Diffie–Hellman problem on a random supersingular elliptic curve, and the hardness of solving the Computational Diffie–Hellman problem when the trapdoor is known. Interestingly, hardness of DDH implies both hardness of the discrete logarithm problem on the curve and hardness of computing an isogeny between a random supersingular curve and a "special" one, with a known distortion map.[4] Our construction solves all open problems of Seurin [37]: the group has public and prime order, hashing onto the group is efficient, and the trapdoor DDH solver always outputs the correct result.

We also provide attacks on the parameters suggested by Dent–Galbraith in their remaining construction, when used in specific applications. We explain how to increase the parameters or modify the scheme to thwart the attack. While these counter-measures defeat both our attacks and previous attacks, we argue that choosing secure parameters for this construction remains a delicate task.

As an additional contribution, we formally define a notion of *trapdoor pairings* which was only implicit in the work of Dent–Galbraith. A trapdoor pairing construction immediately leads to a trapdoor DDH construction, and our new

[4] We stress that DDH is easy for a supersingular curve with a known distortion map, but finding a distortion map on a random curve is believed to be a hard problem [33,18]. See also Section 2.1.

trapdoor DDH groups are in fact trapdoor pairings. However by using trapdoor pairings we are able to improve the efficiency of an identification protocol provided in [17] as an application, while relying on a seemingly weaker computational assumption.

Other constructions based on pairings and isogenies. Pairings and isogeny problems have both considerable applications in cryptography, and since they are both built on elliptic curves, combining them to construct further protocols is a natural idea.

The first work in that direction is due to Koshiba and Takashima [27]. They provided a framework and security definitions for cryptographic protocols involving pairings and isogenies, called *isogenous pairing groups*. They also present key-policy attribute-based encryption schemes based on their framework. We remark that our trapdoor DDH construction does not entirely fit in Koshiba and Takashima's framework: in our construction the pairing is "hidden" and hard to evaluate, whereas in their framework the pairing can be publicly evaluated. Besides, the framework implicitly uses an asymmetric pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ with $\mathbb{G}_1 \neq \mathbb{G}_2$, while we use a symmetric pairing. Finally, we remark that their framework seems to be built with the publicly computable Weil pairing in mind (the Weil pairing is degenerate when $\mathbb{G}_1 = \mathbb{G}_2$), and our construction uses a modified Weil pairing instead.

More recently, De Feo, Masson, Petit and Sanso have constructed a Verifiable Delay Function (VDF) that also uses both pairings and isogenies [15]. Similar ideas have been used to build a Delay Encryption scheme [4]. As in our new trapdoor DDH group, the VDF and Delay Encryption use an isogeny from a "special" supersingular elliptic curve to another "random" curve, and a pairing on the image curve. These constructions crucially differ from ours as their isogeny is not secret and it is of extremely large degree (and of course VDFs, Delay Encryption schemes and Trapdoor DDH groups are different primitives). The pairing used is also the Weil pairing, so it cannot be used to solve DDH instances.

Additionally, the patent [28] presents various trapdoor DDH constructions. The first few are based on the ideas of Dent and Galbraith [17], but the last one actually uses a secret isogeny to produce a trapdoor DDH mechanism. The high-level idea is similar to ours, in particular using an isogeny as the trapdoor information used to compute a pairing. However, they do not provide details on a concrete instantiation, or formal security and efficiency analyses.

Outline. The remaining of this paper is organized as follows. In Section 2 we provide preliminary background on elliptic curves, computational assumptions, trapdoor DDH groups, and previous constructions. In Section 3 we describe our new trapdoor DDH group and we introduce the definition of trapdoor pairing, which the new construction satisfies, and briefly discuss applications. In Section 4, we describe two concrete instantiations of our construction. We also analyze security and suggest concrete parameters. We describe our attacks on Dent–Galbraith's first construction in Section 5, and we conclude the paper in Section 6.

2 Preliminaries

We recall some basic facts about elliptic curves over finite fields in Appendix C.1. For a detailed exposition of the mathematics of elliptic curves, see [38].

2.1 Computational assumptions

We first recall the definitions of the discrete logarithm, the decisional and computational Diffie–Hellmann problems. In the following, we consider groups written with multiplicative notation.

Let $\mathbb{G}$ be a group. The discrete logarithm problem (DLP) is the following: Given $g, h \in \mathbb{G}$, find x (if one exists) such that $g^x = h$. In the computational Diffie–Hellmann problem (CDH) one is given a tuple $(g, g^a, g^b) \in \mathbb{G}^3$ and has to compute g^{ab}. In the decisional Diffie–Hellmann problem the input is a quadruple $(g, g^a, g^b, z) \in \mathbb{G}^4$ and one has to decide whether $z = g^{ab}$. We call a tuple of the form (g, g^a, g^b, g^{ab}) a DDH tuple. Formal definitions of the discrete logarithm, CDH and DDH assumptions can be found in Appendix C.2.

We also define computational problems related to isogenies between elliptic curves.

Definition 1 *Let E_1 and E_2 be two uniformly random supersingular elliptic curves defined over a finite field $\mathbb{F}_q$. Let ℓ be a prime number and e be a positive integer. The* isogeny problem *is to compute an isogeny ϕ between E_1 and E_2 of degree ℓ^e.*

Remark 1. As usually ℓ^e is large, one also asks for an efficient representation of the isogeny, i.e., as a composition of low degree isogenies (and not as a pair of rational functions).

Definition 2 *Let E be a uniformly random supersingular elliptic curve defined over a finite field $\mathbb{F}_q$. The* endomorphism ring problem *asks for a set of endomorphisms of E which generate the endomorphism ring as an abelian group.*

We briefly discuss these computational problems in the context of supersingular elliptic curves.

Discrete logarithm problem. Menezes-Okamoto-Vanstone [30] proposed the following method (referred to as MOV reduction) for reducing the discrete logarithm problem on elliptic curves to the discrete logarithm problem on finite fields. Let E be an elliptic curve defined over a finite field $\mathbb{F}_q$ and let P be a point of order n. Let Q be a point from the subgroup generated by P. In order to find x such that $Q = xP$ the idea is to find the smallest integer k (called the embedding degree) for which $E[n] \subseteq E(\mathbb{F}_{q^k})$ and to use the Weil pairing on E to reduce the elliptic curve discrete logarithm instance to a discrete logarithm instance on $\mathbb{F}_{q^k}$. For general elliptic curves, this reduction does not run in polynomial time as k may be too large. However, for supersingular curves it can be proven that $k \leq 6$ and the reduction does run in polynomial time [30]. This means that for supersingular curves there exist subexponential algorithms for solving the discrete logarithm problem.

Isogeny and endomorphism ring computation problems. For curves over $\mathbb{F}_{p^2}$, the only known algorithms for the isogeny problem run in exponential time, even with the use of a quantum computer. The running time of the best known algorithm is $\tilde{O}(d^{\frac{1}{2}})$ (see [19]) where d is the degree of the desired isogeny. For supersingular curves defined over $\mathbb{F}_p$, there is a quantum algorithm that runs in time $L_p(1/2)$ [2] (for ordinary curves there is an $L_p(1/2)$ algorithm in [6] but the best classical algorithm [16] is still exponential. The computational hardness of these problems has been exploited in various protocols [14,21,15,13,5]. The complexity of the problem does not change if one of the two curves is fixed.

The endomorphism computation problem is closely related to the isogeny problem, as shown in [33,18]. Computing distortion maps (non-scalar endomorphisms) seems to be as hard as the endomorphism ring problem as a small number of endomorphisms will in general generate the full endomorphism ring, unless they satisfy some exceptional conditions. More precisely, one can use a variant of Schoof's point counting algorithm as in Kohel [26, Theorem 81] to compute the Gram matrix associated to a set of maps, and deduce an abstract representation of the endomorphism ring as a $\mathbb{Z}$-basis of elements of $B_{p,\infty}$ (the quaternion algebra ramified at p and at infinity).

While the endomorphism ring computation problem is believed to be hard for random curves, the problem is actually easy for some "special" curves such as the curve $y^2 = x^3 + x$, or more generally any curve defined over $\mathbb{F}_p$ with a small degree non-scalar endomorphism. We note that in [22], an algorithm to produce distortion maps of elliptic curves is given. However, it uses the CM method, which is efficient only for small discriminants. The supersingular curves constructed this way are therefore exactly the "special" curves above. Note that by taking a random isogeny walk we have a negligible probability that the final curve is a special one.

