Chen and Zhang *Cybersecurity (2023)6:54*
https://doi.org/10.1186/s42400-023-00189-2

Cybersecurity

RESEARCH

Open Access

Verifiable delay functions and delay encryptions from hyperelliptic curves

Chao Chen[1,2] and Fangguo Zhang[1,2*]

Abstract

Verifiable delay functions (VDFs) and delay encryptions (DEs) are two important primitives in decentralized systems, while existing constructions are mainly based on time-lock puzzles. A disparate framework has been established by applying isogenies and pairings on elliptic curves. Following this line, we first employ Richelot isogenies and non-degenerate pairings from hyperelliptic curves for a new verifiable delay function, such that no auxiliary proof and interaction are needed for the verification. Then, we demonstrate that our scheme satisfies all security requirements, in particular, our VDF can resist several attacks, including the latest attacks for SIDH. Besides, resorting to the same techniques, a secure delay encryption from hyperelliptic curves is constructed by modifying Boneh and Frankiln's IBE scheme, which shares the identical setup with our VDF scheme. As far as we know, these schemes are the first cryptographic applications from high-genus isogenies apart from basic protocols, i.e., hash functions and key exchange protocols.

Keywords Verifiable delay functions, Delay encryptions, Hyperelliptic curves, Richelot isogenies, Pairings

Introduction

Verifiable delay function (VDF), first introduced by Boneh et al. (2018), is a function $f : \mathcal{X} \to \mathcal{Y}$ that requires a prescribed amount of time for evaluations, even if many parallel computation resources are employed, while the result can be verified efficiently. The most crucial requirement demands that evaluation, as a slow function, must be realized in at least T sequential steps and no acceleration exists. Such scheme allows a prover to demonstrate that a certain amount of time has elapsed. Furthermore, the VDFs with more functionality, e.g., tight VDFs (Döttling et al. 2020) and continuous VDFs (Ephraim et al. 2020), were also proposed for particular situations.

Due to efficient verifications, VDFs have been applied broadly in cryptography, especially for the decentralized setting. A direct application is to construct a trustworthy *randomness beacon* (Rabin 1983), where the beacon is given by a VDF with a long delay on the entropy source, so the malicious participant can not obtain his advantages to adjust the market within a short time. Furthermore, based on the "commit-and-reveal" paradigm, multiparty randomness can be achieved by replacing commitments with VDFs, illustrated in Lenstra and Wesolowski (2017). Another usage of VDFs is to lower the energy consumption of blockchains based on proofs-of-work. Namely, an ingenious method (Cohen and Pietrzak 2018) combines proofs-of-resources with incremental VDFs to achieve Consensus from Proof of Resources. Moreover, proof of data replication (Armknecht et al. 2016; Juels and Kaliski 2007) and computational timestamping (Kiayias et al. 2017) can be realized with VDFs, where more discussions can be found in Boneh et al. (2018).

After the proposal of VDF (Boneh et al. 2018), various instantiations have been established, where VDF can be directly achieved using incrementally verifiable

*Correspondence:
Fangguo Zhang
isszhfg@mail.sysu.edu.cn
[1] School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China
[2] Guangdong Province Key Laboratory of Information Security Technology, Guangzhou 510006, China

Springer Open

© The Author(s) 2023. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

computation (Valiant 2008). Apart from the high-level ideas, class groups and injective rational maps have been leveraged for establishing VDFs (Boneh et al. 2018; Wesolowski 2020).

In practice, computing modular exponentiation is an elegant choice for sequentially slow evaluation functions, where extracting modular square roots in $\mathbb{F}_p$ (Dwork and Naor 1992) and repeated squaring in an RSA group (Rivest et al. 1996) were instantiated for this problem. Thus, a natural idea is to modify the above functions for practical VDF schemes. Specifically, the first can be efficiently verified by a modular square, so it turns out to be a VDF immediately (Boneh et al. 2018). Regrettably, the delay parameter of this approach is only about $O(\log p)$, which would be smaller considering the parallelism of field multiplications, where Lenstra and Wesolowski (2017) introduced Sloth to realize parallel computation for modular square roots.

In contrast, the second was generated from the famous time-lock puzzles (Rivest et al. 1996), i.e., utilizing RSA modulus $N = pq$, the puzzle is $y = x^{2^T} \bmod N$ from a random $x \in \mathbb{Z}_N^*$. Besides, one obtaining $\varphi(N)$ can evaluate y efficiently via reducing the exponent $e \equiv 2^T \bmod \varphi(N)$, while others must compute T sequential modular squares. Following this line, Wesolowski (2020) established an efficient interactive protocol to verify the output y publicly, being non-interactive via the Fiat–Shamir paradigm (Fiat and Shamir 1986). Namely, the verifier sends a random small prime ℓ, and the prover replies with $z = x^{\lfloor 2^T/\ell \rfloor}$, then the verifier accepts when $y = z^\ell x^r$ with $r = 2^T \bmod \ell$. To reduce the proof size, Pietrzak (2019) introduced another interactive protocol via substituting $\mathbb{Z}_N^*$ by the group $QR_N^+ := \{|x|; x = z^2 \bmod N, z \in \mathbb{Z}_N^*\}$ with two strong primes p and q, so that the prover outputs a proof with $O(\log T)$ group elements and the verification only needs $O(\log T)$ with the "halving protocol". Recently, Loe et al. (2022) presented P-VDF without the large proofs, where they leveraged the Blum integer $N = pq$ with $p \equiv q \equiv 3 \bmod 4$ such that the verification relies on the factorization of N. As a result, the efficiency of verification is the fastest among all existing VDFs while we must generate Blum integers for different instantiations.

One notable breakthrough was established in 2019, when De Feo et al. (2019) presented a new framework of VDFs from the BLS signature (Boneh et al. 2001). In this paradigm, the long sequences of isogenies were employed as the slow evaluation functions, while the results can be efficiently verified via non-degenerate pairings, then two schemes were first instantiated from isogenies between elliptic curves over $\mathbb{F}_p$ and $\mathbb{F}_{p^2}$, respectively. More specifically, they employed chains of low-degree isogenies for a "slow" evaluation function since the isogeny computation still takes T sequential steps, while the pairing can be evaluated in $poly(\log N)$ time.

Substituting pairings by the succinct non-interactive arguments (SNARGs), the first post-quantum secure isogeny-based VDF (Chávez-Saab et al. 2021) was constructed without trusted setups, while the verification terminated in quasi-logarithmic time.

Motivated by original time-lock puzzles and VDFs, a new primitive named delay encryption (DE) (Burdges and De Feo 2021) was introduced by Burdges and De Feo, viewed as a time-lock version of Identity-based Encryptions (IBE). Yet it is called an encryption scheme, there are no secrets, and the critical concept is *session*, which is generated by a session identifier and is hard to predict. In particular, the function Extract, as the defining algorithm of generating a session key from an identifier, must run sequentially and slowly. Surprisingly, the instantiations of certain VDFs can be employed for DEs immediately. Namely, the initial construction in Burdges and De Feo (2021) followed the same roadmap from isogeny-based VDF (De Feo et al. 2019) by modifying the IBE scheme (Boneh and Franklin 2003), and it is facile to construct DE from P-VDF (Loe et al. 2022). For cryptographic deployments, some protocols derived from time-lock puzzles, i.e., Vickrey auctions and electronic votings, would obtain additional advantages via utilizing DEs.

Nowadays, isogenies on hyperelliptic curves (Flynn and Ti 2019) have been a hotspot for the shortest key sizes. Although it was a mathematical problem (Lubicz and Robert 2012; Smith 2006) investigated for decades, its cryptographic constructions are relatively new topics. Due to the complicated formulae, efficient implementations are crucial problems with abundant improvements (Bruin et al. 2014; Cosset and Robert 2015; Flynn 2015; Kunzweiler 2022), while only hash functions (Castryck et al. 2020) and key exchange protocols (Flynn and Ti 2019) have been presented for practical constructions. It is shown that the cryptosystem based on isogenies between hyperelliptic Jacobians has a smaller key size than that on elliptic curves (Costello and Smith 2020; Flynn and Ti 2019), thus it is natural to construct more functional applications from isogenies on hyperelliptic curves.

