A New Class of Trapdoor Verifiable Delay Functions

Ahmed Zawia(✉) and M. Anwar Hasan

University of Waterloo, Waterloo, ON, Canada
{azawia,ahasan}@uwaterloo.ca

Abstract. A verifiable delay function (VDF) is a function whose evaluation involves lengthy sequential operations, yet its outcome is publicly verifiable. As an extension, a trapdoor-VDF is a VDF with a shortcut that speeds up the evaluation process. This paper presents a new class of trapdoor-VDFs featuring a large ensemble of trapdoors for each instantiation of the function. This way, a client can randomly choose a private trapdoor from the ensemble, thereby using it to encapsulate a secret to the future as a unique puzzle. To solve the puzzle, the server, which does not know the trapdoor, requires a prescribed number of sequential steps to evaluate the function. Any client can efficiently verify the correctness of the server's evaluation with zero knowledge of the trapdoor being used. We present an approach for constructing the proposed class of trapdoor-VDFs based on bilinear pairings and a long walk on supersingular isogeny graphs. Finally, we examine the security of our construction under trapdoor-VDF security notions.

Keywords: Delay primitives · verifiable delay functions · delay encryption · time-lock puzzle

1 Introduction

This work examines a remedy for the vulnerability that arises from knowing or predicting a protocol's outcome. The vulnerability stems from malicious participants influencing the outcome or gaining an advantage by knowing the outcome beforehand. One way to solve this problem is to impose a prescribed number of sequential steps to obtain the desired outcome. This solution has been introduced in the previous works such as time-lock puzzle (TLP) [29], proofs of sequential work (PoSW) [1,15,22,26], and verifiable delay functions (VDFs) [6]. These primitives are all time-sensitive in that they only release the outcome after a prescribed delay (T). The outcome of VDF and PoSW is publicly verifiable, while that of TLP requires a secret. The public verification of the outcome's *uniqueness* is more efficient in VDF than in PoSW since the latter requires all T steps. The uniqueness property ensures that there are no multiple valid proofs for different outcome.

This work focuses more on the properties, specifically uniqueness and public verifiability, that make VDF stand out from others. Furthermore, we are interested in a VDF-like primitive that allows any participant (other than the trusted

G.-V. Jourdan et al. (Eds.): FPS 2022, LNCS 13877, pp. 71–87, 2023.
https://doi.org/10.1007/978-3-031-30122-3_5

setup) to predict the outcome in advance by knowing a secret trapdoor. In the absence of the secret trapdoor, participants obtain the desired outcome through a prescribed number of sequential evaluations. It is, however, possible to verify the result efficiently and publicly. We refer to primitives with such characteristics by trapdoor-VDF. Considering its importance, trapdoor-VDF is suitable for time-sensitive applications that require both public verification and timely release assurances. There are many possible applications, such as sealed-voting, delayed decapsulation [9,25], and front-running attack prevention [14].

Related Work. Since the work of Boneh et al. [6], several VDF constructions have been proposed based on Rivest, Shamir, and Wagner's time-lock assumption [29], including [5,17,24,28,30], and [34], which use different verification techniques, provide additional properties and offer enhancement. Several other constructions [12,18,31] are based on the difficulty of shortening the evaluation of isogeny with a large degree, which was first introduced by [18]. The verification proof of [18] is based on the bilinear pairing of the Boneh-Lynn-Shacham (BLS) signature scheme [8]. Similar to time-lock functions, Shani's [31] proof requires releasing a secret shortcut to the puzzle to recompute the puzzle's answer. In [12], Chavez-Saab et al. propose an inefficient verification method based upon succinct non-interactive arguments (SNARGs).

Later, a delay encryption scheme [9] was developed using Boneh and Franklin's identity based encryption (IBE) scheme [7] in conjunction with Feo et al.'s delay function [18]. The scheme in [9] can operate in batch mode so that the function can be evaluated once to perform many decryptions. However, the method requires a trusted, unpredictable seed, along with a considerable amount of storage to perform computations (e.g., 12 TB for 1 h delay [9]). Furthermore, their approach does not employ a trapdoor mechanism to predict the answer in advance. Hence, trapdoor-VDF is more comparable to TLP even though the former offers efficient public verification without revealing any secrets (i.e., the secret trapdoor). We note that recent work has extended the security of TLPs and discusses the notion of its public verifiability, some of which are [2,3,13,20].

In addition, the term "trapdoor-VDF" has been used previously in [34], though their approach differs from that presented in this paper. In [34], each participant constructs independent instances of trapdoor-VDF. Every instance has a secret trapdoor that shortens the long evaluation process. Therefore, an instance generator can answer any challenge faster as it has the secret trapdoor. Our work presents a new class of trapdoor-VDFs, where each instantiation includes a description of a finite trapdoor ensemble. Hence, any participant can generate a unique challenge, together with its answer, using a hidden trapdoor sampled at random from the ensemble.

Contributions. In this paper, we first formalize the security notions of trapdoor-VDF. We then introduce a novel approach to trapdoor-VDF; the proposed trapdoor-VDF is a VDF with a large ensemble of distinct efficient shortcuts called trapdoors. The delay function involves sequentially evaluating a large smooth degree supersingular isogeny, which was first proposed by Feo et al. [18]. The trapdoor ensemble is a set of isogenies of a smaller degree defined over $\mathbb{F}_p$, where p is a prime. In public verification, a bilinear pairing equality serves

as proof of the correct evaluation of a secret trapdoor (i.e., a randomly sampled isogeny). Finally, we show that our proposal is secure under trapdoor-VDF security notions.

2 Preliminaries

General Notations. If n is a positive integer, the set $\{1, \ldots, n\}$ is denoted by $[n]$. In general, a finite set is denoted by calligraphic font (e.g., $\mathcal{S}$). The cardinality of a set $\mathcal{S}$ is denoted by $|\mathcal{S}|$. Let $e \leftarrow_R \mathcal{S}$ denote the process of uniformly sampling a random element e from $\mathcal{S}$. The deterministic selection of e from $\mathcal{S}$ is denoted by $e \leftarrow \mathcal{S}$. If Exe is an algorithm, $a \leftarrow_R \mathsf{Exe}$ denotes running Exe on fresh random coins and assigning the output to a. The deterministic execution of Exe, on the other hand, is denoted by $a \leftarrow \mathsf{Exe}$. The notation $\mathsf{Pr}[Evnt : P_1, P_2, \ldots, P_n]$ is used to represent the probability of an event $Evnt$ occurring after the ordered processes $P_1, P_2, \ldots, P_n$. We denote the composition of two functions by $\circ$ such that $\mathsf{f} \circ \mathsf{g}(x) = \mathsf{f}(\mathsf{g}(x))$ for some input x. Let $\boldsymbol{m} \leftarrow_R \mathcal{S}^n$ be the vector $(m_i)_{i \in [n]}$ of size n such that $m_i \leftarrow_R \mathcal{S}$ for all $i \in [n]$. The set of all odd prime numbers that are less than or equal to k is referred to as $\mathsf{Primes}(k)$. For an integer a and an odd prime b, the Legendre symbol is denoted by $(\frac{a}{b})$.

Supersingular Elliptic Curve. Throughout this work, we consider a curve $E/\mathbb{F}_p$ to be a supersingular elliptic curve defined over a prime field $\mathbb{F}_p$ with a large prime p. A point P on $E/\mathbb{F}_p$ is the pair $(x, y) \in \mathbb{F}_p \times \mathbb{F}_p$. The set of $\mathbb{F}_p$-rational points on $E/\mathbb{F}_p$ is denoted as $E(\mathbb{F}_p)$ and the set size as $|E(\mathbb{F}_p)|$. Let ∞_E be the point at infinity on $E/\mathbb{F}_p$ and $\infty_E \in E(\mathbb{F}_p)$. The field $\overline{\mathbb{F}}_p$ is the algebraic closure of $\mathbb{F}_p$. The subgroup of points of order N is called the *N-torsion points* which is defined as $E[N] = \{P \in E(\overline{\mathbb{F}}_p) : [N]P = \infty_E\}$.

Definition 1 ([32]). *An elliptic curve $E/\mathbb{F}_p$ is supersingular if the following equivalent properties are true*

- *There is no $P \in E(\overline{\mathbb{F}}_p)$ with order p (i.e., $E[p] = \{\infty_E\}$).*
- *$|E(\mathbb{F}_p)| = p + 1 - t$ and $p \nmid t$ (i.e., $\gcd(p, t) \neq 1$).*
- *The endomorphism ring of $E/\overline{\mathbb{F}}_p$ is an order in a quaternion algebra.*

Otherwise, $E/\mathbb{F}_p$ is said to be an ordinary curve.