Decisional Diffie–Hellmann problem. On special curves, and more generally when we know a distortion map for a curve, we can build a pairing for which $e(P, P) \neq 1$. In such cases we can solve the DDH problem on the curve using the observation that $A = aP,\ B = bP,\ Z = abP$ for some a, b if and only if $e(A, B) = e(Z, P)$.

It is somewhat folklore belief that the Decisional Diffie–Hellman problem is easy for all supersingular curves (see e.g. [40, Theorem 6]), however we stress that this is only known to hold when provided with a distortion map for the curve. Without this distortion map, the Weil pairing is useless to solve the DDH problem on a curve since $e(P, xP) = 1$ for any x, and DDH remains a plausible hard problem. As discussed above, computing a distortion map for a uniformly random curve is also believed to be hard.

Computational Diffie–Hellman Problem. While a pairing and distortion maps together can help to solve the Decisional Diffie–Hellman problem on a curve, the Computational Diffie–Hellman problem remains a plausible hard problem in this context. When DDH is easy, the assumption that CDH is hard has been called the gap Diffie–Hellman assumption in the cryptography literature [3].

2.2 Trapdoor DDH groups

Trapdoor DDH groups were first introduced by Dent–Galbraith [17]. Intuitively, trapdoor DDH groups are a cryptographic construction in which knowledge of the trapdoor gives its owner the ability to solve DDH instances which are otherwise intractable. Formal definitions have appeared in Dent–Galbraith [17], Seurin [37] and Prabhakaran–Xue [35], with different security requirements in all papers. Here we recall the definition provided in [37].

We denote by $\mathsf{DDH}_{\mathbb{G}}$ the set of DDH tuples of a group $\mathbb{G}$. We shorten deterministic polynomial time to DPT and probabilistic polynomial time to PPT. We say that a function $f(\lambda)$ is negligible, and write $f(\lambda) \approx 0$, if for any positive polynomial $p(\lambda)$ there exists an integer $N > 0$ such that for all $\lambda > N$ we have $f(\lambda) < \frac{1}{p(\lambda)}$. We denote the order of $g \in \mathbb{G}$ by $|g|$.

Definition 3 *A trapdoor DDH group is a pair of algorithms* $(\mathsf{Gen}, \mathsf{Solve})$ *with the following properties. The trapdoor DDH group generator algorithm* Gen *is a PPT algorithm which takes as input a security parameter* 1^λ *and outputs a tuple* $(\mathbb{G}, P, \tau)$ *where* $\mathbb{G}$ *is a group,* $P \in \mathbb{G}$ *is a group element of order* $2^{\Theta(\lambda)}$*, and* τ *is a trapdoor, such that:*

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

Solve *is a DPT algorithm which takes as input* $(\mathbb{G}, P, \tau)$ *and a tuple* $(X, Y, Z, T) \in \mathbb{G}^4$*, either accepts (outputs 1) or rejects (outputs 0), and satisfies the following:*

(iii) Completeness: for all $(\mathbb{G}, P, \tau)$ *possibly output by* Gen*,* Solve *always accepts if* $(X, Y, Z, T) \in \mathsf{DDH}_{\mathbb{G}}$*.*
(iv) Soundness: for any PPT adversary A, we have that:

$$\Pr\left[\begin{array}{c|l}(\mathbb{G}, P, \tau) \leftarrow \mathsf{Gen}(1^\lambda); (X, Y, Z) \leftarrow \langle P \rangle^3; & 1 \leftarrow \mathsf{Solve}(\mathbb{G}, P, \tau; X, Y, Z, T) \\ T \leftarrow A(\mathbb{G}, P; X, Y, Z) & \wedge\, (X, Y, Z, T) \notin \mathsf{DDH}_{\mathbb{G}}\end{array}\right] \approx 0.$$

We say that the trapdoor DDH group has perfect soundness when Solve *always rejects on input a non-DH tuple* (X, Y, Z, T)*, i.e. the above probability is zero.*

The definitions of Seurin and Dent–Galbraith are almost identical, except that the hardness of CDH with the trapdoor is not required explicitly in the definition of Dent–Galbraith. Nevertheless, their constructions satisfy this property. Prabhakaran–Xue additionally impose a Strong RSA assumption and a Diffie–Hellman Knowledge of Exponent assumption on the trapdoor DDH group [35]. These extra assumptions seem plausible for the specific construction of Dent–Galbraith [17] and needed for their application, but they also seem to restrict the range of possible constructions. For example, the Strong RSA assumption does not hold in a group of known order. Obtaining a trapdoor DDH group of known order is actually among the open problems left by Seurin, and in particular the Strong RSA assumption does not hold for our new construction in Section 3.

2.3 Previous constructions

We briefly sketch previous constructions of trapdoor DDH groups. The first one is the most relevant one for this paper.

Dent–Galbraith's "hidden pairing" construction [17]. Choose p_1, p_2 two large primes congruent to 3 mod 4, such that there are large primes $r_i \mid p_i + 1$. Let $N := p_1 p_2$ and let E be an elliptic curve defined by the equation $y^2 = x^3 + x$ over the ring $\mathbb{Z}_N$. Note that the curve is supersingular, with a well-known distortion map $\phi : E \to E : (x, y) \to (-x, \iota y)$ where $\iota^2 = -1$. The number of points of E over $\mathbb{Z}_N$ is $(p_1 + 1)(p_2 + 1)$. Let P be a point of order $r_1 r_2$ and $\mathbb{G}$ be the group generated by P. The key observation is that a quadruple (P, aP, bP, Z) in $E(\mathbb{Z}_N)$ is a valid DDH tuple if and only if it reduces to a valid DDH tuple in $E(\mathbb{F}_{p_1})$ and $E(\mathbb{F}_{p_2})$. The DDH trapdoor in this construction is the factorization of N: given p_1 and p_2 one can solve the DDH problem using the modified Weil pairing described in Section 2.1, since a distortion map on E is known. On the other hand, it seems that without the factorization of N the DDH problem on $E(\mathbb{Z}_N)$ is hard. In Section 5, we will show that in certain contexts, factorization is easier, forcing an increase of the parameters.

Dent–Galbraith's second construction [17]. A second construction was proposed in Dent–Galbraith's paper, based on Frey's idea of disguising an elliptic curve with a Weil descent. However, this construction was subsequently broken in [31].

Seurin's construction based on composite residuosity [37]. Choose two safe primes p_1 and p_2, namely $p_1 = 2p'_1 + 1$ and $p_2 = 2p'_2 + 1$ where p'_1, p'_2 are prime. The group $\mathbb{G}$ is the group of quadratic residues modulo N^2, where $N = p_1 p_2$. The trapdoor is the factorization of N. The group $\mathbb{G}$ is cyclic of order $N p'_1 p'_2$. Let g be a generator of $\mathbb{G}$. Given $y \in \mathbb{G}$, the *partial* discrete logarithm problem asks for the discrete logarithm of y modulo N (and not modulo $N p'_1 p'_2$). As shown by Paillier [32], one can solve *partial* discrete logarithms in $\mathbb{G}$ given the factorization of N, hence one can also solve Diffie–Hellman problems. On the other hand, the security of the construction is based on the hardness of the CDH problem in $\mathbb{G}$ given the factorization of N, as well as on the DDH and partial CDH problems in $\mathbb{G}$ [37].

Seurin [37] also introduced the definition of a *static* trapdoor DDH scheme where the trapdoor can only be used to solve the DDH problems involving a specific pair of elements $(g, g^x) \in \mathbb{G}^2$.

Seurin's static trapdoor DDH construction based on the RSA problem [37]. Let p_1, p_2, N be the same as in the previous construction. Let J_N denote the subgroup of $\mathbb{Z}_N$ consisting of those elements whose Jacobi symbol is 1. This is a cyclic group of order $m = (p_1 - 1)(p_2 - 1)/2$. Let g be a generator of J_N. Generate a random $x \in [0; m - 1]$. The trapdoor is $(m, 1/x \mod m)$, or equivalently, x and the factorization of N. Using the trapdoor one can recognize DDH instances of the form (g, g^x, g^y, g^z) where g and g^x are fixed beforehand. Indeed, (g, g^x, g^y, g^z) is

a DDH tuple if ond only if $(g^z)^{1/x} = g^y$. However, without the knowledge of the trapdoor, this is RSA inversion which seems to be a hard problem.