Our contributions Following the framework of isogeny-based VDF, we establish the verifiable delay function and delay encryption from hyperelliptic curves, which are the first cryptographic applications utilizing isogenies on supersingular hyperelliptic curves. More specifically, our contributions are summarized as follows.

- We first employ Richelot isogenies and non-degenerate pairings on hyperelliptic curves to establish a verifiable delay function without additional interaction, where the output is verified via pairings such that

no proof is required. Then, we demonstrate that our scheme satisfies all security requirements, i.e., the parameters of our scheme are fixed to resist several known attacks. In particular, the defining property "sequentiality" holds under the assumption of high-genus isogeny shortcut problem, as a generalization of that for elliptic curves. As an additional contribution, we illustrate that isogeny-based VDF can resist recent attacks on SIDH.

- Following the same framework, we modify Boneh and Frankiln's IBE scheme for a Δ-IND-CPA secure delay encryption from hyperelliptic curves with the same instantiations, i.e., the session key is obtained through T sequential Richelot isogenies. Afterwards, we show that the scheme is secure having analogous merits and demerits as our VDF scheme.

Organization The rest of this paper is organized as follows. The "Preliminaries on hyperelliptic curves and isogenies" section provides necessary preliminaries on hyperelliptic curves and Richelot isogenies. In "Syntax of verifiable delay functions and delay encryptions" section, the definition and security requirements of VDFs and DEs are reviewed. The verifiable delay function from hyperelliptic curves is depicted in "Verifiable delay functions from hyperelliptic curves" section, followed by the security analysis. The "Delay encryptions from hyperelliptic curves" section presents the delay encryption from hyperelliptic curves. The last section concludes our work.

Preliminaries on hyperelliptic curves and isogenies

In this section, we recall some necessary mathematical backgrounds of hyperelliptic curves, pairings, and isogenies.

Hyperelliptic curves

Let $\overline{\mathbb{F}}_q$ be the algebraic closure of the finite field $\mathbb{F}_q$ with characteristic $p > 3$. A hyperelliptic curve C of genus 2 over $\mathbb{F}_q$ is given by the following equation:

$$C : y^2 = f(x),$$

where $f(x)$ is a squarefree polynomial of degree 5 or 6 such that there are no solutions $(x, y) \in \overline{\mathbb{F}}_q \times \overline{\mathbb{F}}_q$ simultaneously satisfying the equation $y^2 = f(x)$ and the partial derivative equations $y = 0$ and $f'(x) = 0$. For any algebraic extension $\mathbb{F}_{q^k}$ of $\mathbb{F}_q$, we consider the set

$$C(\mathbb{F}_{q^k}) := \{(x, y) \in \mathbb{F}_{q^k} \times \mathbb{F}_{q^k} \mid y^2 = f(x)\} \cup \begin{cases} \{\infty\}, & \text{if } \deg(f) = 5; \\ \{\infty_+, \infty_-\}, & \text{if } \deg(f) = 6; \end{cases}$$

called the set of $\mathbb{F}_{q^k}$-rational points on C.

The set $C(\mathbb{F}_{q^k})$ does not form a group, but we can embed C into an abelian variety of dimension 2, which is called the Jacobian of C and denoted by J_C. The Jacobian J_C is isomorphic to the divisor class group of degree zero Pic_C^0. Let $\mathcal{O}$ be the identity element of J_C.

Every divisor in Jacobian J_C over a field K can be expressed in Mumford representation as a pair $(u(x), v(x))$ of polynomials in $K[x]$, such that $u(x)$ is monic, and $u(x)$ divides $f(x) - v(x)^2$ with $\deg v(x) < \deg u(x) \leq 2$. Let $P_1 = (x_1, y_1), P_2 = (x_2, y_2)$ be two points on the hyperelliptic curve C, then the Mumford representation $(u(x), v(x))$ associated with two points satisfies $u(x_i) = 0$ and $v(x_i) = y_i$, for $i = 1, 2$. For $r \in \mathbb{N}$, we define $J_C[r] := \{D \in J_C \mid rD = \mathcal{O}\}$ as the r-torsion subgroup of J_C.

Hyperelliptic pairings

Pairings on hyperelliptic curves are useful tools in cryptology. The definitions of two familiar pairings on hyperelliptic curves are summarized as follows.

Let C be a hyperelliptic curve of genus 2 over $\mathbb{F}_q$, and J_C be the corresponding Jacobian. Let r be a divisor of $\#J_C$, and coprime to q. The embedding degree is the smallest positive integer k such that $r \mid (q^k - 1)$. The group of r-th roots of unity in $\mathbb{F}_{q^k}^*$ is denoted by $\mu_r = \{z \in \mathbb{F}_{q^k}^* \mid z^r = 1\}$.

The Weil pairing is a non-degenerate bilinear map

$$J_C(\mathbb{F}_{q^k})[r] \times J_C(\mathbb{F}_{q^k})[r] \longrightarrow \mu_r,$$

which is denoted as $e_r(D_1, D_2)$. The Tate-Lichtenbaum pairing is a non-degenerate bilinear map

$$J_C(\mathbb{F}_{q^k})[r] \times J_C(\mathbb{F}_{q^k})/rJ_C(\mathbb{F}_{q^k}) \longrightarrow \mathbb{F}_{q^k}^*/(\mathbb{F}_{q^k}^*)^r,$$

which is denoted as $\langle D_1, D_2 \rangle_r$. To achieve cryptographic applications, we consider the reduced (or modified) pairing

$$t_r(D_1, D_2) = \langle D_1, D_2 \rangle_r^{\frac{q^k-1}{r}}.$$

Similar to the pairings of elliptic curves, Miller's algorithm (Cohen et al. 2005; Miller 2004) is used to compute hyperelliptic pairings. For more detailed discussions, it refers to Galbraith et al. (2007).

Richelot isogenies

It is well-known that we can compute an isogenous elliptic curve from a given kernel through Vélu's formula (Vélu 1971), which is the foundation of isogeny-based cryptography. Nevertheless, with the growth of genus, there is no efficient algorithm to evaluate isogenies between Jacobians.

Since the Jacobian J_C of a curve C is a principally polarized abelian variety (PPAV), we could consider *isogeny* of principally polarized abelian varieties, which is a finite dominant homomorphism of abelian varieties A, and the kernel of isogeny is a finite isotropic group. The Richelot isogeny is the simplest isogeny whose kernel is contained in the 2-torsion subgroup $J_C[2]$ from a genus-2 hyperelliptic curve. Smith (2006) summarized the Richelot isogenies on Jacobians of genus 2, whose kernel is maximal isotropic with regards to the 2-Weil pairing.

Proposition 1 (Smith 2006) *Let R be a proper, non-trivial subgroup of $J_C[2]$. If R is the kernel of an isogeny between principally polarized abelian surfaces, then R is a maximal 2-Weil isotropic subgroup of $J_C[2]$ (that is, the 2-Weil pairing restricts trivially to R, and R is not properly contained in any other such subgroup).*

Now, we present some facts about Richelot isogenies for our construction. Let $C : y^2 = f(x)$ be a genus-2 hyperelliptic curve and J_C be its Jacobian, where $f(x) = c_0 \prod_{i=1}^{6}(x - \alpha_i)$[1]. Then, all 2-torsion divisors of J_C are

$$\{\mathcal{O}\} \cup \{[(\alpha_i, 0) - (\alpha_j, 0)];\ 1 \le i < j \le 6\},$$

where the square brackets denote the equivalence classes of divisors. For a maximal isotropic subgroup with regards to the 2-Weil pairing, the group contains three non-trivial elements such that all α_i, $1 \le i \le 6$, occur exactly once in all the representations of divisors. Thus, there are fifteen disparate isogenous PPAVs from a Jacobian, which are determined by the sets of pairwise coprime quadratic factors of $f(x)$.