Isogenous Curves. An isogeny between curves $(\phi : E_1 \rightarrow E_2)$ is a surjective morphism that has a finite kernel such that $\phi(\infty_{E_1}) = \infty_{E_2}$. We say ϕ is defined over $\mathbb{F}_p$, if the non-constant rational map representing ϕ has coefficients in $\mathbb{F}_p$. Let $\langle S \rangle$ be a cyclic subgroup of $E(\overline{\mathbb{F}}_q)$ generated by S. In this work, we compute an isogeny ϕ with kernel $\langle S \rangle$ using Vélu's formulas [33]. Also, we will focus on separable isogenies where the isogeny degree is its kernel size (i.e., $\deg \phi = |\mathsf{ker}(\phi)|$). An isogeny of degree l is denoted by l-isogeny. Furthermore, an isogeny ϕ has a unique *dual* isogeny $\hat{\phi}$ with the same degree ($\deg \phi = \deg \hat{\phi}$) such that $\hat{\phi} : E_2 \rightarrow E_1$, and $\phi \circ \hat{\phi} = [\deg \phi]$ on E_2, where $[m]$ is multiplication-by m mapping (see [32, Theorem 6.1]). The non-backtracking walk is a sequence of isogenies that is not cyclic or followed by any dual isogeny(s).

The set of all group homomorphisms (i.e., isogenies) from E to itself is called the endomorphism ring of E ($\mathbf{End}(E)$). The endomorphism ring defined over $\mathbb{F}_p$ is denoted as $\mathbf{End}_{\mathbb{F}_p}(E)$. An isogeny is called *horizontal* isogeny if $\mathbf{End}_{\mathbb{F}_p}(E_1) \cong \mathbf{End}_{\mathbb{F}_p}(E_2) \cong \mathcal{O}$ [21] where $\mathcal{O}$ is an order of an imaginary quadratic field. Furthermore, we donate the set of supersingular elliptic curves defined over $\mathbb{F}_p$ with $\mathcal{O}$ by $\mathcal{E}_{\mathbb{F}_p}(\mathcal{O})$. By definition, curves in $\mathcal{E}_{\mathbb{F}_p}(\mathcal{O})$ are connected by horizontal isogenies.

A Supersingular Isogeny Graph Over $\mathbb{F}_p$. The structure of a supersingular isogeny graph over $\mathbb{F}_p$ is studied by Delfs and Galbraith, which is described in [16, Theorem 2.7]. Let $\mathcal{L}$ be a set of distinct primes such that $p \notin \mathcal{L}$, $(\frac{-p}{l_i}) = 1$ for all $l_i \in \mathcal{L}$. For $p > 3$, the graph $\mathcal{G}(\mathbb{F}_p, l_i)$ is a directed supersingular isogeny graph where the vertices are a $\mathbb{F}_p$-isomorphism classes of supersingular elliptic curves represented by j-invariants with an extra information to classify them into their $\mathbb{F}_p$-isomorphism class (i.e., to differentiate between elliptic curve twists). The graph edges are equivalence classes of $\mathbb{F}_p$-rational isogenies of a degree l_i. In our work, we employ a graph that represents the union of all $\mathcal{G}(\mathbb{F}_p, l_i)$, $\forall l_i \in \mathcal{L}$.

Isogeny and Pairing. Let N be a large prime such that $N \neq p$ and $N||E_1(\mathbb{F}_p)|$. Let μ_N be the group of Nth roots of unity in $\mathbb{F}^*_{p^u}$, where u is the smallest integer such that $N|p^u - 1$. The Weil pairing is the map $\hat{\mathsf{e}}_N^E : E[N] \times E[N] \to \mu_N$ that satisfies several properties. In particular, the Weil pairing has the property of being compatible with isogenies (see [32, Proposition 8.2]); and it is trivial to show that

$$\hat{\mathsf{e}}_N^{E_1}(P, [\deg \phi]Q) = \hat{\mathsf{e}}_N^{E_2}(\phi(P), \phi(Q)), \tag{1}$$

where $P \in E_1[N]$, $Q = \hat{\phi}(Q')$ for $Q' \in E_2[N]$, and $\phi \circ \hat{\phi} = [\deg \phi]$ on E_2.

3 Proposed Trapdoor-VDFs

By $\lambda \in \mathbb{N}$, we indicate the security level of a scheme. A function difficulty is denoted by T, which quantifies the amount of sequential work/steps necessary to produce/compute its output against any random input and with a polynomially large number of parallel processes. A function with a small T is identified as a short function, whereas one with a large T is called a long sequential function. Generally, we will denote the long sequential function by EVAL, with T being super-polynomial in λ.

An Informal Exposition. The proposed trapdoor-VDF is a VDF with a large ensemble of distinct shortcuts denoted by $\mathcal{F}$, called trapdoors. We say two trapdoors are equivalent if they produce the same output for the same inputs. The set of equivalent trapdoors is called a class. The class difficulty is the shortest trapdoor difficulty. Hence, the ensemble $\mathcal{F}$ is a collection of trapdoor classes.

Trapdoor-VDF Setup Agreement. Participants of a setup protocol agree on a long function EVAL with a difficulty T, a large ensemble of trapdoors $\mathcal{F}$, and possibly additional parameters for a security level λ.

Generation of a Challenge. In trapdoor-VDF, a Challenger can select a secret trapdoor tr_{sk} indexed (identified) in $\mathcal{F}$ by a random secret (sk). Using the trapdoor, the challenger can efficiently generate a challenge ($\mathbf{c}$) and its unique answer ($\mathbf{a}$). With the same parameters, distinct trapdoors produce different challenges (and accordingly different answers).

Obtaining the Answer. In the absence of sk, a Solver can only evaluate EVAL function in time no less than T to output the answer to $\mathbf{c}$ with a proof Π. Akin to VDF, the Solver gains no advantage from parallel computation. The Challenger can, however, get the answer $\mathbf{a}$ via the trapdoor tr_{sk} (i.e., in time less than T).

The Public Verification. The answer $\mathbf{a}$ is publically verifiable that is also in a zero knowledge of the secret sk. In a timeframe less than T, a **perfect** trapdoor-VDF is one in which the Challenger, who owns $\mathbf{c}$ and knows $\mathbf{a}$, cannot pass the public verification protocol. This is because the required proof cannot be fully computed before the specific time T.

A Formal Definition. We present our formal definition of trapdoor-VDF, which naturally overlaps with [6] and [34]'s VDF definition.

Definition 2. *Let $\mathcal{C}$, $\mathcal{S}$, $\mathcal{Y}$ be the challenge, secret, answer spaces, respectively. Our trapdoor-VDF is a tuple of algorithms (Setup, Challenger, Solver, Verify) defined below*

- *Setup: a randomized algorithm (runs in time $\mathsf{Poly}(\lambda)$) that takes a security parameter λ and a difficulty T and outputs public parameter pk.*
- *Challenger: a randomized algorithm (runs in time $\mathsf{Poly}(\log T, \lambda)$) that takes pk and selects a secret trapdoor tr_{sk} from the trapdoor ensemble $\mathcal{F}$ using a random secret sk (i.e., $\mathsf{tr}_{sk} \leftarrow \mathcal{F}$ given $sk \leftarrow_R \mathcal{S}$); then, it generates a challenge $\boldsymbol{c} \in \mathcal{C}$.*
- *Solver: an algorithm that takes pk and a challenge $\boldsymbol{c} \in \mathcal{C}$ and outputs the answer $\boldsymbol{a} \in \mathcal{Y}$ and a "possibly empty" proof Π. This algorithm must at least run in time T with $\mathsf{Poly}(\lambda)$ parallel processors.*
- *Verify: is a deterministic algorithm (runs in total time polynomial in $\log T$ and λ) takes a challenge $\boldsymbol{c}$, an answer $\boldsymbol{a}$, a proof Π, and pk; the algorithm outputs* ACCEPT *if $\boldsymbol{a}$ is indeed the corresponding answer to $\boldsymbol{c}$ under a given Π, otherwise* REJECT.

Trapdoor-VDF Properties. The following assumes that all statements are true for any λ, T and $pk \leftarrow_R \mathsf{Setup}(1^\lambda, T)$. A trapdoor-VDF construction is well-defined if it is correct, unique, and efficient.