Seurin's static trapdoor DDH construction based on signed quadratic residues Let $N = p_1 p_2$, where p_1 and p_2 are safe primes congruent to 3 modulo 4. Let $J_N^+ = J_N/\{1, -1\}$. The group J_N^+ is cyclic of order $m = (p_1 - 1)(p_2 - 1)/4$ and let g be a generator of J_N^+. Let $x \in [0; m - 1]$. The trapdoor is $t := 2x \pm m$ (note that the computation of m is equivalent to factoring N). Then an instance (g, g^x, g^y, g^z) is a DDH tuple if and only if $(g^y)^t = (g^z)^2$ as squaring in J_N^+ is injective.

2.4 Seurin's open problems

In his "open problems" Section [37, Section 1.4], Seurin highlighs some shortcomings of previous trapdoor DDH constructions:

> *"Two key features of trapdoor DDH 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 [...]. However, none of the two candidates for TDDH groups (the hidden pairing-based proposal of [17], and [Seurin's construction]) fulfills both requirements. We think that providing a plausible candidate possessing both properties is the key to enable powerful applications of TDDH groups.*
> *A related open problem is whether there exists a plausible construction of a trapdoor DDH group with publicly known (ideally prime) order, since they are usually simpler to use in cryptography."*

In Section 5 we will highlight further issues with Dent–Galbraith's construction, namely attacks on the parameters suggested, in the context of some applications. Interestingly, our new trapdoor DDH group construction will both avoid these issues and solve all of Seurin's open problems.

3 New trapdoor DDH groups from pairings and isogenies

In this section, we first describe our new trapdoor DDH construction. We then provide our new security definition of "trapdoor pairing" satisfied by both our construction and Dent–Galbraith's one.

3.1 Our construction

As is widely known, a *non-degenerate symmetric* pairing $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_t$ can be used to solve a DDH instance $(P, aP, bP, T) \in \mathbb{G}^4$ by checking whether

$$e(P, T) = e(aP, bP).$$

Let us now consider an elliptic curve E and the Weil pairing $e : E[m] \times E[m] \to \boldsymbol{\mu}_m$, where $\boldsymbol{\mu}_m \subset \mathbb{F}_{p^k}^*$ is the group of m-th roots of unity. The Weil pairing is degenerate, meaning that $e(P, P) = 1$, and so by itself it is not useful to solve DDH problems. This has been solved by using a distortion map, that is, an endomorphism $\phi : E \to E$ such that $\phi(P) \notin \langle P \rangle$. We then define a new pairing as

$$\hat{e}(P, Q) = e(P, \phi(Q)),$$

which is used instead of the Weil pairing.

The key observation of our new construction is that the ability to compute a non-degenerate symmetric pairing relies on the knowledge of a distortion map. Moreover for a random supersingular elliptic curve obtaining this map is a hard problem, and so it constitutes a suitable trapdoor for a trapdoor pairing group.

More precisely, the Gen algorithm works as follows. Assume that we have a curve E_0 with a known distortion map $\phi : E_0 \to E_0$. We choose an isogeny $\varphi : E_0 \to E$ to perform a walk in the isogeny graph. We assume that we can efficiently perform a walk such that the output curve is essentially uniform. In Section 4 we will discuss the specifics for each instantiations, and ensure that the walks are indeed efficient and random enough.

The public group G is given by the curve E and the trapdoor information τ is some representation of the isogeny φ. The Solve algorithm has access to the trapdoor, and thus can evaluate the pairing $\hat{e} : E \times E \to \boldsymbol{\mu}_q \subset \mathbb{F}_{p^k}$ defined as

$$\hat{e}(P, Q) = e(\hat{\varphi}(P), \phi(\hat{\varphi}(Q))),$$

where e is the Weil pairing on E_0, and use this to solve DDH instances on E.

3.2 Trapdoor pairings

In our new construction, the trapdoor does not only allow to solve DDH instances, but also the ability to evaluate a non-degenerate symmetric pairing. We now formalize this property with a new definition.

We first identify a computational problem that is harder than DDH and better captures the power of being able to compute a pairing. Essentially, given group elements, a pairing allows a multiplication of their discrete logarithms. This translates into solving decisional problems which consist of checking a quadratic equation in the exponent. Note that although the corresponding computational problems remain hard, they are easy if we allow the output to be in the target group of the pairing. In particular, we consider the following computational problem.

Definition 4 *Let $\mathbb{G}$ be a group and $P \in \mathbb{G}$, and let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a pairing. We call the* Target Computational Diffie–Hellman (Target CDH) problem *the problem consisting on, given $\mathbb{G}, P, aP, bP$ for uniformly random a, b, computing*

$$g, g^{ab} \in \mathbb{G}_T,$$

where $g \neq 1$ must be output before[1] *receiving aP, bP.*

Note that a symmetric non-degenerate pairing can be used to solve the Target CDH problem by computing $g = e(P, P)$ and $g^{ab} = e(aP, bP)$. This implies that both Dent–Galbraith's first construction and our new construction are not only trapdoor DDH groups, but also trapdoor pairings.

Breaking the Target CDH problem implies breaking the DDH problem in $\mathbb{G}$, so the Target CDH problem is at least as hard as the DDH problem, but nevertheless it is still easy given an efficiently computable pairing.

We now formalize the idea of trapdoor pairings by mimicking the previous trapdoor DDH definition, but replacing the requirement that DDH should be solvable with the trapdoor with our harder problem.

Definition 5 *A trapdoor Target CDH group is a pair of algorithms* $(\mathsf{Gen}, \mathsf{Solve})$ *with the following properties. The trapdoor pairing group generator algorithm* Gen *is a PPT algorithm which takes as input a security parameter 1^λ and outputs a tuple $(\mathbb{G}, \mathbb{G}_T, P, \tau)$ where $\mathbb{G}$ and $\mathbb{G}_T$ are the descriptions of two group, $P \in \mathbb{G}$ is a group element of order $2^{\Theta(\lambda)}$, and τ is a trapdoor information, such that:*

(i) Hardness of DDH without the trapdoor: the DDH problem is hard for the group generator Gen' which outputs only $(\mathbb{G}, \mathbb{G}_T, P)$, both in $\mathbb{G}$ and $\mathbb{G}_T$.
(ii) Hardness of CDH with the trapdoor: the CDH problem is hard for Gen, both in $\mathbb{G}$ and $\mathbb{G}_T$.

Solve *is a DPT algorithm which takes as input $(\mathbb{G}, \mathbb{G}_T, P, \tau)$ and a tuple $(X, Y) \in \mathbb{G}^2$, and outputs $(g, u) \in \mathbb{G}_T^2$, and satisfies the following:*

(iii) Completeness: for all $(\mathbb{G}, \mathbb{G}_T, P, \tau)$ possibly output by Gen, and if $X = aP, Y = bP$, Solve *always outputs $(g, u) \in \mathbb{G}_T^2$ such that $u = g^{ab}$.*
(iv) Soundness: for any PPT adversary A, the we have that

$$\Pr\left[\begin{array}{c}(\mathbb{G},\mathbb{G}_T, P, \tau) \leftarrow \mathsf{Gen}(1^\lambda); (aP, bP) \leftarrow \langle P\rangle^2; \\ (g, u) \leftarrow \mathsf{Solve}(\mathbb{G}, \mathbb{G}_T, P, \tau; aP, bP)\end{array} : u \neq g^{ab}\right] \approx 0.$$

We say that trapdoor Target CDH group has perfect soundness when the above probability is zero.

An alternative, perhaps more natural definition could require the Target CDH problem to be hard without the trapdoor, as opposed to the DDH problem in Definition 4. We chose to require hardness of DDH (implying hardness of Target CDH) so that trapdoor pairings are naturally trapdoor DDH groups as well. The only difference between them lies in the power provided by the trapdoor: a DDH solver in Definition 3, and a stronger Target CDH solver in Definition 4.

[1] The reason for asking for g is that, since the pairing will not be available to all parties, it is not immediate to produce a canonical generator of $\mathbb{G}_T$ from the generator of $\mathbb{G}$. We ask for it in advance so that it does not depend on aP, bP.