Definition 1 A *quadratic splitting* of a squarefree degree 6 (resp. degree 5) polynomial $f(x) \in \mathbb{F}_q[x]$ is an unordered triple $(G_1, G_2, G_3) \in \overline{\mathbb{F}}_q[x]$ of quadratic (resp. two quadratic and a linear) polynomials satisfying $f(x) = G_1(x)G_2(x)G_3(x)$ under the equivalence

$$\{G_1, G_2, G_3\} \sim \{\beta G_1, \gamma G_2, (\beta\gamma)^{-1} G_3\}, \text{ for all } \beta, \gamma \in \overline{\mathbb{F}}_q^*.$$

The next proposition provides the codomain of Richelot isogeny, where we refer Bruin and Doerksen (2011), Cassels and Flynn (1996), Smith (2006) for more details.

Proposition 2 (Smith 2006) *Let $C : y^2 = G_1(x)G_2(x)G_3(x) \in \mathbb{F}_q[x]$ be a genus 2 curve such that the maximal 2-Weil isotropic subgroup G is determined by $\{G_1, G_2, G_3\}$, where $G_i(x) = g_{i,2}x^2 + g_{i,1}x + g_{0,i}$ for $i \in \{1, 2, 3\}$. Let $\phi : J_C \to A$ be the isogeny with kernel G and*

$$\delta := \det \begin{pmatrix} g_{1,2} & g_{1,1} & g_{1,0} \\ g_{2,2} & g_{2,1} & g_{2,0} \\ g_{3,2} & g_{3,1} & g_{3,0} \end{pmatrix}.$$

1. *If $\delta \neq 0$, then A is isomorphic to the Jacobian of a genus-2 curve*

   $$C' : y^2 = \delta^{-1} H_1(x) \cdot H_2(x) \cdot H_3(x)$$

   with $H_1 := G_2'G_3 - G_2G_3'$, $H_2 := G_3'G_1 - G_3G_1'$, $H_3 := G_1'G_2 - G_1G_2'$, where G_i' is the derivative of G_i. Moreover, the dual isogeny is determined by $\{H_1, H_2, H_3\}$.
2. *If $\delta = 0$, then A is isomorphic to a product of elliptic curves $E_1 \times E_2$, where two elliptic curves are defined by*

   $$E_1 : y^3 = \prod_{i=1}^{3}(a_{i,1}x + a_{i,2}) \quad \text{and} \quad E_2 : y^3 = \prod_{i=1}^{3}(a_{i,2}x + a_{i,1}),$$

   such that $a_{i,1}, a_{i,2}$ satisfy $G_i = a_{i,1}(x - t_1)^2 + a_{i,2}(x - t_2)^2$ for some $t_1, t_2 \in \mathbb{F}_q$.

Remark 1
For the second case, the map ϕ is induced by $\phi_1 \times \phi_2$, where

$$\begin{aligned} \phi_1 : C &\to E_1 \\ (x, y) &\mapsto \left(\tfrac{(x-t_1)^2}{(x-t_2)^2}, \tfrac{y}{(x-t_2)^3}\right) \end{aligned} \quad \text{and} \quad \begin{aligned} \phi_2 : C &\to E_2 \\ (x, y) &\mapsto \left(\tfrac{(x-t_2)^2}{(x-t_1)^2}, \tfrac{y}{(x-t_1)^3}\right). \end{aligned}$$

It is essential to evaluate the image of divisors in J_C under the isogeny ϕ for the first case of the above proposition. Nevertheless, Richelot isogenies work on the

[1] If $\deg(f) = 5$, we define $\alpha_6 = \infty$ and $x - \alpha_6 = 0 \cdot x + 1$

hyperelliptic curve, as the morphisms between hyperelliptic curves. For this aim, we map two points to a unique divisor on J_C, then the divisor is directly calculated from the image points. Furthermore, the above method can be realized via the Richelot correspondence $\mathcal{R} \subset C \times C'$ with

$$\begin{pmatrix} G_1(u)H_1(u_1) + G_2(u)H_2(u_1) = 0 \\ vv_1 = G_1(u)H_1(u_1)(u - u_1) \end{pmatrix}$$

for $(u, v) \in C$ and $(u_1, v_1) \in C'$. This correspondence presents the connection of points on hyperelliptic curves, but there are always two solutions for these equations. To fill the gap, Kunzweiler (2022) established an algorithm to uniquely determine the image on $J_{C'}$. Namely, two solutions determine a divisor, then two divisors from different points, generating the preimage divisor on J_C, compose the unique divisor, which is the image under Richelot isogenies.

Syntax of verifiable delay functions and delay encryptions

In this section, we review the model of verifiable delay functions (VDFs) and delay encryptions (DEs), followed by the security requirements.

Verifiable delay functions

The definition of verifiable delay function has been first established in Boneh et al. (2018). In general, a VDF contains three algorithms:

1. $\mathsf{Setup}(\lambda, T) \rightarrow (ek, vk)$: is an algorithm whose inputs are the security parameter λ and a delay parameter T. The outputs are an evaluation key ek and a verification key vk. We require that Setup runs in polynomial time of λ and T. Then, the input space $\mathcal{X}$ and the output space $\mathcal{Y}$ are determined by (ek, vk), where we assume that $\mathcal{X}$ is efficiently sampleable.
2. $\mathsf{Eval}(ek, s) \rightarrow (a, \tau)$: is a procedure to evaluate on input $s \in \mathcal{X}$. The outputs consist of $a \in \mathcal{Y}$ from s, and a (possibly empty) proof τ. The requirement of this procedure is the time of computation can not be less than T.
3. $\mathsf{Verify}(vk, s, a, \tau) \rightarrow \{\text{True, False}\}$: is a procedure to verify whether a is the correct output for s with the help of proof τ if necessary. In general, it is an efficient algorithm compared with Eval, i.e., running in $ploy(\lambda, T)$.

The VDF should satisfy three security properties: *Correctness*, *Soundness*, and *Sequentiality*. The formal definitions of security requirements are depicted below.

Correctness This property requires that every output of Eval must be accepted by Verify.

Definition 2 The VDFs are correct if for any λ, T, public parameters (ek, vk), and input s, if $(a, \tau) \leftarrow \mathsf{Eval}(ek, s)$, then $\mathsf{Verify}(vk, s, a, \tau)$ outputs true.

Soundness It states that the incorrect output $(\tilde{a}, \tilde{\tau})$, generated by any adversary without performing Eval, can not be accepted by the Verify.

Definition 3 (Soundness) A VDF is sound if for any λ, T, public parameters (ek, vk), and input s, if $(\tilde{a}, \tilde{\tau}) \neq \mathsf{Eval}(ek, s)$, then $\mathsf{Verify}(vk, s, \tilde{a}, \tilde{\tau})$ outputs true with negligible probability.

Sequentiality This is the defining property of VDFs. Namely, this property demands that it is impossible to evaluate the VDF faster than running Eval, even given a boundless amount of parallel computers and precomputations, which are generated after the setup of public parameters. Whereas, the adversary with $|\mathcal{Y}|$ processors can evaluate outputs by simultaneously trying all output in $\mathcal{Y}$. Therefore, it is crucial to bound the adversary's ability of parallelism. For more detailed discussions, please refer to Boneh et al. (2018), De Feo et al. (2019).

Definition 4 (*Sequentiality*) A VDF is sequential if no pair of randomized algorithms $\mathcal{A}_0$, which runs in total time $ploy(T, \lambda)$, and $\mathcal{A}_1$, which runs in parallel time less than T, can win the following sequential game with non-negligible probability.