- *Correctness*: A trapdoor-VDF is correct only if the Verifier accepts, with probability one, an honest Solver's answer $\mathbf{a}$ for any honest challenge $\mathbf{c} \leftarrow_R \mathsf{Challenger}(pk)$.
- *Uniqueness*: A trapdoor-VDF is unique only if there is only one valid answer $\mathbf{a}$, accepted by Verify, to every challenge $\mathbf{c} \leftarrow_R \mathsf{Challenger}(pk)$ with a secret sk.
- *Efficiency*: A trapdoor-VDF is efficient if the Verify algorithm runs in a time $\mathsf{Poly}(\log T, \lambda)$ that is significantly faster than the Solver algorithm, which has a

total running time polynomial in T and λ. Further, trapdoor-VDF must retain efficiency for any public parameters generated by Setup, runs in $\mathsf{Poly}(\log T, \lambda)$, and any challenge generated by $\mathsf{Challenger}$, which runs in time $\mathsf{Poly}(\log T, \lambda)$.

Let $\mathcal{A}$ be a polynomially bounded adversary who has no knowledge of the secret sk. Let $\mathcal{A}_1$ be an algorithm that outputs pre-computation on pk. Let $\mathcal{A}_2$ be an online efficient evaluating algorithm that runs in parallel time with $\mathsf{Poly}(\lambda)$ processors and returns an answer $\mathbf{a}'$. Let $\mathcal{A}_3$ be an online forging algorithm that runs in time $\mathsf{Poly}(T, \lambda)$ and returns a malicious answer and proof $(\mathbf{a}' \neq \mathbf{a}, \Pi')$. To be secure, a well-defined trapdoor-VDF should satisfy two key properties: sequentiality and soundness.

- *Sequentiality*: A trapdoor-VDF is sequential only if there is no adversary $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2)$ that has an online attack $(\mathcal{A}_2)$ running in time less than T and has a probability of success

$$\mathsf{Pr}\left[\mathbf{a}' = \mathbf{a} : \begin{array}{l} pk \leftarrow_R \mathsf{Setup}(1^\lambda, T), \\ \mathsf{pc} \leftarrow \mathcal{A}_1(pk), \\ \mathbf{c} \leftarrow_R \mathsf{Challenger}(pk), \\ (\mathbf{a}', -) \leftarrow \mathcal{A}_2(pk, \mathbf{c}, \mathsf{pc}), \\ (\mathbf{a}, \Pi) \leftarrow \mathsf{Solver}(pk, \mathbf{c}). \end{array}\right]$$

  that is greater than a negligible function of λ.
- *Soundness*: A trapdoor-VDF is sound only if the Verifier rejects any proof Π' for any answer $\mathbf{a}'$ that is not an output from $\mathsf{Solver}(pk, \mathbf{c})$ on any $\mathbf{c} \leftarrow_R \mathsf{Challenger}(pk)$. The probability of success for the adversary $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_3)$ to output a proof Π' for an answer $(\mathbf{a}', -) \neq \mathsf{Solver}(pk, \mathbf{c})$ is

$$\mathsf{Pr}\left[\begin{array}{l} \text{ACCEPT} \leftarrow \mathsf{Verify}(pk, \mathbf{c}, \mathbf{a}', \Pi') \\ \text{and } \mathbf{a}' \neq \mathbf{a} \end{array} : \begin{array}{l} pk \leftarrow_R \mathsf{Setup}(1^\lambda, T), \\ \mathsf{pc} \leftarrow \mathcal{A}_1(pk), \\ \mathbf{c} \leftarrow_R \mathsf{Challenger}(pk), \\ (\mathbf{a}, \Pi) \leftarrow \mathsf{Solver}(pk, \mathbf{c}), \\ (\mathbf{a}', \Pi') \leftarrow \mathcal{A}_3(pk, \mathbf{c}, \mathsf{pc}). \end{array}\right]$$

  that is a negligible function of λ.

Additionally, a well-defined trapdoor-VDF may comprise further properties, and one that is most relevant to our work is given below.

- *Perfectness*: The knowledge of tr_{sk} and the pair $(\mathbf{c}, \mathbf{a})$ solely does not provide an advantage in passing the public verification protocol. Let $\hat{\mathcal{A}}$ be an algorithm implementing $\mathsf{Challenger}$, which outputs the pair $(\mathbf{c}, \mathbf{a})$ and an algorithm $\hat{\mathcal{A}}_2$. The probability of success for $\hat{\mathcal{A}}_2$ to output an acceptable proof Π', in time less than T, for any pair $(\mathbf{c}, \mathbf{a})$ is

$$\mathsf{Pr}\left[\begin{array}{l} \text{ACCEPT} \leftarrow \mathsf{Verify}(pk, \mathbf{c}, \mathbf{a}, \Pi') \\ \text{and ACCEPT} \leftarrow \mathsf{Verify}(pk, \mathbf{c}, \mathbf{a}, \Pi) \end{array} : \begin{array}{l} pk \leftarrow_R \mathsf{Setup}(1^\lambda, T), \\ (\mathbf{c}, \mathbf{a}, \hat{\mathcal{A}}_2) \leftarrow \hat{\mathcal{A}}(\mathsf{Challenger}, pk), \\ (-, \Pi') \leftarrow \hat{\mathcal{A}}_2(pk, \mathbf{c}, \mathbf{a}), \\ (\mathbf{a}, \Pi) \leftarrow \mathsf{Solver}(pk, \mathbf{c}). \end{array}\right]$$

that is a negligible function of λ.

Further on Trapdoor-VDF Properties. A secure well-defined trapdoor-VDF accounts also for the trapdoor properties and assumption(s). This is due to the fact that a secure well-defined Challenger requires a secure well-defined trapdoor. Formally, a trapdoor $\mathsf{tr}_{sk} \in \mathcal{F}$, associated with a domain $\mathcal{D}(\mathsf{tr}_{sk})$ and range $\mathcal{R}(\mathsf{tr}_{sk})$, is defined as follows

- tr_{sk}: a short function in $sk \leftarrow_R \mathcal{S}$, with evaluation time $\mathsf{Poly}(\log T, \lambda)$, that takes an input pk; the function evaluation returns a challenge and answer pair $(\mathbf{c}, \mathbf{a}) \in \mathcal{C} \times \mathcal{Y}$.

For any random secret $sk \leftarrow_R \mathcal{S}$ and $\mathsf{tr}_{sk} \leftarrow \mathcal{F}$, the trapdoors in trapdoor-VDF feature several properties.

- *Challenger correctness:* Let $(\mathbf{c}, \mathbf{a})$ be a challenge and its answer pair generated by tr_{sk}. The correctness property requires that the answer $(\mathbf{a}', -) \leftarrow \mathsf{Solver}(pk, \mathbf{c})$ be equal to $\mathbf{a}$ with probability one (i.e., we must have $\mathbf{a}' = \mathbf{a}$ with probability one). There is, however, an extension to the previous statement. For instance, we can allow $\mathbf{a}' \neq \mathbf{a}$ only if there is a one-way public function (f) such that $\mathbf{a} \leftarrow \mathsf{f}(\mathbf{a}')$ and $\text{ACCEPT} \leftarrow \mathsf{Verify}(pk, \mathbf{c}, \mathsf{f}(\mathbf{a}'), \Pi)$ is *true* for all $\mathbf{c} \in \mathcal{C}$ and all valid proofs Π. Having such an extension allows us to construct a **prefect** trapdoor-VDF in which the Challenger, who owns $\mathbf{c}$ and knows $\mathbf{a}$, cannot pass the public verification protocol before time T.
- *Challenger efficiency:* All trapdoors in a trapdoor-VDF must also be efficient. For any λ, T and $pk \leftarrow_R \mathsf{Setup}(1^\lambda, T)$, the efficiency implies that (as in [4]):
  - There is an algorithm that runs in time $\mathsf{Poly}(\log T, \lambda)$ and implements the process of sampling tr_{sk} from $\mathcal{F}$ for all $sk \in \mathcal{S}$.
  - There is an algorithm that runs in time $\mathsf{Poly}(\log T, \lambda)$ and implements the process of sampling an element from $\mathcal{D}(\mathsf{tr}_{sk})$ (and/or $\mathcal{R}(\mathsf{tr}_{sk})$).
  - There is an algorithm that evaluates tr_{sk} in time $\mathsf{Poly}(\log T, \lambda)$ for any element of $\mathcal{D}(\mathsf{tr}_{sk})$.
- *Challenger security:* To be secure, the following problems must be hard for any λ, T and $pk \leftarrow_R \mathsf{Setup}(1^\lambda, T)$.
  - Given any challenge $\mathbf{c} \in \mathcal{C}$ generated under the secret $\mathsf{tr}_{sk} \in \mathcal{F}$, find the challenge's answer $\mathbf{a} \in \mathcal{Y}$ in time less than T.
  - Given any challenge and answer pair $(\mathbf{c}, \mathbf{a}) \in \mathcal{C} \times \mathcal{Y}$ generated by $\mathsf{tr}_{sk} \in \mathcal{F}$, find $sk \in \mathcal{S}$.
  - Let $(\mathbf{c}', \mathbf{a}') \in \mathcal{C} \times \mathcal{Y}$ be any challenge and answer pair generated by $\mathsf{tr}_{sk'} \in \mathcal{F}$. Given a challenge $\mathbf{c} \in \mathcal{C}$ under $\mathsf{tr}_{sk} \in \mathcal{F}$, find its answer $\mathbf{a} \in \mathcal{Y}$.