3.3 Security of our new construction

We now prove that our new construction is a trapdoor pairing in the above sense (hence it is also a trapdoor DDH group).

Theorem 1 *Suppose that the distribution of the curve E output by algorithm* Gen *is statistically equivalent to the uniform distribution. Then, if the DDH problem in E is hard, and the CDH problem in E is hard given the trapdoor, the construction above is a secure trapdoor pairing group.*

Proof: It is clear from the discussion above that the Target CDH problem can be solved efficiently when the trapdoor is known, and by assumption the CDH problem is hard.

Without the trapdoor, solving DDH in $\mathbb{G}$ is exactly the DDH problem on the curve E. While E is not a uniformly random curve, it is the output of a random walk, which is close to uniformly random so that the two problems are equivalent. □

We now argue that the DDH and CDH assumptions of Theorem 1 are plausible. First, the DDH has been widely studied and used in the literature, and is believed to hold when a symmetric pairing is not available, and as discussed in Section 2.1, the DDH problem is easy for supersingular curves only when a distortion map is known.

We remark that constructing a curve with a distortion map is easy: one can choose a special curve, or do a random walk from one of these special curves as in our trapdoor pairing construction. On the other hand given a randomly chosen supersingular curve, computing a distortion map appears to be a difficult problem, as discussed in Section 2.1. Conversely, given the endomorphism ring of a curve E, one can also compute an isogeny between E_0 and E (see [33,18]), and any such isogeny can be used as a trapdoor in our scheme.

While DDH is easy on E with the trapdoor, the CDH problem still appears to be hard on E. Indeed this is formalized by the so-called Gap-CDH assumption in pairing-based cryptography. Moreover, given the trapdoor the CDH problems on the curves E_0 and E are equivalent, as we can use a trapdoor $\varphi : E_0 \to E$ to send a CDH instance (P, aP, bP) in E_0 to $(\varphi(P), \varphi(aP), \varphi(bP))$. Note that scalar multiplication commutes with any isogeny, so this is a CDH instance on E.

The assumption that the output of the group generation algorithm is close to uniformly random will be discussed for the particular instantiations of the algorithm, in Section 4, as the argument is different in each case.

3.4 Applications

We considered two applications of our construction, and more generally of trapdoor pairings. We briefly discuss these applications here and refer to appendix for details.

First, we build an identification scheme that improves on a previous construction by Dent–Galbraith [17] that was based on trapdoor DDH groups. The protocol by Dent–Galbraith has to be iterated several times in order to achieve soundness, while our protocol allows for arbitrarily long challenges. Our improved construction highlights how using a trapdoor pairing in place of a general trapdoor DDH group leads to a more efficient scheme.

We also discuss ElGamal voting and similar protocols, and warn against the use of random supersingular curves in such protocols. Indeed, we show how a construction similar to ours could then be used as a backdoor to break anonymity in these contexts.

4 Two concrete instantiations

In [37, Section 2.4], Seurin requests the following useful features for a trapdoor DDH group:

- The group order can be publicly revealed.
- The group order is a prime number.
- There is an efficient hashing algorithm into the group.

We note that no previous construction has achieved these properties at the same time. In particular, all of them use composite-order groups.

We consider two instantiations of our idea, one using curves over $\mathbb{F}_{p^2}$, and another using curves over $\mathbb{F}_p$. The first one satisfies the first property, and either the second or the third, but not both simultaneously. The second instantiation achieves the three properties at the same time.

4.1 Curves over $\mathbb{F}_{p^2}$

We start by stating a simple result that ensures that our isogenies will be defined over $\mathbb{F}_{p^2}$, and we will not need to move to extension fields.

Lemma 1. *Let $p \geq 3$ be a prime, and let E be a supersingular elliptic curve such that $\#E(\mathbb{F}_{p^2}) = (p+1)^2$. Then the 2-isogenies from E are $\mathbb{F}_{p^2}$-rational.*

Proof. Since E has $(p+1)^2$ points over $\mathbb{F}_{p^2}$, we have that $E(\mathbb{F}_{p^2})$ is isomorphic to $\mathbb{Z}_{p+1} \times \mathbb{Z}_{p+1}$ [11, Theorem 54]. Since p is an odd prime, we have that $2 \mid (p+1)$, and so $\mathbb{Z}_{p+1} \times \mathbb{Z}_{p+1}$ contains a copy of $\mathbb{Z}_2 \times \mathbb{Z}_2$, which is necessarily the 2-torsion of the curve. Thus the 2-torsion is $\mathbb{F}_{p^2}$-rational, and so are the 2-isogenies. □

We also recall a result that ensures that the output of an isogeny random walk has an almost-uniform distribution.

Theorem 2 ([21], Theorem 1) *Let p be a prime number, let N_p be the number of vertices in the supersingular isogeny graph, and let j_0 be a supersingular invariant in characteristic p. Let j be the final j-invariant reached by a random walk of degree $n = \ell^e$ from j_0. Then for every j-invariant $\tilde{j}$ we have*

$$\left| \Pr[j = \tilde{j}] - \frac{1}{N_p} \right| < \left(\frac{2\sqrt{\ell}}{\ell + 1} \right)^e .$$

By taking $e > 2(1+\epsilon)\log_\ell p$, it is easy to see that the right-hand side in the equation above is smaller than $2/p^{1+\epsilon}$, for any $\epsilon > 0$, so the output distribution of the random walk is close to uniform (the statistical distance is negligible).

Let p be a prime such that $p = 3 \bmod 4$ and $p+1 = qf$, where q is also prime and f is a small cofactor. We consider the curve $E_0 : y^2 = x^3 + x$ over $\mathbb{F}_{p^2}$. This curve is in the conditions of Lemma 1 above, has j-invariant $j = 1728$, and its endomorphism ring is known (see e.g. [33]).

To generate the trapdoor DDH group, we take a random walk $\varphi : E_0 \to E$ composed of 2-isogenies, long enough to ensure that the output curve E is statistically uniform in the graph. Since isogenies preserve the supersingular property and the number of points, any curve that we reach from E_0 through 2-isogenies is also in the conditions of the lemma, and therefore every step of the walk is defined over $\mathbb{F}_{p^2}$.

At this point, we have two options:

- We consider $E(\mathbb{F}_{p^2})$ as the trapdoor group. The group order is $(p+1)^2$, and is public, and we can efficiently hash into the group using standard techniques [23], but the group is not of prime order. In fact, the group is not even cyclic.
- We consider a subgroup of $E(\mathbb{F}_{p^2})$ of order q as the trapdoor group. It is easy to find a point of order $p+1$ in $E(\mathbb{F}_{p^2})$ and multiply it by f to obtain a point of order q, which is close in size to p. In this case, the group order is public and prime, but there is no obvious way to securely hash into the group.

Algorithm 1 Trapdoor group generation (curves over $\mathbb{F}_{p^2}$)

Require: security parameter λ.
Ensure: group description $(\mathbb{G}, P)$, trapdoor φ.
1: Choose primes $p, q = \Theta(2^\lambda)$ such that $p = 3 \bmod 4$ and $p+1 = qf$ for small f.
2: Define the curve $E_0 : y^2 = x^3 + x$ over $\mathbb{F}_{p^2}$.
3: Take a random walk $\varphi : E_0 \to E$ of length $2\log p$ composed of 2-isogenies.
4: Choose a point $Q \in E(\mathbb{F}_{p^2})$ of order $p+1$. Set $P = fQ$.
5: Output $(\langle P \rangle, P, \varphi)$.

The hardness of computing a distortion map then relies on the hardness of computing an isogeny between a fixed curve and a random curve over $\mathbb{F}_{p^2}$, for which only exponential-time attacks are known, as discussed in Section 2.1. This justifies the assumptions of Theorem 1.

4.2 Curves over $\mathbb{F}_p$

We now present an alternative instantiation that uses curves over $\mathbb{F}_p$. In the previous section, we have seen an instantiation that uses a prime-order group

which we cannot efficiently hash into. The reason is that $E(\mathbb{F}_{p^2})$ does not have a unique subgroup of order q, so the trapdoor group must be specified though a generator. With this description, there is no obvious way to hash into the group without knowing the discrete logarithm of the hash, which is undesirable for security. To solve this, we want to find a group in which we can canonically identify a subgroup of order q.