1. $(ek, vk) \xleftarrow{R} \mathsf{Setup}(\lambda, T)$;
2. $L \xleftarrow{R} \mathcal{A}_0(\lambda, ek, vk, T)$;
3. $s \xleftarrow{R} \mathcal{X}$;
4. $\tilde{a} \xleftarrow{R} \mathcal{A}_1(L, ek, vk, s)$,

where winning is defined as outputing $\tilde{a} = a$, where $(a, \tau) = \mathsf{Eval}(ek, s)$.

Construction framework Inspired by pairing-based BLS signature scheme (Boneh et al. 2001), a construction framework of VDFs (De Feo et al. 2019) has been proposed, i.e., the framework is depicted as follows.

Let $e_X : X_1 \times X_2 \rightarrow G \subset \mathbb{F}_{q^k}$ and $e_Y : Y_1 \times Y_2 \rightarrow G \subset \mathbb{F}_{q^k}$ be non-degenerate pairings, where X_1, X_2, Y_1, Y_2, G are subgroups of order N, and k is denoted by the embedding degree. In addition, suppose that there is a pair of bijections $\phi : X_1 \rightarrow Y_1$ and $\hat{\phi} : Y_2 \rightarrow X_2$ such that the following diagram is commutative.

$$\begin{array}{ccc} X_1 \times Y_2 & \xrightarrow{\phi \times 1} & Y_1 \times Y_2 \\ {\scriptstyle 1\times\hat{\phi}}\downarrow & & \downarrow{\scriptstyle e_Y} \\ X_1 \times X_2 & \xrightarrow[e_X]{} & G \end{array}$$

Let P be the generator of X_1, then the system parameters are initialized by $(N, X_1, X_2, Y_1, Y_2, e_X, e_Y, P, \phi(P))$.

Delay encryptions

Motivated by VDFs, Burdges and De Feo (2021) introduced delay encryptions (DE), first instantiated with supersingular isogenies and pairings by modifying the famous IBE scheme (Boneh and Franklin 2003). DE is similar to the time-lock puzzles (Rivest et al. 1996), while the protocol outputs a session key rather than the proofs.

A DE consists of four algorithms:

1. $\mathsf{Setup}(\lambda, T) \to (ek, pk)$. Take a security parameter λ, a delay parameter T as inputs, and produce public parameters consisting of an extraction key ek and an encryption key pk. Setup must run in time $poly(\lambda, T)$ and the encryption key pk must have size $poly(\lambda)$, but the evaluation key ek is allowed to have size $poly(\lambda, T)$.
2. $\mathsf{Extract}(ek, id) \to idk$. Take the extraction key ek and a session identifier $id \in \{0,1\}^*$ as inputs, and output a session key idk. $\mathsf{Extract}$ is expected to run in time exactly T.
3. $\mathsf{Encaps}(pk, id) \to (c, k)$. Take the encryption key pk and a session identifier $id \in \{0,1\}^*$ as inputs, and output a ciphertext c and a key k. Encaps must run in time $poly(\lambda)$.
4. $\mathsf{Decaps}(pk, id, idk, c) \to k$. Take the encryption key pk, a session identifier id, a session key idk, and a ciphertext c as inputs, and output a key k. Decaps must run in time $poly(\lambda)$.

A DE scheme is correct if for any $(\mathsf{ek}, \mathsf{pk}) = \mathsf{Setup}(\lambda, T)$ and any $id \in \{0,1\}^*$,

$$\begin{aligned} idk = &\mathsf{Extract}(ek, id) \wedge (c, k) = \mathsf{Encaps}(pk, id) \Rightarrow \\ &\mathsf{Decaps}(pk, id, idk, c) = k. \end{aligned}$$

As an encryption scheme, the security of DE is similar to that of most public key encryption schemes, i.e., in particular of the IBE schemes. Whereas, one additional requirement of DE is that it is negligible to output idk for a random identifier id in time less than T. The security games of DE are depicted in Burdges and De Feo (2021).

Verifiable delay functions from hyperelliptic curves

In this section, we establish the concrete VDF under the framework in "Syntax of verifiable delay functions and delay encryptions" section, utilizing the Richelot isogenies and Weil pairings from supersingular hyperelliptic curves, then the security analysis is presented.

Our scheme

Following the framework in De Feo et al. (2019), we introduce the VDF from genus-2 hyperelliptic curves.

The prime is the form $p = 2^T \ell f - 1$ such that $p + 1$ has a large prime factor ℓ. Then, we leverage the algorithm in Burdges and De Feo (2021) to generate two trusted setups, i.e., two supersingular elliptic curves E_1, E_2 over $\mathbb{F}_{p^2}$, and transform the above curves into a supersingular hyperelliptic curve C, then the Jacobian J_C is obtained. Let $e_\ell(\cdot,\cdot)$ be a non-degenerate Weil pairing on $J_C[\ell]$.

From the supersingularity of J_C, we have $\#J_C(\mathbb{F}_{p^2}) = (p+1)^4$, and $J_C[2^T] \cong \mathsf{C}_{2^T}^4$ is a subgroup with four generators, where C_n is a cyclic group of order n. Flynn and Ti (2019) demonstrated that the maximal 2^n-isotropic subgroups of J_C must be isomorphic to $\mathsf{C}_{2^n} \times \mathsf{C}_{2^j} \times \mathsf{C}_{2^{n-j}}$, where $0 \le j \le \lfloor n/2 \rfloor$. To fulfill the condition of maximal isotropy, the secret selection method has been established in Flynn and Ti (2019), so we leverage this algorithm to create the kernel subgroup G with three generators Q_1, Q_2, Q_3 such that the isogeny $\phi : C \to C'$ is fixed[2], i.e., the hyperelliptic curve C' is decided by ϕ with kernel G. Immediately, the dual isogeny $\hat{\phi}$ is determined.

In practice, we decompose the isogeny into a sequence of T Richelot isogenies so that the dual isogeny $\hat{\phi}$ can be evaluated in the linear time of T.

Remark 2

We know that every 2-dimension supersingular PPAV is isomorphic to either the Jacobian of a genus-2 hyperelliptic curve or a product of two elliptic curves, and the second case occurs with a probability $10/(p+10)$ *(Castryck et al. 2020). Upon our choice, the probability of the intermediate PPAVs isomorphic to a product of elliptic curves is negligible. Even if this event has occurred, we can simply choose another kernel group* G' *to evaluate a new isogeny with overwhelming probability.*

Since the prime is in the particular form, we could sample an ℓ-torsion divisor $P \in J_C[\ell]$, and $X_1 = \langle P \rangle$ is a subgroup of order ℓ. From the isogeny $\phi : J_C \to J_{C'}$, we

[2] There are several choices of G for given n, j, and the decomposition of the isogeny derived from given G is not unique, where the details can be found in Flynn and Ti (2019, Sect. 2.2).

$\mathsf{Setup}(\lambda, T)$

1. Choose primes ℓ, p according to the security parameter λ;
2. Select a Jacobian J_C from a supersingular hyperelliptic curve $C/\mathbb{F}_{p^2}$;
3. Having fixed a kernel subgroup $G \subset J_C$, perform the non-backtracking $(2^T, 2^T)$-isogeny $\phi : J_C \to J_{C'}$ such that its dual is determined by $\hat{\phi}$;
4. Choose a generator $P \in J_C[\ell]$, and evaluate $\phi(P)$;
5. Output $(\mathsf{ek}, \mathsf{vk}) = \left(\hat{\phi}, (J_C, J_{C'}, P, \phi(P))\right)$.