Similar to VDF, the difficulty T is restricted to subexponential in λ. It is therefore cheaper to perform the T-sequential evaluation than to compromise the trapdoor-VDF security.

4 Design Rationale

This section discusses our approach to construct a proof-of-concept instance of trapdoor-VDF. To construct trapdoor-VDF, we begin by defining a long sequential public function $\mathsf{EVAL} : \mathcal{X} \rightarrow \mathcal{X}$ with a public challenge and answer (i.e., $x, y \in \mathcal{X}$ such that $y \leftarrow \mathsf{EVAL}(x)$). The secret is a random string ($sk \leftarrow_R \mathcal{S}$) that indexes a secret short map ($\mathsf{tr}_{sk} : \mathcal{X} \rightarrow \mathcal{X}$) in the ensemble $\mathcal{F}$. Using a random secret map, one may craft a trapdoor. A new challenge (x') can be obtained by masking x with tr_{sk} (i.e., $x' \leftarrow \mathsf{tr}_{sk}(x)$). The secret map tr_{sk}, which can also determine $y' \leftarrow \mathsf{tr}_{sk}(y)$, becomes the trapdoor. This statement is true assuming that the action $\mathsf{tr}_{sk} \circ \mathsf{EVAL}$ is equivalent to $\mathsf{EVAL} \circ \mathsf{tr}_{sk}$. In the absence of tr_{sk}, one can obtain y' by evaluating $y' \leftarrow \mathsf{EVAL}(x')$, which involves a large number of sequential steps.

Lastly, the verification procedure involves validating tr_{sk}'s correct computation statement with zero knowledge of tr_{sk}, where the Solver's answer (y'') serves as the statement witness. The validation arguments should be efficient, validating only the unique answer (i.e., a Verifier accepts only if $y'' = y'$).

In the following, we present a sketch construction of the proposed trapdoor-VDF. Essentially, the construction consists of evaluating a series of non backtracking *horizontal* isogenies with a large degree (representing EVAL), whereas the secret trapdoor is a shorter *horizontal* isogeny walk in graphs over a finite field $\mathbb{F}_p$ (representing tr_{sk}). Our public verification protocol uses bilinear pairings similar to BLS signature scheme [8], which is also used in [9] and [18].

4.1 Construction Elements

First, we define the public parameters that we will use to construct our scheme. Following that, we will briefly describe the supersingular isogeny graph, upon which both the long evaluation function and short trapdoor operate. Lastly, we will discuss the scheme group structure and the scheme's public verification method.

Selection of the Scheme Parameters. The parameters for our scheme are generated with the help of the following algorithms.

- $(p, \mathcal{L}, N, \mathcal{S}, \boldsymbol{t}, E_0) \leftarrow \mathsf{GGen}(1^\lambda, T)$ is a public parameter generation algorithm that takes a security parameter λ and difficulty T as an input and outputs:
  - a large odd prime p such that $p = 7 \bmod 8$,
  - a set $\mathcal{L}$ of n small distinct primes defined as follow

$$\mathcal{L} := \{2\} \cup \{l \in \mathsf{Primes}(6\log(p)^2) : (\frac{-p}{l}) = 1\},$$

  - a large prime N such that $N \notin \mathcal{L}$ and $N \mid p + 1$,
  - the set $\mathcal{S} := \{-e, \ldots, e\}$ for a positive integer e where $|\mathcal{S}| = 2e + 1$,
  - the vector $\boldsymbol{t} \leftarrow \mathcal{T}^n$ of n elements, where $\mathcal{T} := \{-\lceil \frac{m_{\max}}{n} \rceil, \ldots, \lceil \frac{m_{\max}}{n} \rceil\}$, $\sum_{m \in \boldsymbol{t}} \mathsf{abs}(m) = T < 2^{o(\lambda)}$, and a positive integer $m_{\max} < 2^{o(\lambda)}$,

- a supersingular elliptic curve $E_0/\mathbb{F}_p$ on the surface of $\mathcal{G}(\mathbb{F}_p, 2)$, where $|E_0(\mathbb{F}_p)| = p + 1$ and possibly with j-invariant $\mathsf{j}(E_0) \in \{0, 1728\}$.

– $E \leftarrow_R \mathsf{Alg}(1^\lambda, E', \mathcal{L})$ is a randomized algorithm that takes an initial curve E', the set $\mathcal{L}$, and security parameter λ. The algorithm Alg outputs a random supersingular elliptic curve $E/\mathbb{F}_p$ on the graph surface. Basically, this algorithm involves taking a random horizontal isogeny walk (see [18]) of a length of at least $\mathsf{Poly}(\log p)$. It is required that the probability of finding isogeny (path) between the output curve E and an initial curve E' be a negligible function in λ.

4.2 The Graph in Use and the Single Step of Computation

The graph over $\mathbb{F}_p$ described in [16, Theorem 2.7] is the abstraction behind our scheme. In our scheme, the graph consists of two levels, namely a surface (i.e., $\mathcal{E}_{\mathbb{F}_p}(\mathbb{Z}[\frac{1+\sqrt{-p}}{2}])$) and a floor (i.e., $\mathcal{E}_{\mathbb{F}_p}(\mathbb{Z}[\sqrt{-p}])$). In this graph, surface and floor have one-to-one connection by 2-isogenies, and there are no odd-degree isogenies connecting them. On the graph surface, there are two horizontal isogenies of degree $l_i \in \mathcal{L}$ from each vertex, whereas the floor is connected by isogenies of degree $l_i \in \mathcal{L}/\{2\}$.

In our construction, a *sequential walk* on the graph is represented by the group action of the ideal class group $\mathbf{cl}(\mathcal{O})$ of an imaginary quadratic order $\mathcal{O} \cong \mathbb{Z}[\frac{1+\sqrt{-p}}{2}]$ on the set $\mathcal{E}_{\mathbb{F}_p}(\mathcal{O})$. Hence, both EVAL and tr_{sk} act on the set $\mathcal{E}_{\mathbb{F}_p}(\mathcal{O})$. The choice of $\mathcal{L}$'s elements enables us to represent the elements of $\mathbf{cl}(\mathcal{O})$ as a product of ideals of small norm $\mathsf{N}(\mathfrak{l})$ such that $\mathsf{N}(\mathfrak{l}) \in \mathcal{L}$. Thus, we represent a *single step* in the sequential walk by the group action of an ideal of a norm in $\mathcal{L}$ that acts on $\mathcal{E}_{\mathbb{F}_p}(\mathcal{O})$.

The elements of $\mathcal{L}$ are chosen to be Elkies primes. Hence, the ideal $l_i\mathcal{O}$ splits into $\mathfrak{l}_i = (l_i, \pi - 1)$ and $\hat{\mathfrak{l}}_i = (l_i, \pi + 1)$ for every $l_i \in \mathcal{L}$ (i.e., $l_i\mathcal{O} = \mathfrak{l}_i\hat{\mathfrak{l}}_i, \forall l_i \in \mathcal{L}$). This defines the direction of a *single step*. From every vertex in the graph surface, there are two actions of an ideal with norm l_i that can be applied, either $\mathfrak{l}_i$ or $\hat{\mathfrak{l}}_i$. Further, each direction can be computed via an isogeny using Vélu's formulas [33]. In other words, a single step in one direction, denoted by ϕ_{l_i}, is an isogeny of a kernel of order l_i intersecting with $\mathsf{ker}(\pi - [1])$. As for the step in the opposite direction, denoted by $\phi_{l_i}^{-1}$, represents an isogeny of a kernel of order l_i intersecting with $\mathsf{ker}(\pi + [1])$. Finally, for an integer m_i, we denote l_i-sequential walk by $\phi_{l_i}^{m_i}$, which represents m_i sequences of l_i-isogeny evaluation in the same direction.