To do so, we work over $\mathbb{F}_p$ instead of $\mathbb{F}_{p^2}$, taking an approach similar to CSIDH [5]. We choose a prime $p = 3 \bmod 4$ such that $p + 1 = 4\ell_1 \dots \ell_n q$, for a large prime q, and we consider again the curve $E_0 : y^2 = x^3 + x$, now over $\mathbb{F}_p$.

The idea is again to take a random walk in the isogeny graph, using only $\mathbb{F}_p$-rational isogenies.

Lemma 2. *$E(\mathbb{F}_p)$ has a unique subgroup of order q.*

Proof. $E(\mathbb{F}_p)$ has $p+1$ points. Since $q \mid (p+1)$ but $q^2 \nmid (p+1)$, we have that there is a subgroup of order q in $E(\mathbb{F}_p)$, but the $\mathbb{F}_p$-rational curve does not contain the whole q-torsion. Then $E(\mathbb{F}_p)$ contains only one subgroup $\mathbb{G}$ of order q. □

We will use this unique subgroup $\mathbb{G}$ as the trapdoor DDH group. Note that the embedding degree of the pairing is the smallest integer k such that $\#\mathbb{G} \mid (p^k - 1)$, so in this case we have $k = 2$. Hashing into this group is now easy. We make it explicit in the following result.

Lemma 3. *There is an efficient algorithm to hash into $\mathbb{G}$.*

Proof. Given a string, we hash it into $\mathbb{F}_p$. We interpret the result as the x-coordinate of a point in $E(\mathbb{F}_p)$. Note that a uniformly random element of $\mathbb{F}_p$ will correspond to a point in the curve with probability roughly $1/2$. If that is not the case, we increase the x-coordinate until it corresponds to a point P. Then we compute $\frac{p+1}{q}P$, landing into the unique subgroup of order q. Note that $\frac{p+1}{q}$ is coprime to q, so given a uniformly random $P \in E(\mathbb{F}_p)$, we have that $\frac{p+1}{q}P$ is a uniformly random point in $\mathbb{G}$.

□

It only remains to specify how to move through the graph using only $\mathbb{F}_p$-rational isogenies. Note that the same argument used in Lemma 2 for q works for any ℓ_i, so $E(\mathbb{F}_p)$ contains only one subgroup of order ℓ_i for each $i = 1, \dots, n$. We make use of the following result (see [12, Section 15] for a proof).

Lemma 4. *Let $\ell \geq 3, p \geq 5$ be different prime numbers, such that $\left(\frac{D}{\ell}\right) = 1$, where $D = t^2 - 4p$, and t is the trace of Frobenius. Let E be a supersingular elliptic curve. Then $\#E(\mathbb{F}_p) = p + 1$ and there are two ℓ-isogenies from E that are $\mathbb{F}_p$-rational. Moreover, these correspond to:*

- *The unique subgroup H of order ℓ of $E(\mathbb{F}_p)$.*
- *The unique subgroup $\tilde{H}$ of order ℓ of $\tilde{E}(\mathbb{F}_p)$, where $\tilde{E}$ is the quadratic twist of E.*

Then, the random walk consists of choosing exponents $e_1, \ldots, e_n \in [-B, B]$. An exponent corresponds to $|e_i|$ steps in which we use the isogeny with kernel H or $\tilde{H}$, respectively, depending on whether the sign of e_i is positive or negative. The distribution of the output of the random walk depends on the structure of the class group of $\mathrm{End}_p(E_0)$. Although for certain parameters it has been computed [1], in general we make the heuristic assumption that, for B and n large enough, the random walk reaches any point in the graph with roughly the same probability.[5]

Algorithm 2 Trapdoor group generation (curves over $\mathbb{F}_p$)

Require: security parameter λ.
Ensure: group description $(\mathbb{G}, P)$, trapdoor φ.
1: Choose primes $p, q = \Theta(2^\lambda)$ such that $p = 3 \bmod 4$, and small odd primes $\ell_1, \ldots, \ell_n$ and integers $e_1, \ldots, e_n$ such that $p + 1 = 4\ell_1 \ldots \ell_n q$, and $\prod_n \ell_i > 2\sqrt{p}$.
2: Define the curve $E_0 : y^2 = x^3 + x$ over $\mathbb{F}_p$.
3: Take a random walk $\varphi : E_0 \to E$ composed of e_i isogenies of degree ℓ_i, for each $i = 1, \ldots, n$.
4: Let $\mathbb{G}$ be the unique subgroup of $E(\mathbb{F}_p)$ of order q, and let P be a generator.
5: Output $(\mathbb{G}, P, \varphi)$.

In a similar way to the case above, the distortion map is protected by the hardness of computing an isogeny between curves over $\mathbb{F}_p$. We note that there is a quantum subexponential algorithm for this problem, due to Biasse-Jao-Sankar [2]. However, our construction depends on the discrete logarithm assumption, which is broken in the quantum setting anyway, so we focus on classical security.

4.3 Parameter choices

Let λ be the security parameter. There are two main ways to break the security of the constructions: recovering the trapdoor or solving the discrete logarithm problem. The first approach amounts to finding a non-scalar endomorphism on E or an isogeny to E_0. Recall that for supersingular elliptic curves, the best known classical algorithm [19] has complexity $\tilde{O}(p^{\frac{1}{2}})$.

As for the discrete logarithm problem, one can apply the MOV reduction [30] to reduce any discrete logarithm problem over either E_0 or E to a discrete logarithm problem over $\mathbb{F}_{p^k}$, where $k = 3$ in the first construction and $k = 2$ in the second. Note that the reduction from E is only available if the trapdoor is known, but nevertheless we do not want a party that knows the trapdoor to be able to solve CDH. The best algorithm for computing discrete logarithms in finite fields of large characteristic is the number field sieve and its variants [24,25],

[5] In CSIDH [5], the authors suggest $B = 5$ and $n = 74$ for a prime p of length 512 bits.

which have complexity $L_n(1/3)$, where in this case $n = p^k$. On the other hand, the best algorithms for solving the discrete logarithm directly in the curve are the generic ones, with complexity $\tilde{O}(q^{1/2})$. One should therefore choose $\log p = \Omega(\lambda^3)$ to avoid these attacks.

- For the construction over $\mathbb{F}_{p^2}$, recall that $p + 1 = qf$ for a small cofactor f, so roughly $\log p = \log q$. Thus, the trapdoor group $\mathbb{G}$ is formed by $\Theta(2^{\lambda^3})$ elements over $\mathbb{F}_{p^2}$.
- For the construction over $\mathbb{F}_p$, we have that $p+1 = 4\ell_1 \dots \ell_n q$, and we require that $\prod_i \ell_i > 2\sqrt{p}$, so roughly $\log p \approx 2 \log q$. Therefore, our trapdoor group is again formed by $\Theta(2^{\lambda^3})$ elements over $\mathbb{F}_p$.

The trapdoor is easy to store, as a d-isogeny requires $\log d = O(\log p)$ bits.

4.4 Comparison with previous constructions.

Dent–Galbraith's construction. Since the trapdoor is the factorization of N, which in turn can be obtained from the factorization of $r_1 r_2$, as explained in Section 5, we need to ensure that this is hard. We must therefore choose $\log(r_1 r_2) = \Omega(\lambda^3)$ to prevent the number field sieve, and since we require $r_i < \sqrt{p_i}$, we need at least $N = p_1 p_2 = \Omega((r_1 r_2)^2)$. We refer to Section 5 for a discussion of the case $r_1 = r_2$ and potential further attack developments.

Seurin's construction. This construction also relies on the factorization of $N = p_1 p_2$, so we must ensure that $\log N = \Omega(\lambda^3)$. Then the trapdoor DDH group is of order $N p_1' p_2' \approx N^2$.

We note that our new construction is asymptotically comparable to the previous proposals in terms of efficiency, while satisfying a stronger definition than Seurin's construction and some desirable properties missing in previous constructions. Also, choosing parameters is more straightforward than in Dent–Galbraith's construction, as the new construction is in a prime-order group, hence we do not need to account for potential factorization attacks, as those described described in the next section.

5 Partial attacks on Dent–Galbraith's construction

Dent–Galbraith's hidden pairing construction uses pairings on elliptic curves defined over RSA rings. As already pointed out in [20], selecting parameters for such constructions may be tricky. We now demonstrate this by showing attacks on the construction when the group order is revealed. Note that Dent–Galbraith suggest to reveal this information in some applications, for example to allow delegation of the pairing computation.