$\mathsf{Eval}(\hat{\phi}, S \in Y_2)$

1. Evaluate and output $\hat{\phi}(S)$.

$\mathsf{Verify}(J_C, J_{C'}, P, \phi(P), S, \hat{\phi}(S))$

1. Verify $\hat{\phi}(S) \in X_2$;
2. Verify $e_\ell(P, \hat{\phi}(S)) = e_\ell(\phi(P), S)$.

Fig. 1 Verifiable delay functions from hyperelliptic curves

know that $\phi(P) \in J_{C'}$ is still an ℓ-torsion divisor. We set $Y_1 = \langle \phi(P) \rangle$. After that, we output the evaluation key and verification key as $\hat{\phi}$ and $(J_C, J_{C'}, P, \phi(P))$, respectively.

Similarly, $J_{C'}[\ell]$ has four generators and let $e'_\ell(\cdot, \cdot)$ be a non-degenerate Weil pairing. Since the map $R \mapsto e'_\ell(R, \phi(P))$ is surjective, there exists at least one divisor $Q \in J_{C'}$ such that $e'_\ell(\phi(P), Q) \neq 1$, i.e., $Q \notin \langle \phi(P) \rangle$. We can randomly sample a divisor $Q \in J_{C'}[\ell] \backslash \langle \phi(P) \rangle$, and set $Y_2 = \langle Q \rangle \subset J_{C'}[\ell]$. For every divisor $S \in Y_2$, $e'_\ell(Q, S) = 1$ is always holds while $e'_\ell(\phi(P), S) = 1$ can not be satisfied unless $S = \mathcal{O}$. In the same way, we have $X_2 = \left\langle \hat{\phi}(Q) \right\rangle$.

Remark 3

1. This divisor Q can be obtained with probability $(\ell - 1)/\ell$. If it fails, we can sample another divisor with overwhelming probability.

2. Since the degree of isogeny is coprime to ℓ, we can also select a generator Q' in $J_C[\ell]$ and obtain the image $\phi(Q')$ under the isogeny ϕ.

The four groups X_1, X_2, Y_1, Y_2 are all cyclic groups, so it is facile to uniformly sample a point from these subgroups. The function Eval takes a random divisor $S \in Y_2$ and outputs the image $\hat{\phi}(S)$ under the isogeny $\hat{\phi}$. For the verification Verify, first check

$$e_\ell(\hat{\phi}(Q), \hat{\phi}(S)) \stackrel{?}{=} 1,$$

otherwise, the verification fails. After that, $\hat{\phi}(S)$ passes the verification if

$$e_\ell(P, \hat{\phi}(S)) = e'_\ell(\phi(P), S).$$

Our VDF scheme is depicted in Fig. 1.

Security analysis

Now we present the security analysis of our VDF scheme, i.e., three properties are all satisfied.

Theorem 1 *Our scheme is correct and sound.*

Proof

Assume $S = aQ \in Y_2$ for $a \in \mathbb{Z}/\ell\mathbb{Z}$, then the honest output $\hat{\phi}(S) = a\hat{\phi}(Q) \in X_2$, so it holds that $e_\ell(\hat{\phi}(Q), \hat{\phi}(S)) = 1$. The equation

$$e_\ell(P, \hat{\phi}(S)) = e'_\ell(\phi(P), S) \tag{1}$$

comes from Mumford (1970). Thus, the legitimate output can pass the verification.

For the soundness, we know that $e_\ell(\hat{\phi}(Q), \cdot)$ is degenerate on X_2. However, the map $S' \mapsto e_\ell(\hat{\phi}(Q), S')$ is surjective, so a random divisor $S' \in J_{C'}[\ell]$ satisfies $e_\ell(\hat{\phi}(Q), S') = 1$ with a probability $1/\ell$. The second equation $e_\ell(P, S') = e'_\ell(\phi(P), S)$ is satisfied if and only if $e_\ell(P, S') = e_\ell(P, \hat{\phi}(S))$, which indicates $e_\ell(P, S' - \hat{\phi}(S)) = 1$. Then, from the surjectivity of $e_\ell(P, S')$, we know it occurs with a probability $1/\ell$.

Thus the verification succeeds with a probability $1/\ell^2$ when the output is a random divisor $S' \in J_{C'}[\ell]$. □

Table 1 Complexities of the attacks on the sequentiality of our VDF, consisting of the cases for classical and quantum computers

Attacks	Classical	Quantum
Computing shortcuts	$\tilde{O}(p^{g-1})$	$\tilde{O}(\sqrt{p^{g-1}})$
Pairing inversion	$L_p(1/3)$	$poly(\log p)$

Remark 4
The perfect soundness for isogeny-based VDF (De Feo et al. 2019) is invalid here. There are four generators in $J_{C'}[\ell]$, so two equations for verification can not determine a unique divisor.

Although *Sequentiality* is the most crucial property, it is hard to illustrate that there is no algorithm of "running in parallel time less than T". We now shift our attention to the following problem for isogenies between hyperelliptic curves, which has been introduced for elliptic curves in De Feo et al. (2019).

Definition 5 Let J_C be the Jacobian over $\mathbb{F}_{p^2}$, isomorphic to a supersingular PPAV. Fixed an isogeny $\phi : J_C \to J_{C'}$ with the maximal 2^T-isotropic subgroup G and allowed a precomputation in time $poly(\lambda, T)$, evaluate $\hat{\phi}(S)$ on a random divisor $S \in Y_2$ in parallel time less than T.

To set parameter sizes, we discuss several attacks on the high-genus isogeny shortcut problems, similar to the attacks mentioned in Burdges and De Feo (2021) and De Feo et al. (2019). The complexities of attacks are summarized in Table 1.

Pairing inversion The simplest attacks focus on the properties of Weil pairings. Namely, for given $P, \phi(P), S$, to compute $\hat{\phi}(S) \in J_C[\ell]$ is enough to obtain a divisor $S' \in J_C[\ell]$, more specifically, $S' \in X_2$, such that $e_\ell(P, S') = e'_\ell(\phi(P), S)$, i.e., to solve the *pairing inverse problem* $e_\ell(P, \cdot) = e'_\ell(\phi(P), S)$.

Due to the surjection of Weil pairings $e_\ell(P, \cdot)$, the equation

$$e_\ell(P, S') = e'_\ell(\phi(P), S) \tag{2}$$

is satisfied with probability $1/\ell$ for a random divisor $S' \in X_2$. Thus, a better strategy is to randomly sample a divisor $S_0 \in Y_2$, then find $m \in \mathbb{Z}/\ell\mathbb{Z}$ such that $e'_\ell(\phi(P), S) = e_\ell(P, S_0)^m$, then the divisor mS_0 is one legitimate output for verification. Therefore, the security of DLP impacts the hardness of the pairing inversion problem.

From Setup, the embedding degree is 2, indicating the best algorithm is the Number Field Sieve (NFS) for $\mathbb{F}_{p^2}$, whose (heuristic) complexity is $L_p(1/3)$. With the progress on NFS, the DLP in $\mathbb{F}_{p^2}$ for a prime of around 300 bits has been solved in Barbulescu et al. (2015), then Barbulescu and Duquesne (2019) have selected the parameters for pairings under several security levels. It is suggested to utilize prime p of around 1500 bits and ℓ of 256 bits for 128-bit security. Unfortunately, the pairing inversion problem is insecure under quantum computers.

Computing shortcuts One natural attack comes from finding a "simple" isogeny between J_C and $J_{C'}$, agreeing with ϕ on $J_C[\ell]$, but requiring less parallel time to evaluate.

However, considering supersingular PPAVs, there are no generic algorithms to compute (ℓ_0, ℓ_0)-isogenies for odd primes ℓ_0. When $\ell_0 = 3, 5$, explicit formulae of (ℓ_0, ℓ_0)-isogenies exist for some special cases (Bruin et al. 2014; Flynn 2015), but the formulae become more complicated when ℓ_0 is a large prime, including the algorithms introduced in Cosset and Robert (2015). Furthermore, the structures of the (ℓ_0, ℓ_0)-isogenous graphs are more complicated than those for elliptic curves, and we only know a few properties for $\ell_0 = 2$. Therefore, it is tough to utilize (ℓ_0, ℓ_0)-isogeny with odd primes ℓ_0 for finding an isogenous path between supersingular Jacobians. For Richelot isogenies, some properties are discussed in Florit and Smith (2022), and only the local neighborhoods in the (2, 2)-isogeny graph have been exploited in Florit and Smith (2022), thus the generic properties of global Richelot isogeny graphs are still hard mathematical problems.