The Long Evaluation Function. As with [18] and [31], the long evaluation is represented by an isogeny of large degree, exponential in T,

$$\mathsf{EVAL} := \phi_{\mathcal{L}}^{\boldsymbol{t}} : E \rightarrow E_A$$

which is a composite of all $\phi_{l_i}^{m_i}$ for all $l_i \in \mathcal{L}$ and $m_i \in \boldsymbol{t}$ (i.e., $\phi_{\mathcal{L}}^{\boldsymbol{t}} := \phi_{l_n}^{m_n} \circ_n \cdots \circ_1 \phi_{l_1}^{m_1}$, where $m_i \in \boldsymbol{t}$ and $|\boldsymbol{t}| = n$). The degree of $\phi_{\mathcal{L}}^{\boldsymbol{t}}$ is $\prod_{i=1}^{n} l_i^{\mathsf{abs}(m_i)}$ and it has a difficulty of $\sum_{i=1}^{n} \mathsf{abs}(m_i) = T$. The presumption is that to compute EVAL efficiently, all T composites of $\phi_{\mathcal{L}}^{\boldsymbol{t}}$ must be evaluated sequentially.

The Secret Trapdoor and Secret Trapdoor Set. As with [11,31], we want to be able to efficiently sample a random trapdoor from their ensemble $\mathcal{F}$. We will therefore randomly sample $sk := \boldsymbol{s} \leftarrow_R \mathcal{S}^n$ to represent the trapdoor as

$$\mathsf{tr}_{sk} := \phi_{\mathcal{L}}^{\boldsymbol{s}} : E \rightarrow E_B.$$

The set $\mathcal{S}$ is chosen so that the trapdoor $\phi_{\mathcal{L}}^{\boldsymbol{s}}$ is much shorter than $\phi_{\mathcal{L}}^{\boldsymbol{t}}$. However, the size of $\mathcal{S}$ must be large enough to ensure that there exist $|\mathcal{S}|^n \geq 2^{2\lambda}$ possible secrets.

4.3 The Scheme Group Structure and Its Public Verification

Let G_i^E be a subgroup of $E[N]$ where $E \in \mathcal{E}_{\mathbb{F}_p}(\mathcal{O})$ and $i \in [N]$. The trace-zero and the base-field subgroup of $E[N]$ are represented by subscripts 1 and 2, respectively[1]. Under the assumption that (i) N is coprime to $|\mathsf{ker}(\phi_{\mathcal{L}}^{\boldsymbol{t}})|$ and $N \neq p$, and (ii) E and E_A are isogenous (so that $|E[N]| = |E_A[N]|$), the surjective morphism $\phi_{\mathcal{L}}^{\boldsymbol{t}} : E[N] \rightarrow E_A[N]$ must also be injective induced by the Lagrange's theorem; hence it is a bijective group homomorphism on the N-torsion subgroup (similar argument applies for $\phi_{\mathcal{L}}^{\boldsymbol{s}}$). Let σ^{-1} be a *quadratic twist* defined as follows

$$\sigma^{-1} : E \rightarrow E^{(d)} \in \mathcal{E}_{\mathbb{F}_p}(\mathcal{O})$$
$$\sigma^{-1} : G_{i\in\{1,2\}}^E \rightarrow G_{2i \bmod 3}^{E^{(d)}}$$

where $E^{(d)}$ is a twist of the curve E. The inverse quadratic twist (i.e., $\sigma : E^{(d)} \rightarrow E$) is efficiently computable and it is defined as follows

$$\sigma : (x, y) \rightarrow (x/w^2, y/w^3)$$

where $w \in \mathbb{F}_{p^2}$, $w \notin \mathbb{F}_p$, and it has $w^2 \in \mathbb{F}_p$. In our configuration, it should be noted that $\sigma \circ \phi_{l_i}^{m_i}(E)$ and $\phi_{l_i}^{-m_i} \circ \sigma(E)$ are equivalent.

Pairing Based Public Verification in a Nutshell Let $\hat{\mathsf{e}}_N^E : E[N] \times E[N] \rightarrow \mu_N$ be a non-degenerate Weil pairing map on E, where $\mu_N \subset \mathbb{F}_{p^2}^*$. A non-trivial bilinear Weil pairing can be defined as $\hat{\mathsf{e}}_N^E : G_1^E \times G_2^E \rightarrow \mu_N$. This definition is equivalent to $\hat{\mathsf{e}}_N^E : \sigma(G_2^{E^{(d)}}) \times G_2^E \rightarrow \mu_N$, which is efficiently computable with inputs of short representation. Let E, $E_A = \phi_{\mathcal{L}}^{\boldsymbol{t}}(E)$, $E_B = \phi_{\mathcal{L}}^{\boldsymbol{s}}(E)$, $E_{AB} = \phi_{\mathcal{L}}^{\boldsymbol{s}}(E_A)$, and $E_{BA} = \phi_{\mathcal{L}}^{\boldsymbol{t}}(E_B)$ be a set of isogenous supersingular elliptic curves. The public verification proceeds as follows:

- A Challenger, Verifier, and Solver all share as an input the description of the isogeny $\phi_{\mathcal{L}}^{\boldsymbol{t}} : E \rightarrow E_A$ of difficulty T, and the point pair $(x^{G_1^E}, y^{G_1^{E_A}}) \in G_1^E \times G_1^{E_A}$ such that $y^{G_1^{E_A}} = \phi_{\mathcal{L}}^{\boldsymbol{t}}(x^{G_1^E})$. All inputs are chosen for a security parameter λ.

[1] The subgroup G_1^E is defined as $G_1^E := E[N] \cap \mathsf{ker}(\pi + [1])$, whereas, $G_2^E := E[N] \cap \mathsf{ker}(\pi - [1])$.

- A Challenger selects $\boldsymbol{s} \leftarrow_R \mathcal{S}^n$ that defines the secret short isogeny $\phi_{\mathcal{L}}^{\boldsymbol{s}} : E \to E_B$; then, it
  - randomly samples integers k_A, k_B, and k_{AB} such that $k_A k_B^{-1} k_{AB}^{-1} = \deg \phi_{\mathcal{L}}^{\boldsymbol{s}} (\mathrm{mod} N)$,
  - computes and broadcasts the challenge $([k_A]Q^{G_2^{E_A}}, \hat{x}^{G_1^{E_B}}, Q^{G_2^{E_{AB}}}) \in G_2^{E_A} \times G_1^{E_B} \times G_2^{E_{AB}}$ such that $\hat{x}^{G_1^{E_B}} = [k_B]\phi_{\mathcal{L}}^{\boldsymbol{s}}(x^{G_1^E})$, $Q^{G_2^{E_A}} \leftarrow_R G_2^{E_A}$, and $Q^{G_2^{E_{AB}}} = [k_{AB}]\phi_{\mathcal{L}}^{\boldsymbol{s}}(Q^{G_2^{E_A}})$.
- A Solver computes the answer $\hat{y}^{G_1^{E_{BA}}} \leftarrow \phi_{\mathcal{L}}^{\boldsymbol{t}}(\hat{x}^{G_1^{E_B}})$, which requires T sequential isogeny evaluations, and then broadcasts $\hat{y}^{G_1^{E_{BA}}}$.
- A Verifier outputs ACCEPT if $\hat{y}^{G_1^{E_{BA}}}$ satisfies the following equality

$$\hat{\mathrm{e}}_N^{E_A}(y^{G_1^{E_A}}, [k_A]Q^{G_2^{E_A}}) = \hat{\mathrm{e}}_N^{E_{AB}}(\hat{y}^{G_1^{E_{BA}}}, Q^{G_2^{E_{AB}}}), \tag{2}$$

otherwise returns REJECT.

From Eq. (1), the above-mentioned verification is complete, as shown below.

$$\begin{aligned}
\hat{\mathrm{e}}_N^{E_A}(y^{G_1^{E_A}}, [k_A]Q^{G_2^{E_A}}) &= \hat{\mathrm{e}}_N^{E_{AB}}(\phi_{\mathcal{L}}^{\boldsymbol{t}} \circ [k_B]\phi_{\mathcal{L}}^{\boldsymbol{s}}(x^{G_1^E}), [k_{AB}]\phi_{\mathcal{L}}^{\boldsymbol{s}}(Q^{G_2^{E_A}})), \\
\hat{\mathrm{e}}_N^{E_A}(y^{G_1^{E_A}}, [\deg \phi_{\mathcal{L}}^{\boldsymbol{s}}]Q^{G_2^{E_A}}) &= \hat{\mathrm{e}}_N^{E_{AB}}(\phi_{\mathcal{L}}^{\boldsymbol{s}}(y^{G_1^{E_A}}), \phi_{\mathcal{L}}^{\boldsymbol{s}}(Q^{G_2^{E_A}})).
\end{aligned}$$

Moreover, it is obvious that the correctness of the scheme depends on the terminal elliptic curves (i.e., E_{AB} and E_{BA}) being identical. To be consistent with previous work [18], Fig. 1 shows the proof system's structure.