5.1 Case $r_1 r_2$ known and small, $r_1 \neq r_2$

We first give a simple attack on the parameters suggested by Dent–Galbraith ($p_i \approx 2^{512}$ and $r_i \approx 2^{160}$) for their construction.

Let p_1, p_2, r_1, r_2 as in Dent–Galbraith's construction, and assume that $r_1 \neq r_2$ (this condition is not explicitly required in their paper, but it is implied by their later statement that P has order $r_1 r_2$). With $r_i \approx 2^{160}$ the product $r_1 r_2$ can be easily factored with current techniques, so we can assume knowledge of r_1 and r_2. We can then apply a technique from [20, Section 4] to factor N. Namely, we apply x-only addition and doubling formulae to compute the x-coordinate of $[r_1]P$. This leads to the point at infinity modulo p_1 but not modulo p_2, hence a factor of N can be recovered as in the elliptic curve factorization method [29].

To defeat this attack one can choose parameters such that r_1 and r_2 cannot be computed from their product $r_1 r_2$, and make sure other attacks are not feasible. One condition stated in [17] is that $r_i < \sqrt{p_i}$, so the attack requires to at least double the size of p_1 and p_2.

An a priori plausible alternative way to defeat the attack is to enforce $r_1 = r_2$. In this case $E(\mathbb{Z}_N)$ is the direct product of two cyclic groups of order $p_i + 1$ and similarly $\mathbb{G}$ is the direct product of two cyclic groups of order r. With this configuration, multiplying any point in $\mathbb{G}$ by r gives ∞ modulo both p_1 and p_2, hence no factor is recovered. We now consider this case more thoroughly.

5.2 Case $r_1 = r_2$ a known prime

The setting for a known $r := r_1 = r_2$ was in fact already studied in [20], and the best attack presented there has a complexity $O(N^{1/4}/r)$. Taking p_1 and p_2 with 512 bits and r with 160 bits leads to a cost of 2^{96} for this attack, which seems impractical today.

However, we now present an alternative attack in this setting, using Coppersmith's techniques for finding small integer roots of bivariate polynomials [7] and its generalizations by Coron [8,9,10]. [6] In order to factor N, we only need to find x and y such that $N = (rx - 1)(ry - 1)$, i.e., we are looking for roots of the bivariate polynomial

$$p(x, y) = 1 - N - rx - ry + r^2 xy.$$

For the parameters above there is a root (x_0, y_0) such that $|x_0| \leq 2^{352}$ and $|y_0| \leq 2^{352}$. We will use the following result.

Theorem 3 ([7], Corollary 2) *Let $p(x, y) \in \mathbb{Z}[x, y]$ be a bivariate irreducible polynomial of maximum degree δ in each variable. Let X, Y be upper bounds on the desired integer solution (x_0, y_0) and let $W = max_{i,j} \{|p_{ij}| X^i Y^j\}$. If $XY < W^{2/(3\delta)}$, then in time polynomial in $(\log W, 2^\delta)$ one can find an integer solution (x_0, y_0) to the equation $p(x, y) = 0$ such that $|x_0| \leq X$, $|y_0| \leq Y$.*

[6] This attack can be readily extended when $r_1 \neq r_2$, but in that case the attack from the previous section will be simpler.

An easy calculation shows that we cannot apply Theorem 3 directly here: indeed our polynomial p has degree 1 in each variable, and we have $XY \approx 2^{704}$ and $W^{2/3} \approx N^{2/3} \approx 2^{683}$. However, we can still apply the theorem by guessing a few bits of both x and y and iterating Coron's algorithm. Specifically, we set $x := 2^{12}x' + c_1$ and $y := 2^{12}y' + c_2$ where $0 \leq c_i \leq 2^{12}$ and we try to find a solution for each admissible pair (c_1, c_2). With this approach we now have bounds $X = Y = 2^{340}$ on x' and y', and we still have $W^{2/3} \approx N^{2/3} \approx 2^{683}$. As there are 2^{12} choices for each of the c_i, we only need to run the algorithm from [9] at most 2^{24} times to find p_1 and p_2.

One way to defeat this attack in practice is to increase the number of guesses needed; we now estimate the parameters needed to guarantee that this is bigger than 2^{80}. Assume r is a k bit integer and the p_i are $k + \ell$ bit primes, where k and ℓ are positive integers. Then XY is a 2ℓ bit integer and the number of bits of $W^{2/3}$ is $\frac{4}{3}(k + \ell)$. In order to achieve the desired security we need that $2\ell - 80 > \frac{4}{3}(k + \ell)$ or $\ell > 2k + 120$. When r has $k = 160$ bits, we need p_i with at least $\ell + k > 600$ bits, hence N should have at least 1200 bits.

A discussion on potential extensions of these attacks can be found in Appendix B.

6 Conclusion and further work

In this paper, we presented a new trapdoor DDH group construction based on supersingular elliptic curves and pairings. We also gave partial attacks on a previous trapdoor DDH group construction, and we provided a formal security definition for a related but more powerful primitive called "trapdoor pairing" (which our new construction also satisfies). Our new construction has a number of interesting properties; in particular it has all the properties identified by Seurin in his "open problems" Section [37, Section 1.4] as crucial for applications.

Although trapdoor DDH groups were introduced in 2006, the number of applications of it has been so far quite limited. Seurin [37] identified some limitations of all the previous constructions (included their own), and hoped that solving these would allow for more meaningful applications. Our new construction satisfies all the properties required by Seurin, yet no obvious application seems to arise. The notions of trapdoor DDH groups and trapdoor pairings seem to fit quite naturally with the idea of a distinguished party, which would use the trapdoor to perform some special operation that is only allowed to him. This suggests that trapdoor DDH groups might be useful in constructing schemes where there is an authority figure. For example, in group signatures, members of the group can sign messages anonymously on behalf of the group. There is a group manager that is allowed to trace the signer, but is not able to produce forgeries. In this setting, a manager with a trapdoor could maybe identify a signer by noticing a DDH tuple that involves the user's public key, the message and the signature. At the same time, hardness of DDH for the rest of the parties would keep the signatures anonymous for them. We leave the development of such a scheme to further work.

Acknowledgements. We thank Jens Groth, Steven Galbraith and Frederik Vercauteren for discussions related to this work. In particular, some of our applications were suggested by Jens Groth. We also thank the anonymous reviewers.Work by the first and second author was supported by an EPSRC New Investigator grant (EP/S01361X/1). The third author was supported by a PhD grant from the Spanish government, co-financed by the ESF (Ayudas para contratos predoctorales para la formación de doctores 2016). This work was partially done while the third author visited the University of Birmingham.

References

1. W. Beullens, T. Kleinjung, and F. Vercauteren. CSI-FiSh: Efficient isogeny based signatures through class group computations. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 227–247. Springer, 2019. 15
2. J.-F. Biasse, D. Jao, and A. Sankar. A quantum algorithm for computing isogenies between supersingular elliptic curves. In *International Conference on Cryptology in India*, pages 428–442. Springer, 2014. 5, 15
3. I. F. Blake, G. Seroussi, and N. P. Smart. *Advances in elliptic curve cryptography*, volume 317. Cambridge University Press, 2005. 5
4. J. Burdges and L. De Feo. Delay encryption. 2020. 3
5. W. Castryck, T. Lange, C. Martindale, L. Panny, and J. Renes. CSIDH: an efficient post-quantum commutative group action. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 395–427. Springer, 2018. 5, 14, 15
6. A. Childs, D. Jao, and V. Soukharev. Constructing elliptic curve isogenies in quantum subexponential time. *Journal of Mathematical Cryptology*, 8(1):1–29, 2014. 5
7. D. Coppersmith. Small solutions to polynomial equations, and low exponent RSA vulnerabilities. *Journal of Cryptology*, 10(4):233–260, 1997. 17
8. J.-S. Coron. Finding small roots of bivariate integer polynomial equations revisited. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 492–505. Springer, 2004. 17
9. J.-S. Coron. Finding small roots of bivariate integer polynomial equations: A direct approach. In *Annual International Cryptology Conference*, pages 379–394. Springer, 2007. 17, 18
10. J.-S. Coron, A. Kirichenko, and M. Tibouchi. A note on the bivariate Coppersmith theorem. *Journal of Cryptology*, pages 1–5, 2013. 17
11. L. De Feo. Mathematics of isogeny-based cryptography. *arXiv preprint arXiv:1711.04062*, 2017. 12
12. L. De Feo. Isogeny graphs in cryptography. 2019. 14
13. L. De Feo and S. D. Galbraith. SeaSign: Compact isogeny signatures from class group actions. Technical report, IACR Cryptology ePrint Archive, 2018. 5
14. L. De Feo, D. Jao, and J. Plût. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. *Journal of Mathematical Cryptology*, 8(3):209–247, 2014. 5