To break our scheme, the attack needs to compute another isogenous map between supersingular hyperelliptic curves, i.e., isogenous to the product of g supersingular elliptic curves, thus a natural idea is to find another isogeny between two superspecial abelian varieties, as the more specifical PPAVs isogenous to g supersingular elliptic curves. In general, Costello and Smith (2020) demonstrated that for two superspecial abelian varieties A_1 and A_2, finding a path $\phi : A_1 \to A_2$ in the $(\ell_0, \ldots, \ell_0)$-isogeny graph requires $\tilde{O}(p^{g-1})$ field operations on a classical computer, and $\tilde{O}(\sqrt{p^{g-1}})$ field operations on a quantum computer. Therefore, computing a shortcut of known Richelot isogenies between two supersingular hyperelliptic curves requires more than $\tilde{O}(\sqrt{p^{g-1}})$ field operations for quantum computers, thus it is negligible to compute another path between two isogenous hyperelliptic curves of known isogenies.

Even if we have obtained a short isogeny $\psi : J_C \to J_{C'}$, we must find ω such that $\omega \circ \hat{\psi} = \hat{\phi}$ on $J_{C'}[\ell]$. More specifically, ω satisfies $\omega \circ \hat{\psi} \mid_{Y_2} = \hat{\phi} \mid_{Y_2}$. Yet the structure of $J_C[\ell]$ is clear, it is inefficient to determine map ω with the property of endomorphism. To simplify this problem, we restrict our attention to the subgroup X_2. If $\hat{\psi}(Y_2) = X_2$,

Table 2 Comparison of VDFs. For simplity, all times are assumed to be bounded by a constant factor, where T and λ are the delay parameter and security parameter, respectively

VDF	Setup	Eval	Verify	Proof size
Wesolowski's VDF (Wesolowski 2020)	λ^3	$(1+\frac{2}{\log T})T$	λ^4	λ^3
Pietrzak's VDF (Pietrzak 2019)	λ^3	$(1+\frac{2}{\sqrt{T}})T$	$\log T$	λ^3
Leo's VDF (Loe et al. 2022)	λ^3	T	1	–
Isogeny VDF (De Feo et al. 2019)	$T\log\lambda$	T	λ^4	–
This work	$T\log\lambda$	T	λ^4	–

ω is determined via computing discrete logarithms on X_2, while it hardly occurs. In general, we have $\hat{\psi}(Y_2) \neq X_2$, then ω induces a group isomorphism from $\hat{\psi}(Y_2)$ to X_2, where computing discrete logarithms on X_2 is also involved. As a result, the problem of searching ω is more complicated than pairing inversion problems.

Parallel isogeny evaluation Finally, another obvious attack would utilize more parallel resources for evaluating chains of isogenies, whose aim is to accelerate evaluations of the sequential and slow function Eval. However, Richelot isogenies utilize unique maximal 2-Weil isotropic subgroups for the next isogeny, requiring a maximal 2-Weil isotropic subgroup in Jacobians, so all existing algorithms go through all T intermediate PPAVs. In addition, for chains of 2-isogenies, replacing two 2-isogenies by one 4-isogeny is the generic technique for SIDH (De Feo et al. 2014), which will reduce the total cost by a constant factor, while there is no algorithm to evaluate $(2^n, 2^n)$-isogenies directly.

Consequently, the implementations of Richelot isogenies must be in a straight line, similar to the iterative isogenies for isogeny-based VDFs. Hence, an adversary can not accelerate the computation even using $poly(T)$ processors at present.

Other known attacks In Kunzweiler et al. (2021), an adaptive attack has been proposed, where the gist is the symplectic basis related to Weil pairings. In general, finding the symplectic basis is equivalent to solving DLP for Weil pairings, which is practical for smooth order ℓ_0^n. Whereas, our scheme leverages the divisors of large prime order, then determining the symplectic basis of order ℓ is at least as hard as the pairing inversion problems.

Recently, Castryck and Decru (2023) established an attack for SIDH, then Robert (2023) generalized this method to the PPAVs of all genera. However, this strategy employs all generators of torsion groups in two PPAVs and leverages the parameter tweaks for the smooth prime factorization, then the secret isogeny is recovered. In our VDF, the isogeny has been fixed with the output, i.e., we already have an isogeny path, and the ultimate goal is to evaluate the isogeny faster, so it may work for computing shortcuts. Luckily, this attack can not apply to VDFs straightforwardly. On the one hand, only two generators of $J_C[\ell]$ with the corresponding images in $J_{C'}$ are known, such that there are not enough divisors to apply this attack. On the other hand, the prime p is particularly selected such that $p+1$ has a large prime factor ℓ, so the parameter tweaks have few choices since the guess processions search for small isogenies upon the factorization of the difference value of divisors of 2^T and ℓf. Hence, for the huge difference between 2^T and ℓf, the first guessing isogeny is of large degree, almost identical to 2^T, which makes it almost impossible to find a legitimate isogeny since there is one unique choice among all likely isogenies, whose degree is close to 2^T. Consequently, this attack has no influence on our scheme. Similarly, yet two generators with the corresponding images are fixed, the isogeny-based VDF (De Feo et al. 2019) can resist the attacks for genus one SIDH (Castryck and Decru 2023; Maino et al. 2023; Robert 2023).

Parameters and comparison

Based on the above analysis, we choose a 256-bit prime ℓ, then set $T = 1244$ and $f = 63$ to obtain the prime $p = 2^{1244} \cdot 63\ell - 1$ of 1506 bits, as the same prime in De Feo et al. (2019) for the security level of 128 bits. It is believed that computing discrete logarithm in a subgroup of order ℓ in a finite field $\mathbb{F}_{p^2}$ requires more than 2^{128} operations.

As for the implementation, Weil pairings can be realized by pairing-based cryptography, where Miller's algorithm (Miller 2004) is suggested. Moreover, one can substitute Weil pairings by Tate pairings, then half of Miller loops are saved at the expense of one final exponentiation. However, Richelot isogenies is a relatively new topic, and we refer to the relevant algorithms in Castryck and Decru (2023), Castryck et al. (2020), Flynn and Ti (2019) and Kunzweiler (2022). In general, isogenies between high genus PPAVs are more inefficient than elliptic curve isogenies, which means that our VDFs may obtain larger delay effect under the same parameter set than isogeny-based VDF (De Feo et al. 2019).

The overall comparison of different VDF schemes is depicted in Table 2. Compared with other VDFs, e.g., Pietrzak (2019) and Wesolowski (2020), one notable advantage is the empty proof, where the pairings play the role of proof. Note that Leo's VDF (Loe et al. 2022) can achieve empty proof, but it is a prerequisite to generate a fresh Blum integer for every VDF, while the prime in our scheme can be applied for all schemes. Apart from that, our VDF is non-interactive such that the output can

$\mathsf{Setup}(\lambda, T)$

1. Choose primes ℓ, p according to the security parameter λ;
2. Select Jacobian J_C from a supersingular hyperelliptic curve $C/\mathbb{F}_{p^2}$;
3. Having fixed a kernel subgroup $G \subset J_C$, perform the non-backtracking $(2^T, 2^T)$-isogeny $\phi : J_C \to J_{C'}$ such that its dual is determined by $\hat{\phi}$;
4. Choose a generator $P \in J_C[\ell]$, and evaluate $\phi(P)$;
5. Output $(\mathsf{ek}, \mathsf{pk}) = \Big((J_{C'}, \hat{\phi}), (J_{C'}, P, \phi(P))\Big)$.