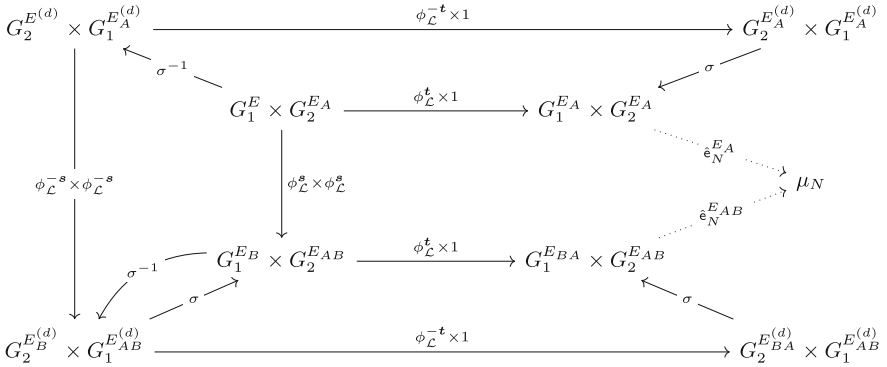

Fig. 1. The diagram illustrates the structure of the proposed trapdoor-VDF's proof system. The map $\phi_{\mathcal{L}}^{-\boldsymbol{t}}$ is the $\mathcal{L}$-sequential walk $\phi_{\mathcal{L}}^{\boldsymbol{t}}$ in the opposite direction (and similar for $\phi_{\mathcal{L}}^{\boldsymbol{s}}$).

4.4 An Instance of the Trapdoor-VDF

The following is a formal description of a trapdoor-VDF instance constructed from the action of class groups on supersingular elliptic curves with bilinear pairings being used for public verification. They are all defined by the parameters chosen in the previous section.

Setup$(1^\lambda, T)$:
$(\mathcal{L}, N, p, \mathcal{S}, \boldsymbol{t}, E_0) \leftarrow \mathsf{GGen}(1^\lambda, T)$
$E \leftarrow_R \mathsf{Alg}(1^\lambda, E_0, \mathcal{L})$
$x^{G_1^E} \leftarrow_R G_1^E$, s.t $\langle x^{G_1^E} \rangle = G_1^E$
$E_A \leftarrow \phi_{\mathcal{L}}^{\boldsymbol{t}}(E)$
$y^{G_1^{E_A}} \leftarrow \phi_{\mathcal{L}}^{\boldsymbol{t}}(x^{G_1^E})$
$pk \leftarrow$
$(\mathcal{L}, N, p, \mathcal{S}, \boldsymbol{t}, E, E_A, x^{G_1^E}, y^{G_1^{E_A}})$
return (pk)

Solver$(pk, \mathbf{c})$:
if $c.Q^{G_2^{E_A}} \notin G_2^{E_A}$ **then**
  return REJECT
if $c.\hat{x}^{G_1^{E_B}} \notin G_1^{E_B}$ **then**
  return REJECT
if $c.Q^{G_2^{E_{AB}}} \notin G_2^{E_{AB}}$ **then**
  return REJECT
$E_{BA} \leftarrow \phi_{\mathcal{L}}^{\boldsymbol{t}}(\mathbf{c}.E_B)$
$\hat{y}^{G_1^{E_{BA}}} \leftarrow \phi_{\mathcal{L}}^{\boldsymbol{t}}(\mathbf{c}.\hat{x}^{G_1^{E_B}})$
$\mathbf{a} \leftarrow (E_{BA}, \hat{y}^{G_1^{E_{BA}}})$
return $(\mathbf{a})$

Challenger(pk):
$\boldsymbol{s} \leftarrow_R pk.\mathcal{S}^n$
$k_B^{-1}, k_{AB}^{-1} \leftarrow_R (\mathbb{Z}_N^*)^2$
$k_A = k_B k_{AB} \deg \phi_{\mathcal{L}}^{\boldsymbol{s}} \bmod N$
$Q^{G_2^{E_A}} \leftarrow_R G_2^{E_A}$
$E_{AB} \leftarrow \phi_{\mathcal{L}}^{\boldsymbol{s}}(pk.E_A)$
$Q^{G_2^{E_{AB}}} \leftarrow [k_{AB}]\phi_{\mathcal{L}}^{\boldsymbol{s}}(Q^{G_2^{E_A}})$
$E_B \leftarrow \phi_{\mathcal{L}}^{\boldsymbol{s}}(pk.E)$
$\hat{x}^{G_1^{E_B}} \leftarrow [k_B]\phi_{\mathcal{L}}^{\boldsymbol{s}}(pk.x^{G_1^E})$
$Q^{G_2^{E_A}} \leftarrow [k_A]Q^{G_2^{E_A}}$
$\mathbf{c}_{\mathsf{EVAL}} \leftarrow (E_B, \hat{x}^{G_1^{E_B}})$
$\mathbf{c}_{\text{verify}} \leftarrow$
$(E_{AB}, Q^{G_2^{E_A}}, Q^{G_2^{E_{AB}}})$
$\mathbf{c} \leftarrow (\mathbf{c}_{\mathsf{EVAL}}, \mathbf{c}_{\text{verify}})$
return $(\mathbf{c})$

Verify$(pk, \mathbf{c}, \mathbf{a})$:
if $a.\hat{y}^{G_1^{E_{BA}}} \notin G_1^{E_{BA}}$ **then**
  return REJECT
if $c.Q^{G_2^{E_{AB}}} \notin G_2^{E_{AB}}$ **then**
  return REJECT
if $c.Q^{G_2^{E_A}} \notin G_2^{E_A}$ **then**
  return REJECT
$l_{\text{Side}} \leftarrow \hat{\mathsf{e}}_N^{E_A}(pk.y^{G_1^{E_A}}, \mathbf{c}.Q^{G_2^{E_A}})$
$r_{\text{Side}} \leftarrow \hat{\mathsf{e}}_N^{E_{BA}}(\mathbf{a}.\hat{y}^{G_1^{E_{BA}}}, \mathbf{c}.Q^{G_2^{E_{AB}}})$
if $l_{Side} = r_{Side}$ & $l_{Side} \neq 1_{\mu_N}$ **then**
  return ACCEPT
return REJECT

Fig. 2. The proposed trapdoor-VDF instance is defined by the four algorithms described above.

Discussion. It is imperative that a trusted process runs Setup due to an attack that exploits an elliptic curve with known endomorphism rings (see [18]). To generate the public parameters, the Setup must select E and then compute $\phi_{\mathcal{L}}^{\boldsymbol{t}}$, which takes $O(T)$ steps. A Challenger must sample a one-time secret $\boldsymbol{s}$ to compute $\hat{x}^{G_1^{E_B}}$ and $Q^{G_2^{E_{AB}}}$, with the evaluation time being a polynomial function in λ. Essentially the challenge consists of three elliptic curve points. To obtain the answer, the Solver must compute $\phi_{\mathcal{L}}^{\boldsymbol{t}}$, which is a separable isogeny of a degree exponential in T. Typically, the best approach for computing $\phi_{\mathcal{L}}^{\boldsymbol{t}}$ is to sequentially compute each of its T compositions, of degrees $l_i \in \mathcal{L}$, using Vélu's formulas. A single point on the target curve E_{BA} serves as both the answer and the proof. The verification involves two bilinear pairings. It also involves determining whether all points are members of the appropriate group, a relatively trivial operation.

5 Security

We will show that the trapdoor-VDF proposed in Sect. 4.4 is secure well-defined with an honest challenge and the aforementioned setup that takes $O(T)$ steps.

Theorem 1. *The trapdoor-VDF instance in Sect. 4.4 is a secure well-defined trapdoor-VDF with a $O(T)$ steps long* Setup *and under the assumption of an honest Challenger.*

Correctness, and Uniqueness. The following assumes that all statements are true for any λ, T, $pk \leftarrow_R \mathsf{Setup}(1^\lambda, T)$. For all *honest* $\mathbf{c} \leftarrow_R \mathsf{Challenger}(pk)$, the correctness argument implies that any honest evaluator should be able to obtain the answer $\mathbf{a} \leftarrow \mathsf{Solver}(pk, \mathbf{c})$ which is Verify's acceptable answer.

However, the correctness argument is not complete yet. We also require that the Challenger and Solver must land on the same terminal elliptic curve and not only curves on the same isomorphism class. In other words, the evaluation output $(\phi_{\mathcal{L}}^{\boldsymbol{t}}(\mathbf{c}.\hat{x}^{G_1^{E_B}}))$ and the challenge $(\mathbf{c}.Q^{G_2^{E_{AB}}})$ have to be on the same curve (i.e., $E_{BA} = E_{AB}$). This is required so that the Verifier can evaluate Eq. (2). The verification in Eq. (2) is likewise unique, as we employ a similar version of the [8]'s (and [18]) verifier.