15. L. De Feo, S. Masson, C. Petit, and A. Sanso. Verifiable delay functions from supersingular isogenies and pairings. Technical report, Cryptology ePrint Archive, Report 2019/166, 2019. 3, 5
16. C. Delfs and S. D. Galbraith. Computing isogenies between supersingular elliptic curves over $\mathbb{F}_p$. *Designs, Codes and Cryptography*, 78(2):425–440, 2016. 5
17. A. W. Dent and S. D. Galbraith. Hidden pairings and trapdoor DDH groups. In *International Algorithmic Number Theory Symposium*, pages 436–451. Springer, 2006. 1, 2, 3, 6, 7, 8, 12, 17
18. K. Eisenträger, S. Hallgren, K. Lauter, T. Morrison, and C. Petit. Supersingular isogeny graphs and endomorphism rings: reductions and solutions. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 329–368. Springer, 2018. 2, 5, 11
19. S. D. Galbraith. Constructing isogenies between elliptic curves over finite fields. *LMS Journal of Computation and Mathematics*, 2:118–138, 1999. 5, 15
20. S. D. Galbraith and J. F. McKee. Pairings on elliptic curves over finite commutative rings. In *IMA International Conference on Cryptography and Coding*, pages 392–409. Springer, 2005. 16, 17
21. S. D. Galbraith, C. Petit, and J. Silva. Identification protocols and signature schemes based on supersingular isogeny problems. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 3–33. Springer, 2017. 5, 12
22. S. D. Galbraith and V. Rotger. Easy decision Diffie-Hellman groups. *LMS Journal of Computation and Mathematics*, 7:201–218, 2004. 5
23. T. Icart. How to hash into elliptic curves. In *Annual International Cryptology Conference*, pages 303–316. Springer, 2009. 13
24. A. Joux, R. Lercier, N. Smart, and F. Vercauteren. The number field sieve in the medium prime case. In *Annual International Cryptology Conference*, pages 326–344. Springer, 2006. 15
25. T. Kim and R. Barbulescu. Extended tower number field sieve: A new complexity for the medium prime case. In *Annual International Cryptology Conference*, pages 543–571. Springer, 2016. 15
26. D. R. Kohel. *Endomorphism rings of elliptic curves over finite fields*. PhD thesis, University of California, Berkeley, 1996. 5
27. T. Koshiba and K. Takashima. Pairing cryptography meets isogeny: A new framework of isogenous pairing groups. *IACR Cryptology ePrint Archive*, 2016:1138, 2016. 3
28. K. E. Lauter, D. Charles, and A. Mityagin. Trapdoor pairings, May 15 2012. US Patent 8,180,047. 3
29. H. W. Lenstra Jr. Factoring integers with elliptic curves. *Annals of Mathematics*, pages 649–673, 1987. 17
30. A. J. Menezes, T. Okamoto, and S. A. Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Transactions on information Theory*, 39(5):1639–1646, 1993. 4, 15
31. D. J. M. Morales. An attack on disguised elliptic curves. *Journal of Mathematical Cryptology*, 2(1):1–8, 2008. 2, 7
32. P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 223–238. Springer, 1999. 7
33. C. Petit and K. E. Lauter. Hard and easy problems for supersingular isogeny graphs. *IACR Cryptology ePrint Archive*, 2017:962, 2017. 2, 5, 11, 13

34. A. K. Pizer. Ramanujan graphs and Hecke operators. *Bulletin of the American Mathematical Society*, 23(1):127–137, 1990. 25
35. M. Prabhakaran and R. Xue. Statistically hiding sets. In *Cryptographers Track at the RSA Conference*, pages 100–116. Springer, 2009. 2, 6
36. R. Schoof. Counting points on elliptic curves over finite fields. *Journal de Théorie des Nombres de Bordeaux*, 7(1):219–254, 1995. 24
37. Y. Seurin. New constructions and applications of trapdoor DDH groups. In *International Workshop on Public Key Cryptography*, pages 443–460. Springer, 2013. 2, 6, 7, 8, 12, 18
38. J. H. Silverman. *The arithmetic of elliptic curves*, volume 106. Springer Science & Business Media, 2009. 4
39. J. Tate. Endomorphisms of abelian varieties over finite fields. *Inventiones Mathematicae*, 2(2):134–144, 1966. 24
40. E. R. Verheul. Evidence that XTR is more secure than supersingular elliptic curve cryptosystems. *Journal of Cryptology*, 17(4):277–296, 2004. 5

Supplementary material

A Applications

A.1 Identification scheme

By observing that we have not only a trapdoor DDH, but a more general trapdoor pairing construction, we can improve upon the Dent–Galbraith identification scheme. Essentially, in their scheme a party has a secret pairing and identifies itself by showing that it can distinguish if a challenge tuple is a DDH tuple or not. As the prover can cheat with probability $\frac{1}{2}$, this protocol must be repeated many times to ensure a negligible cheating probability. By relying on a computational problem instead, we can remove the need for repetition.

- **Setup.** Let $(\mathbb{G}, \mathbb{G}_T, P, \tau) \leftarrow \mathit{Gen}(1^\lambda)$ be a trapdoor pairing group. The prover's secret key will be the trapdoor τ, which allows to compute a non-degenerate pairing
$$e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T,$$
as described above. The public key is $(\mathbb{G}, \mathbb{G}_T, P)$, where $P \leftarrow \mathbb{G}$.
- **Interaction.**
  - The prover picks $r \leftarrow \mathbb{F}_p$ and sends $g = e(P, P)^r$ to the verifier.
  - The verifier picks $a, b \leftarrow \mathbb{F}_p$ and sends aP, bP to the prover.
  - The prover computes $u = e(aP, bP)^r$, and sends u to the verifier.

  The verifier accepts the proof if and only if $u = g^{ab}$.

Clearly a cheating prover can solve the Target CDH problem. By assumption this will only happen with negligible probability, so there is no need to repeat the protocol. We formalize the security in the following theorem.

Theorem 4 *The identification scheme above is complete, sound and zero-knowledge when instantiated in a trapdoor pairing group.*

Proof: Completeness is easy to check, as

$$u = e(aP, bP)^r = (e(P, P)^r)^{ab} = g^{ab}.$$

For soundness, assume that a cheating prover $\mathcal{A}$ can produce accepting proofs. We build an adversary $\mathcal{B}$ to break the Target CDH problem as follows: upon receiving $(\mathbb{G}, \mathbb{G}_T, P)$, adversary $\mathcal{B}$ passes them to $\mathcal{A}$, who answers with g. $\mathcal{B}$ forwards g to the challenger and receives aP, bP, which are again sent to $\mathcal{A}$, who answers with u. Because the proof is accepting, we have that $u = g^{ab}$, so u is a valid solution for the Target CDH problem.

We argue that the scheme is zero-knowledge, that is, no information about the trapdoor pairing is leaked. To do so, we describe a simulator that produces, without knowledge of the trapdoor, transcripts indistinguishable from transcripts from honest executions of the scheme.

– **Simulator.** Pick $g \leftarrow \mathbb{G}_T$. Choose $a, b \leftarrow \mathbb{F}_p$, and set $u = g^{ab}$. The first message of the transcript is g, the second is (aP, bP), and the third is u.