$\mathsf{Extract}(J_C, J_{C'}, \mathsf{id}, \hat{\phi})$

1. Compute $Q = \mathsf{H}_1(\mathsf{id})$;
2. Compute and output $\hat{\phi}(Q)$.

$\mathsf{Encaps}(J_C, J_{C'}, \mathsf{id}, P, \phi(P))$

1. Compute $Q = \mathsf{H}_1(\mathsf{id})$;
2. Randomly select $r \in \mathbb{Z}/\ell\mathbb{Z}$, and compute $s = e'_\ell(\phi(P), Q)^r$;
3. Output $(rP, \mathsf{H}_2(s))$.

$\mathsf{Decaps}(J_C, J_{C'}, rP, \hat{\phi}(Q))$

1. Compute $s = e_\ell(rP, \hat{\phi}(Q))$;
2. Output $\mathsf{H}_2(s)$.

Fig. 2 Delay encryption from hyperelliptic curves

be verified without interactions. Nevertheless, the computation of Richelot isogenies is inefficient, and the time needed for setup is almost the same as evaluation, which is also the drawback of isogeny-based VDFs. To this problem, a possible solution may come from Kummer varieties, where Kummer lines have been employed for acceleration (Chen et al. 2023), which needs more development in this realm. Moreover, the property of perfect soundness fails for our design, which is the shortage compared with isogeny-based VDFs (De Feo et al. 2019).

Delay encryptions from hyperelliptic curves

Since the original DE has been derived from isogeny-based VDFs (Burdges and De Feo 2021), we present new DE from hyperelliptic curves in a similar roadmap. The IBE scheme (Boneh and Franklin 2003) is modified for our construction, i.e., the master secret is substituted by a long chain of Richelot isogenies, so that the decryption key from a fixed identity is a slow operation.

Our design

Setup is almost identical to the VDFs in Fig. 1, where the prime $p = 2^T \ell f - 1$ and Jacobian J_C are fixed, then an isogeny $\phi : J_C \to J_{C'}$ with the image of an ℓ-torsion divisor is established. To depict other routines, we would introduce two hash functions. Let $\mathsf{H}_1 : \{0,1\}^\lambda \to J_{C'}[\ell]$ be used to hash *id* to divisors of order ℓ, and $\mathsf{H}_2 : \mathbb{F}_{q^k} \to \{0,1\}^\lambda$ be a key derivation. The detailed protocol is described in Fig. 2, where the notations coincide with those in Fig. 1.

Remark 5
If Weil pairing $e'_\ell(Q, \phi(P)) = 1$, Q is substituted by $\mathsf{H}_1(\mathsf{id}||Q)$, where the failure probability is $1/\ell$. Moreover, this strategy applies to all involved evaluations of H_1.

The correctness of our scheme is satisfied by the following equation

$$e_\ell(rP, \hat{\phi}(Q)) = e_\ell(P, \hat{\phi}(Q))^r = e'_\ell(\phi(P), Q)^r. \tag{3}$$

Remark 6
Note that two hash identities Q, Q' such that $e'_\ell(\phi(P), Q) = e'_\ell(\phi(P), Q')$ determine the same s for Encaps *and* Decaps*, then the adversary only compute the image of one divisor from them under $\hat{\phi}$. Whereas, from the analysis in the proof of Theorem 1, it occurs with probability $1/\ell$, which is still negligible.*

Security analysis

To illustrate the security of our scheme, we follow the line in Burdges and De Feo (2021). We first generalize the *bilinear isogeny shortcut games* to high genus as follows:

Precomputation The adversary receives $p, \ell, J_C, J_{C'}, \phi$ and outputs an algorithm $\mathcal{B}$.

Challenge The challenger outputs uniformly random $P_0 \in J_C[\ell]$ and $Q_0 \in J_{C'}[\ell]$.

Guess Algorithm $\mathcal{B}$ runs on the pair (P_0, Q_0). Then the adversary wins if $\mathcal{B}$ outputs

$$\mathcal{B}(P_0, Q_0) = e_\ell(P_0, \hat{\phi}(Q_0)) = e'_\ell(\phi(P_0), Q_0).$$

Similarly, we say the *high-genus bilinear isogeny shortcut game* is Δ-hard if no adversary running the precomputation in time $poly(\lambda, T)$ produces an algorithm $\mathcal{B}$ that wins in time less than Δ with non-negligible probability. Consequently, the following theorem illustrates that our scheme satisfies Δ-IND-CPA if the above game is hard.

Theorem 2 *The delay encryption scheme from hyperelliptic curves is Δ-IND-CPA secure, assuming the Δ'-hardness of the high-genus bilinear isogeny shortcut game with $\Delta \in \Delta' - o(\Delta')$, where H_1 and H_2 are assumed as the random oracles.*

Proof

The proof of this theorem is almost identical to that of delay encryption from elliptic curves in Burdges and De Feo (2021), so we omit the proof for brevity. □

For the security of the high-genus bilinear isogeny shortcut game, it follows the analysis of three attacks in "Security analysis" section. Consequently, the parameters are the same as those for our VDFs from hyperelliptic curves with analogous merits and demerits.

Conclusion

In this work, we present the first VDF and DE from hyperelliptic curves by utilizing Richelot isogenies and non-degenerate pairings, which broaden the cryptographic applications of high-genus isogenies. In particular, we employ the framework in isogeny-based VDF for two schemes with analogous merits and demerits as those from isogeny-based ones, while our scheme is secure under generalized assumptions on high-genus isogenies. To further implement those schemes, the study on efficient Richelot isogenies would be welcome, which is the main obstacle to the implementation of isogeny-based cryptography from hyperelliptic curves. Apart from that, the cryptographic applications from high-genus isogenies may share the smallest key sizes among post-quantum cryptosystems, thus it is natural to enrich this cryptosystem with abundant cryptographic constructions.

Acknowledgements
Not applicable.

Author contributions
CC is completed the main work of the paper and drafted the manuscript. FZ is participated in problem discussions and improvements of the manuscript. All authors read and approved the final manuscript.

Funding
This work is supported by the National Natural Science Foundation of China (No. 62272491) and the Guangdong Major Project of Basic and Applied Basic Research (2019B030302008) and the National R &D Key Program of China under Grant (2022YFB2701500).

Availability of data and materials
Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

Declarations

Competing interests
The authors declare that there is no conflict of interest regarding the publication of this article.

Received: 7 June 2023 Accepted: 4 September 2023
Published online: 08 November 2023

References

Armknecht F, Barman L, Bohli J, Karame GO (2016) Mirror: enabling proofs of data replication and retrievability in the cloud. In: Holz T, Savage S (eds) 25th USENIX security symposium. USENIX Association, Berkeley, pp 1051–1068
Barbulescu R, Duquesne S (2019) Updating key size estimations for pairings. J Cryptol 32(4):1298–1336
Barbulescu R, Gaudry P, Guillevic A, Morain F (2015) Improving NFS for the discrete logarithm problem in non-prime finite fields. In: Oswald E, Fischlin M (eds) EUROCRYPT 2015, LNCS, vol 9056. Springer, pp 129–155
Boneh D, Franklin MK (2003) Identity-based encryption from the Weil pairing. SIAM J Comput 32(3):586–615
Boneh D, Lynn B, Shacham H (2001) Short signatures from the Weil pairing. In: Boyd C (ed) ASIACRYPT 2001, LNCS, vol 2248. Springer, New York, pp 514–532
Boneh D, Bonneau J, Bünz B, Fisch B (2018) Verifiable delay functions. In: Shacham H, Boldyreva A (eds) CRYPTO 2018, LNCS, vol 10991. Springer, pp 757–788
Bruin N, Doerksen K (2011) The arithmetic of genus two curves with (4, 4)-split Jacobians. Can J Math 63(5):992–1024
Bruin N, Flynn EV, Testa D (2014) Descent via (3, 3)-isogeny on Jacobians of genus 2 curves. Acta Arith 165(3):201–223
Burdges J, De Feo L (2021) Delay encryption. In: Canteaut A, Standaert F (eds) EUROCRYPT 2021, LNCS, vol 12696. Springer, New York, pp 302–326
Cassels JWS, Flynn EV (1996) Prolegomena to a middlebrow arithmetic of curves of genus 2, London Mathematical Society lecture note series, vol 230. Cambridge University Press, Cambridge
Castryck W, Decru T (2023) An efficient key recovery attack on SIDH. In: Hazay C, Stam M (eds) EUROCRYPT 2023, LNCS, vol 14008. Springer, New York, pp 423–447
Castryck W, Decru T, Smith B (2020) Hash functions from superspecial genus-2 curves using Richelot isogenies. J Math Cryptol 14(1):268–292
Chávez-Saab J, Rodríguez-Henríquez F, Tibouchi M (2021) Verifiable isogeny walks: towards an isogeny-based postquantum VDF. In: AlTawy R, Hülsing A (eds) SAC 2021, LNCS, vol 13203. Springer, New York, pp 441–460