Lemma 1. *Given any λ, T, and $pk \leftarrow_R$ Setup$(1^\lambda, T)$, a Solver's honest output $\boldsymbol{a}.\hat{y}^{G_1^{E_{BA}}}$ is indeed a point on the Challenger's terminal curve (E_{AB}) such that $\boldsymbol{a}.\hat{y}^{G_1^{E_{BA}}} = \hat{y}^{G_1^{E_{AB}}}$ and* ACCEPT $\leftarrow$ *Verify$(pk, \boldsymbol{c}, \boldsymbol{a})$, for all honest $\boldsymbol{c} \leftarrow_R$ Challenger(pk); further, there is no answer $\boldsymbol{a}'$ that is a valid answer (i.e.,* ACCEPT $\leftarrow$ *Verify$(pk, \boldsymbol{c}, \boldsymbol{a}')$) unless $\boldsymbol{a}' = \boldsymbol{a} = \phi_{\mathcal{L}}^{\boldsymbol{t}}(\boldsymbol{c}.\hat{x}^{G_1^{E_B}})$.*

Proof. The proof to the first part of the lemma follows from the result of Leonardi [23, Theorem 3.1]. Leonardi's result implies the equivalency between the action of $\phi_{\mathcal{L}}^{\boldsymbol{s}} \circ \phi_{\mathcal{L}}^{\boldsymbol{t}}$ and $\phi_{\mathcal{L}}^{\boldsymbol{t}} \circ \phi_{\mathcal{L}}^{\boldsymbol{s}}$ when all computed with Vélu's formulas. As for the uniqueness, we have set our parameters so that (i) as mentioned in Sect. 4.3, the surjective morphism $\phi_{\mathcal{L}}^{\boldsymbol{t}} : E_B[N] \to E_{BA}[N]$ is a group isomorphism on the N-torsion subgroup and (ii) as discussed in [18], Eq. (2)'s right-hand side is a group isomorphism from $G_1^{E_{BA}}$ to μ_N for a given $\mathbf{c}.Q^{G_2^{E_{AB}}}$. Hence, $\mathsf{Verify}((pk, \mathbf{c}), \mathbf{a}')$ outputs ACCEPT if and only if $\mathbf{a}' = \mathbf{a} = \phi_{\mathcal{L}}^{\boldsymbol{t}}(\mathbf{c}.\hat{x}^{G_1^{E_B}}) \in G_1^{E_{BA}}$.

Soundness. Since there is only one valid answer, the verification is unique, as is a property of [8]'s verifier. Thus, and similar to [18], our trapdoor-VDF is perfectly sound.

Sequentiality. In the following, we introduce the concept of sequentiality in our trapdoor-VDF by presenting the shortcut game.

Definition 3 (The shortcut game). *Let λ and T be a security and a difficulty parameters, respectively. Let $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2)$ be a party that participates in the following game. (i) A trusted process computes and publishes $pk \leftarrow_R$ Setup$(1^\lambda, T)$,*

(ii) $\mathcal{A}$ *preforms a pre-computation in time* $\mathsf{Poly}(T, \lambda)$ *and outputs* $\mathsf{pc} \leftarrow \mathcal{A}_1(pk)$, *(iii) a trusted process then computes and publishes* $\boldsymbol{c} \leftarrow_R \mathsf{Challenge}(pk)$, *(iv)* $\mathcal{A}$ *computes and outputs* $\hat{y}^{G_1^{E_{BA}}} \leftarrow \mathcal{A}_2(pk, \boldsymbol{c}, \mathsf{pc})$ *in parallel time less than* T, *where* $\hat{y}^{G_1^{E_{BA}}} \in G_1^{E_{BA}}$.

To win the game, $\mathcal{A}$*'s output,* $\hat{y}^{G_1^{E_{BA}}}$, *is required to be the correct evaluation of* $\phi_{pk.\mathcal{L}}^{pk.\boldsymbol{t}}$ *on* $\boldsymbol{c}.\hat{x}^{G_1^{E_B}}$ *(i.e., to be equal to* $\boldsymbol{a} \leftarrow \mathsf{Solver}(pk, \boldsymbol{c})$*).*

The proposed trapdoor-VDF is sequential if there is no polynomially bounded player $\mathcal{A}$ with a non-negligible probability of winning the above shortcut game. A player $\mathcal{A}$ can win the game by finding (i) a shorter isogeny than $\phi_{\mathcal{L}}^{\boldsymbol{t}}$, (ii) a faster method for computing $\phi_{\mathcal{L}}^{\boldsymbol{t}}$, (iii) the inverse of pairing Eq. (2), or (iv) the secret isogeny $\phi_{\mathcal{L}}^{\boldsymbol{s}}$ (or a short equivalent isogeny). The three former points have already been discussed in [18, Section 6]. With regards to the last point, the sequentiality is also determined by recovering the secret isogeny $\mathsf{tr}_{sk} := \phi_{\mathcal{L}}^{\boldsymbol{s}}$ or obtaining a short equivalent isogeny. In other words, this is equivalent to recovering the ideal class $[\mathfrak{b}] \in \mathbf{cl}(\mathcal{O})$ that is computed via the secret isogeny $\phi_{\mathcal{L}}^{\boldsymbol{s}}$.

Definition 4 (Key recovery [11]). *Given two supersingular elliptic curves* E *and* E_B *defined over* $\mathbb{F}_p$ *with the same* $\mathbb{F}_p$*-rational endomorphism ring* $\mathcal{O}$, *find an ideal* $\mathfrak{b}$ *such that* $[\mathfrak{b}]E = E_B$ *and* $\mathfrak{b}$ *is the product of ideals of small norm in* $\mathcal{L}$.

There are several possible attacks for recovering the secret, some of which are discussed in [11].

Brute-force Attack on the Secret Key. A basic attack searches over all potential keys. For a secret $\boldsymbol{s}$, its corresponding secret ideal class $[\mathfrak{b}]$ is represented as $[\mathfrak{l}_1^{m_1}\mathfrak{l}_2^{m_2}\cdots\mathfrak{l}_n^{m_n}] \in \mathbf{cl}(\mathcal{O})$ such that $\mathsf{N}(\mathfrak{l}_i) = l_i \in \mathcal{L}$ and $m_i \in \boldsymbol{s}$. Thus, it can be argued that $[\mathfrak{b}]$ has several representations (i.e., there are an equivalent $\hat{\boldsymbol{s}} \leftarrow \mathcal{S}^n$ that yields $[\mathfrak{b}] \in \mathbf{cl}(\mathcal{O})$), which is vulnerable to exhaustive search. Castryck et al. [11] show that the expected number of equivalent representations is $|\mathcal{S}|^n/\mathsf{ord}(G)$, assuming that $\mathbf{cl}(\mathcal{O})$ is almost cyclic with a very large cyclic component of order $\mathsf{ord}(G)$ close to $|\mathbf{cl}(\mathcal{O})|$. Therefore, it suffices to choose $n \log |\mathcal{S}| \approx \log(\sqrt{p})$, where $|\mathbf{cl}(\mathcal{O})| \approx \sqrt{p}$.

Pohlig-Hellman-style Attack. To our knowledge, the Pohlig-Hellman style attacks cannot be effectively applied to the construction in Sect. 4.4 to recover the secret degree $(\deg \phi_{\mathcal{L}}^{\boldsymbol{s}})$ or the answer $(\hat{y}^{G_1^{E_{BA}}})$.

Castryck-Decru-style Attack [10]. As in [19], we mask the torsion points in pairing-based verification. Therefore, the point pairs $(y^{G_1^{E_A}}, [k_A]Q^{G_2^{E_A}})$ on E_A and $(\hat{y}^{G_1^{E_{AB}}}, Q^{G_2^{E_{AB}}})$ on E_{AB} are not images of the secret isogeny $\phi_{\mathcal{L}}^{\boldsymbol{s}}$. In addition, the secret isogeny $\phi_{\mathcal{L}}^{\boldsymbol{s}}$ has a hidden degree determined by $\boldsymbol{s}$, which is similar to the countermeasure suggested in [27]. Hence, to the best of our knowledge, our proposal is naturally resistant to a Castryck-Decru-style attack because we conceal torsion points' preimage and the secret isogeny degree.