Clearly the second message is distributed as in a real execution. In the third message u is correctly distributed as long as g is correctly distributed, and g is in both cases a uniformly random element of the target group. □

A.2 Breaking anonymity in ElGamal voting

We recall the ElGamal encryption scheme.[7] A group $\mathbb{G}$ and a generator P are publicly known. A user's secret key is $sk = x \leftarrow \mathbb{F}_p$ and the corresponding public key is $pk = Q = xP$. To encrypt a message $m \in \mathbb{F}_p$, we choose randomness $r \leftarrow \mathbb{F}_p$ and set

$$\mathsf{Enc}_{pk}(m; r) = (rP, mP + rQ).$$

To decrypt a ciphertext (C_1, C_2), we compute

$$\mathsf{Dec}_{sk}(C_1, C_2) = \log_P(C_2 - xC_1).$$

Note that the discrete logarithm of $C_2 - xC_1$ can be efficiently computed only if the set of possible messages is small. This is often the case in voting, in which the set of messages is a small set of candidates, or even just 'yes'/'no'.

We observe that an encryption, together with public information, contains a DDH tuple. Indeed, consider

$$(P, pk, C_1, C_2 - mP) = (P, xP, rP, xrP).$$

Hence, if someone can solve the DDH problem, and the set of possible messages is small enough, it is possible to identify the message by checking whether $(P, pk, C_1, C_2 - \tilde{m}P)$ is a DDH tuple for each possible message $\tilde{m}$, until a positive result is found.

This rules out the use of supersingular curves for electronic voting and similar purposes, as the party that sets up the group $\mathbb{G}$ potentially has access to a trapdoor that allows to open any vote. This idea extends naturally to other contexts. For example, usually zero-knowledge proofs involve using commitments, and sometimes ElGamal encryption is used as a commitment there. We note that a DDH or pairing trapdoor would allow to break the hiding property of the commitment scheme, hence compromising the security of the zero-knowledge proof and the protocols derived from it.

B Potential extensions of the attacks

In the previous subsections we have merely applied existing results from the literature to demonstrate that the parameters suggested by Dent–Galbraith are

[7] We present the variant known as lifted ElGamal, in which the message is an element of $\mathbb{F}_p$ instead of $\mathbb{G}$.

insecure when the group order is revealed. We expect more elaborate and dedicated algorithms to give better results and to require further increases of the parameters.

In particular, we expect further lattice attacks to exist in the case $r_1 \neq r_2$ when $R := r_1 r_2$ is known but cannot be efficiently factored (the setting originally proposed by Dent–Galbraith, but with bigger parameters). In this case we have two equations (with variables x, y, r_1, r_2):

$$\begin{cases} N = (r_1 x - 1)(r_2 y - 1), \\ R = r_1 r_2. \end{cases}$$

One could apply multivariate generalizations of Coppersmith's method and deduce new constraints on the parameters' sizes; we leave this to further work.

As this section demonstrates, selecting parameters for Dent–Galbraith's trapdoor DDH group construction is far from trivial. Note that our new construction does not have this issue as it uses supersingular curves over $\mathbb{F}_{p^2}$ instead of $\mathbb{Z}_N$.

C Background

C.1 Elliptic curves

Let E_1, E_2 be elliptic curves defined over a finite field $\mathbb{F}_q$. An isogeny between E_1 and E_2 is a surjective morphism which sends the point of infinity of E_1 to the point of infinity of E_2. An isogeny induces a group homomorphism from E_1 to E_2. The degree of the isogeny is its degree as a finite map of curves. When the isogeny is separable, then the degree of the isogeny is equal to the cardinality of its kernel. Given an isogeny ϕ from E_1 to E_2 of degree d, there exists another isogeny $\hat{\phi}$ from E_2 to E_1 of degree d called the dual isogeny with the property that $\phi \circ \hat{\phi} = [d]$, where $[d]$ denotes multiplication by d on E_2. We say that two elliptic curves are isogenous if there exists an isogeny between them. By a theorem of Tate [39], two elliptic curves defined over $\mathbb{F}_q$ are isogenous over $\mathbb{F}_q$ if and only if they have the same number of points over $\mathbb{F}_q$.

An isogeny from an elliptic curve E to itself is called an endomorphism of E, and the endomorphisms of an elliptic curve form a ring under addition and composition. A well-known theorem of Deuring states that the endomorphism ring of an elliptic curve defined over a finite field is either an order in a quadratic number field (such curves are called ordinary) or a maximal order in a quaternion algebra (such curves are called supersingular). Another way to distinguish ordinary and supersingular curves is through their number of points over their field of definition. An elliptic curve defined over $\mathbb{F}_{p^k}$ (where p is a prime number) is supersingular if and only if $|E(\mathbb{F}_{p^k})| \equiv 1 \bmod p$, or equivalently, if and only if the trace of Frobenius is divisible by p. One can compute the number of points of an elliptic curve using Schoof's algorithm and its variants [36]. In particular, one can test supersingularity in polynomial time.

Let ℓ be a prime number different from p. Two elliptic curves are ℓ-isogenous if there exists an isogeny of degree ℓ between them (note that this is a symmetric relation as an isogeny and its dual have the same degree). Thus one can consider the ℓ-isogeny graph of an isogeny class of elliptic curves, where the vertices correspond to isomorphism classes of elliptic curves and there is an edge between two vertices for every isogeny of degree ℓ between. The ℓ-isogeny graph in the case of supersingular curves is connected, $\ell+1$-regular and has the Ramanujan property [34].

Let $\mathbb{G}, \mathbb{G}_T$ be groups. In practice $\mathbb{G}$ will be a subgroup of an elliptic curve and $\mathbb{G}_T$ a subgroup of a finite field, so we write $\mathbb{G}$ additively and $\mathbb{G}_T$ multiplicatively. A symmetric pairing is a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, such that:

1. $e(P_1 + P_2, Q) = e(P_1, Q)e(P_2, Q)$.
2. $e(P, Q_1 + Q_2) = e(P, Q_1)e(P, Q_2)$.

The group $\mathbb{G}_T$ is called the target group.

Let E be an elliptic curve defined over a finite field of characteristic p and let m be an integer coprime to p. Let $E[m]$ denote the m-torsion subgroup of E. When $\mathbb{G} = E[m]$ and $\mathbb{G}_T$ is the multiplicative group of the mth roots of unity, then a well-known example of such a pairing is the Weil pairing. However, the Weil pairing has the property that $e(P, P) = 1$ for any point P on E. This is inconvenient as this implies that the Weil pairing is degenerate when restricted to a cyclic subgroup of the curve. One can remedy this by taking an endomorphism ϕ of E (in this context such a map is called a distortion map) and by instead considering the *modified Weil pairing* $\hat{e}(P, Q) = e(P, \phi(Q))$ where e denotes the Weil pairing. This way, for a general point P and a non-scalar endomorphism ϕ, the point P is not in the subgroup generated by $\phi(P)$.

C.2 Formal definitions of DL, CDH and DDH assumptions

Definition 6 *Let* Gen *be an algorithm that takes as input a security parameter* 1^λ *and returns* $(\mathbb{G}, n, g)$*, where* $\mathbb{G}$ *is a group of order* n*, and* $g \in \mathbb{G}$*.*

- *The DLP assumption holds with respect to* Gen *if for all PPT algorithms* $\mathcal{A}$*, we have that*

$$\Pr\left[(\mathbb{G}, n, g) \leftarrow \mathsf{Gen}(1^\lambda); h \leftarrow \langle g \rangle; x \leftarrow \mathcal{A}(\mathbb{G}, n, g, h) : h = g^x\right] \approx 0.$$

- *The CDH assumption holds with respect to* Gen *if for all PPT algorithms* $\mathcal{A}$*, we have that*

$$\Pr\left[(\mathbb{G}, n, g) \leftarrow \mathsf{Gen}(1^\lambda); a, b \leftarrow \mathbb{Z}_{|g|}; z \leftarrow \mathcal{A}(\mathbb{G}, n, g, g^a, g^b) : z = g^{ab}\right] \approx 0.$$

- *The DDH assumption holds with respect to* Gen *if for all PPT algorithms* $\mathcal{A}$*, we have that*

$$\left|\Pr\left[\begin{array}{c}(\mathbb{G}, n, g) \leftarrow \mathsf{Gen}(1^\lambda); a, b \leftarrow \mathbb{Z}_{|g|}; z_0 = g^{ab}; z_1 \leftarrow \langle g \rangle; \\ ch \leftarrow \{0,1\}; ch^* \leftarrow \mathcal{A}(\mathbb{G}, n, g, g^a, g^b, z_{ch})\end{array} : ch = ch^*\right] - \frac{1}{2}\right| \approx 0.$$