Chen C, Zhang F, Zhao C (2023) Isogeny computation on Kummer lines and applications. J Inf Secur Appl 76(103):546
Cohen B, Pietrzak K (2018) Simple proofs of sequential work. In: Nielsen JB, Rijmen V (eds) EUROCRYPT 2018, LNCS, vol 10821. Springer, New York, pp 451–467
Cohen H, Frey G, Avanzi R, Doche C, Lange T, Nguyen K, Vercauteren F (eds) (2005) Handbook of elliptic and hyperelliptic curve cryptography. Chapman and Hall/CRC, Boca Raton
Cosset R, Robert D (2015) Computing (ℓ,ℓ)-isogenies in polynomial time on Jacobians of genus 2 curves. Math Comput 84(294):1953–1975
Costello C, Smith B (2020) The supersingular isogeny problem in genus 2 and beyond. In: Ding J, Tillich J (eds) PQCrypto 2020, LNCS, vol 12100. Springer, New York, pp 151–168
De Feo L, Jao D, Plût J (2014) Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. J Math Cryptol 8(3):209–247
De Feo L, Masson S, Petit C, Sanso A (2019) Verifiable delay functions from supersingular isogenies and pairings. In: Galbraith SD, Moriai S (eds) ASIACRYPT 2019, LNCS, vol 11921. Springer, New York, pp 248–277
Döttling N, Garg S, Malavolta G, Vasudevan PN (2020) Tight verifiable delay functions. In: Galdi C, Kolesnikov V (eds) SCN 2020, LNCS, vol 12238. Springer, New York, pp 65–84
Dwork C, Naor M (1992) Pricing via processing or combatting junk mail. In: Brickell EF (ed) CRYPTO 92, LNCS, vol 740. Springer, New York, pp 139–147
Ephraim N, Freitag C, Komargodski I, Pass R (2020) Continuous verifiable delay functions. In: Canteaut A, Ishai Y (eds) EUROCRYPT 2020, LNCS, vol 12107. Springer, New York, pp 125–154
Fiat A, Shamir A (1986) How to prove yourself: practical solutions to identification and signature problems. In: Odlyzko AM (ed) CRYPTO 1986, LNCS, vol 263. Springer, New York, pp 186–194
Florit E, Smith B (2022) An atlas of the Richelot isogeny graph. In: Theory and applications of supersingular curves and supersingular abelian varieties, RIMS Kôkyûroku Bessatsu, B90. Research Institute for Mathematical Sciences (RIMS), Kyoto, pp 195–219
Florit E, Smith B (2022) Automorphisms and isogeny graphs of abelian varieties, with applications to the superspecial Richelot isogeny graph. In: Arithmetic, geometry, cryptography, and coding theory 2021, contemporary mathematics. American Mathematical Society, Providence, vol 779, pp 103–132
Flynn EV (2015) Descent via $(5,5)$-isogeny on Jacobians of genus 2 curves. J Number Theory 153:270–282
Flynn EV, Ti YB (2019) Genus two isogeny cryptography. In: Ding J, Steinwandt R (eds) PQCrypto 2019, LNCS, vol 11505. Springer, New York, pp 286–306
Galbraith SD, Hess F, Vercauteren F (2007) Hyperelliptic pairings. In: Takagi T, Okamoto E, Okamoto T, Okamoto T (eds) Pairing 2007, LNCS, vol 4575. Springer, New York, pp 108–131
Juels A, Kaliski BS Jr (2007) PORs: proofs of retrievability for large files. In: Ning P, di Vimercati SDC, Syverson PF (eds) CCS 2007. ACM, Bhageerath, pp 584–597
Kiayias A, Russell A, David B, Oliynykov R (2017) Ouroboros: a provably secure proof-of-stake blockchain protocol. In: Katz J, Shacham H (eds) CRYPTO 2017, LNCS, vol 10401. Springer, New York, pp 357–388
Kunzweiler S (2022) Efficient computation of $(2^n,2^n)$-isogenies. IACR Cryptol ePrint Arch, p 990. https://eprint.iacr.org/2022/990
Kunzweiler S, Ti YB, Weitkämper C (2021) Secret keys in genus-2 SIDH. In: AlTawy R, Hülsing A (eds) SAC 2021, LNCS, vol 13203. Springer, New York, pp 483–507
Lenstra AK, Wesolowski B (2017) Trustworthy public randomness with sloth, unicorn, and trx. Int J Appl Cryptogr 3(4):330–343
Loe AF, Medley L, O'Connell C, Quaglia EA (2022) TIDE: a novel approach to constructing timed-release encryption. In: Nguyen K, Yang G, Guo F, Susilo W (eds) ACISP 2022, LNCS, vol 13494. Springer, New York, pp 244–264
Lubicz D, Robert D (2012) Computing isogenies between abelian varieties. Compos Math 148(5):1483–1515
Maino L, Martindale C, Panny L, Pope G, Wesolowski B (2023) A direct key recovery attack on SIDH. In: Hazay C, Stam M (eds) EUROCRYPT 2023, LNCS, vol 14008. Springer, New York, pp 448–471
Miller VS (2004) The Weil pairing, and its efficient calculation. J Cryptol 17(4):235–261
Mumford D (1970) Abelian varieties, Tata Institute of Fundamental Research Studies in Mathematics. Oxford University Press, London, Bombay, Published for the Tata Institute of Fundamental Research, vol 5
Pietrzak K (2019) Simple verifiable delay functions. In: Blum A (ed) ITCS 2019, LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, vol 124, pp 60:1–60:15
Rabin MO (1983) Transaction protection by beacons. J Comput Syst Sci 27(2):256–267
Rivest RL, Shamir A, Wagner DA (1996) Time-lock puzzles and timed-release crypto. Technical report, USA
Robert D (2023) Breaking SIDH in polynomial time. In: Hazay C, Stam M (eds) EUROCRYPT 2023, LNCS, vol 14008. Springer, New York, pp 472–503
Smith B (2006) Explicit endomorphisms and correspondences. Bulletin of the Australian Mathematical Society
Valiant P (2008) Incrementally verifiable computation or proofs of knowledge imply time/space efficiency. In: Canetti R (ed) TCC 2008, LNCS, vol 4948. Springer, New York, pp 1–18
Vélu J (1971) Isogénies entre courbes elliptiques. C. C. R. Acad. Sci. Paris Sér. A–B 273:A238–A241
Wesolowski B (2020) Efficient verifiable delay functions. J Cryptol 33(4):2113–2147

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen journal and benefit from:

- ► Convenient online submission
- ► Rigorous peer review
- ► Open access: articles freely available online
- ► High visibility within the field
- ► Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com