Meet-in-the-middle Attack (MITM). To find the path $(\phi^{s}_{\mathcal{L}} : E_A \rightarrow E_{AB})$ in the isogeny graph $\bigcup_{l_i \in \mathcal{L}} \mathcal{G}(\mathbb{F}_p, l_i)$, MITH starts from E_A and E_{AB} to construct search trees. The attacker looks for the collision in the tree. The halfway point between E_A and E_{AB} is the isogeny evaluation of the halves of the set $\mathcal{L}$, $\mathcal{L}_{\textbf{left}} := \{l_1, l_2, \ldots, l_{n/2}\}$, and $\mathcal{L}_{\textbf{right}} := \{l_{\frac{n}{2}+1}, \ldots, l_n\}$ (for simplicity assume that n is even). Let $\mathcal{J}$ be the set size of all elliptic curves defined over $\mathbb{F}_p$ that are isogenous to E_A, constrained to $\mathcal{S}$, in the halfway to E_{AB}. We observe that the set size is $|\mathcal{J}| = |\mathcal{S}|^{n/2} - 1$; thus the attack average-case isogeny computation is about $\approx 2^{n \log(|\mathcal{S}|)/2}$. Setting $n \log(|\mathcal{S}|)$ to be $\approx \log(p)/2$, the attack's average-case complexity is $2^{\log(p)/4}$.

6 Conclusion

We have presented a new class of trapdoor-VDFs that have a large ensemble of trapdoors for each VDF instantiation. In the proposed scheme, a secret trapdoor, chosen randomly from the ensemble by Challenger, is used to generate a challenge. In a few steps, Challenger can obtain the answer to a challenge by using the secret trapdoor. Without knowing the secret trapdoor, a Solver on the other hand must perform the long sequential computations. We have also presented a trapdoor-VDF construction based on [11,18,31].

Our future work will focus on proving that the proposed construction eliminates computational advantage of the instance generator (i.e., not being able to obtain the trapdoor or break the scheme sequentiality). Further, the setup of the proposed instantiation requires evaluating the lengthy sequential computations, so it is beneficial to examine newer approaches to make the procedure efficient.

References

1. Abusalah, H., Kamath, C., Klein, K., Pietrzak, K., Walter, M.: Reversible proofs of sequential work. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019. LNCS, vol. 11477, pp. 277–291. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-17656-3_10
2. Baum, C., David, B., Dowsley, R., Nielsen, J.B., Oechsner, S.: CRAFT: composable randomness and almost fairness from time. IACR Cryptology ePrint Archive, p. 784 (2020)
3. Baum, C., David, B., Dowsley, R., Nielsen, J.B., Oechsner, S.: TARDIS: a foundation of time-lock puzzles in UC. In: Canteaut, A., Standaert, F.-X. (eds.) EUROCRYPT 2021. LNCS, vol. 12698, pp. 429–459. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-77883-5_15
4. Bellare, M., Halevi, S., Sahai, A., Vadhan, S.P.: Many-to-one trapdoor functions and their relation to public-key cryptosystems. IACR Cryptol., p. 19 (1998)
5. Block, A.R., Holmgren, J., Rosen, A., Rothblum, R.D., Soni, P.: Time- and space-efficient arguments from groups of unknown order. In: Malkin, T., Peikert, C. (eds.) CRYPTO 2021. LNCS, vol. 12828, pp. 123–152. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-84259-8_5

6. Boneh, D., Bonneau, J., Bünz, B., Fisch, B.: Verifiable delay functions. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018. LNCS, vol. 10991, pp. 757–788. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-96884-1_25
7. Boneh, D., Franklin, M.K.: Efficient generation of shared RSA keys. J. ACM **48**(4), 702–722 (2001)
8. Boneh, D., Lynn, B., Shacham, H.: Short signatures from the Weil pairing. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 514–532. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45682-1_30
9. Burdges, J., De Feo, L.: Delay encryption. In: Canteaut, A., Standaert, F.-X. (eds.) EUROCRYPT 2021. LNCS, vol. 12696, pp. 302–326. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-77870-5_11
10. Castryck, W., Decru, T.: An efficient key recovery attack on SIDH (preliminary version). Cryptology ePrint Archive, Paper 2022/975 (2022)
11. Castryck, W., Lange, T., Martindale, C., Panny, L., Renes, J.: CSIDH: an efficient post-quantum commutative group action. In: Peyrin, T., Galbraith, S. (eds.) ASIACRYPT 2018. LNCS, vol. 11274, pp. 395–427. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03332-3_15
12. Chavez-Saab, J., Rodríguez-Henríquez, F., Tibouchi, M.: Verifiable isogeny walks: towards an isogeny-based postquantum VDF. In: AlTawy, R., Hülsing, A. (eds.) SAC 2021. LNCS, vol. 13203, pp. 441–460. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-99277-4_21
13. Chvojka, P., Jager, T., Slamanig, D., Striecks, C.: Versatile and sustainable timed-release encryption and sequential time-lock puzzles (extended abstract). In: Bertino, E., Shulman, H., Waidner, M. (eds.) ESORICS 2021. LNCS, vol. 12973, pp. 64–85. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-88428-4_4
14. Cline, D., Dryja, T., Narula, N.: Clockwork: an exchange protocol for proofs of non front-running (2020). https://dci.mit.edu/clockwork, the Stanford Blockchain Conference
15. Cohen, B., Pietrzak, K.: Simple proofs of sequential work. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018. LNCS, vol. 10821, pp. 451–467. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-78375-8_15
16. Delfs, C., Galbraith, S.D.: Computing isogenies between supersingular elliptic curves over $\mathbb{F}_p$. Des. Codes Cryptogr. **78**(2), 425–440 (2016)
17. Ephraim, N., Freitag, C., Komargodski, I., Pass, R.: Continuous verifiable delay functions. In: Canteaut, A., Ishai, Y. (eds.) EUROCRYPT 2020. LNCS, vol. 12107, pp. 125–154. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-45727-3_5
18. De Feo, L., Masson, S., Petit, C., Sanso, A.: Verifiable delay functions from supersingular isogenies and pairings. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT 2019. LNCS, vol. 11921, pp. 248–277. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-34578-5_10
19. Fouotsa, T.B.: SIDH with masked torsion point images. Cryptology ePrint Archive, Paper 2022/1054 (2022). https://eprint.iacr.org/2022/1054
20. Freitag, C., Komargodski, I., Pass, R., Sirkin, N.: Non-malleable time-lock puzzles and applications. In: Nissim, K., Waters, B. (eds.) TCC 2021. LNCS, vol. 13044, pp. 447–479. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-90456-2_15
21. Kohel, D.R.: Endomorphism rings of elliptic curves over finite fields. Ph.D. thesis, University of California, Berkeley (1996)
22. Lenstra, A.K., Wesolowski, B.: A random zoo: sloth, unicorn, and trx. Cryptology ePrint Archive, Report 2015/366 (2015)

23. Leonardi, C.: A note on the ending elliptic curve in SIDH. IACR Cryptology ePrint Archive 2020, 262 (2020). https://eprint.iacr.org/2020/262
24. Loe, A.F., Medley, L., O'Connell, C., Quaglia, E.A.: A practical verifiable delay function and delay encryption scheme. IACR Cryptology ePrint Archive, p. 1293 (2021)
25. Loe, A.F., Medley, L., O'Connell, C., Quaglia, E.A.: Tide: A novel approach to constructing timed-release encryption. In: 27th Australasian Conference on Information Security and Privacy, 28–30 November 2022, Wollongong, Australia (2022)
26. Mahmoody, M., Moran, T., Vadhan, S.P.: Publicly verifiable proofs of sequential work. In: ITCS, pp. 373–388. ACM (2013)
27. Moriya, T.: Masked-degree SIDH. Cryptology ePrint Archive, Paper 2022/1019 (2022). https://eprint.iacr.org/2022/1019
28. Pietrzak, K.: Simple verifiable delay functions. In: ITCS. LIPIcs, vol. 124, pp. 60:1–60:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2019)
29. Rivest, R.L., Shamir, A., Wagner, D.A.: Time-lock puzzles and timed-release crypto (1996)
30. Rotem, L.: Simple and efficient batch verification techniques for verifiable delay functions. In: Nissim, K., Waters, B. (eds.) TCC 2021. LNCS, vol. 13044, pp. 382–414. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-90456-2_13
31. Shani, B.: A note on isogeny-based hybrid verifiable delay functions. IACR Cryptology ePrint Archive, p. 205 (2019)
32. Silverman, J.H.: The Arithmetic of Elliptic Curves, Graduate Texts in Mathematics, vol. 106. Springer, New York (1986). https://doi.org/10.1007/978-1-4757-1920-8
33. Vélu, J.: Isogénies entre courbes elliptiques. CR Acad. Sci. Paris Sér. A **273**, 305–347 (1971)
34. Wesolowski, B.: Efficient verifiable delay functions. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019. LNCS, vol. 11478, pp. 379–407. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-17659-4_13