Isogeny-based time-release cryptography

Parsa Tasbihgou

Supervisor: Dr. Boris Fouotsa

March 2024

EPFL

Overview

Time-release primitives

Delay encryption

- In delay encryption there are no senders or receivers
- Messages are encrypted by a session id.
- Decryption requires the associated session key.
- The session key is extracted from the session id and the "extraction" process is expected to take time T.
- Once the session key is extracted any ciphertext in the current session can be decrypted.

Delay encryption

There are 4 algorithms, λ is the security parameter and T is the delay parameter:

- Setup(λ, T) $\rightarrow$ (ek, pk):
  Setup algorithm should run in time poly(λ, T).
- Extract(ek, id) $\rightarrow$ idk:
  Extract is expected to run in time exactly T.
- Encaps(pk, id) $\rightarrow$ (c, k):
  Encaps should run in time poly(λ).
- Decaps(pk, id, idk, c) $\rightarrow$ k:
  Decaps should run in time ploy(λ).

- Correctness:
  $(ek, pk) \leftarrow \text{Setup}(\lambda,\ T)$ and $idk \leftarrow \text{Extract}(ek, id)$ and $(c, k) = \text{Encaps}(pk, id)$
  $\Rightarrow \text{Decaps}(pk, id, idk, c) = k$.
- Δ-indistinguishable CPA game:
  - Precomputation:
    The adversary receives (ek, pk) and outputs algorithm D.
  - Challenge:
    The challenger receives a random id and computes (c, k_0) $\leftarrow$ Encaps(pk, id). It also chooses a random $k_1 \in K$ and a bit b $\in \{0, 1\}$ and outputs (id, c, k_b).
  - Guess:
    Algorithm D is run on (id, c, k_b). The adversary wins if D halts in time less than Δ and D(id, c, k_b) = b.

Verifiable Delay Function

A function $f : X \to Y$ such that computing $f(x)$ is a slow and sequential process for all $x \in X$ but for any $y \in Y$ verifying $f(x) = y$ is efficient.

- Setup$(\lambda, T) \to (ek, vk)$
  Setup should run in time poly(λ, T).
- Eval$(ek, x) \to (y, \pi)$
  This process is meant to infeasible in time less than T.
- Verify$(vk, x, y, \pi) \to True, False$
  Verification should run in poly$(\log(T), \lambda)$.

Security of VDF

- **Completeness:**
  The honest evaluator always convinces the verifier.
- **Soundness:**
  A VDF has soundness error ϵ if for any PPT algorithm A and $x \in X$ the following holds.

$$Pr\left(Verify(vk, x, y', \pi') = true \,\middle|\, \begin{array}{l} (vk, ek) \leftarrow Setup(T, \lambda), \\ (y', \pi') \leftarrow A(ek, x), \\ y' \neq f(x), \end{array} \right) \leq \epsilon(\lambda)$$

- **Sequentiality:**
  It is infeasible to compute $f(x)$ for any $x \in X$ in time less than T even with $poly(T)$ many CPUs.

Preliminaries

Isogenies

- A non-constant rational map $\phi : E \to E'$ on elliptic curves that takes $\mathcal{O}_E$ to $\mathcal{O}_{E'}$ is an isogeny.
- An isogeny $\phi : E \to E$ from a curve into it self is called an endomorphism.
- Frobenius endomorphism is defined $\pi(x, y) = (x^q, y^q)$.
- Degree of an isogeny is size of its kernel.
- Any isogeny $\phi : E \to E'$ has a dual isogeny $\hat{\phi} : E' \to E$ of the same degree.

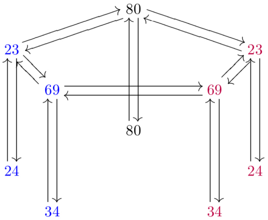
Endomorphisms

Set of all endomorphism on E denoted by $\mathrm{End}(E)$ forms a ring under addition and composition.

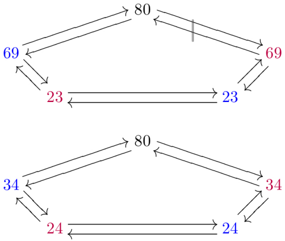
- $\mathrm{End}(E)$ is isomorphic to a maximal order in the quaternion algebra $B_{p,\infty}$.
- $\mathrm{End}_{\mathbb{F}_p}(E)$ is isomorphic to $\mathbb{Z}[\sqrt{-p}]$ or $\mathbb{Z}[\frac{1+\sqrt{-p}}{2}]$.
- $\{j(E) | E \text{ is supersingular}\} \leftrightarrow \{\text{Maximal orders in } B_{p,\infty}\}$
- $\{[\phi] | \phi \text{ is isogeny on } E\} \leftrightarrow \{\mathrm{cl}(\mathrm{End}(E))\}$

$\mathbb{F}_p$-restricted supersingular isogeny graph

- Number of j-invariants defined over $\mathbb{F}_p$ is about $\sqrt{p}$.
- If $\left(\frac{-p}{\ell}\right) = 1$ there are exactly two isogenies.
- If $\ell = 2$ and $p \equiv 7 \mod 8$, curves on the floor have one ascending isogeny and curves on the surface have two horizontal isogenies and one descending isogeny.



(a) $\ell = 2$



(b) $\ell = 7$

Figure: $p = 103$

Delay from isogenies

De Feo et al. isogeny-based VDF

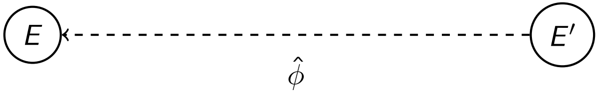
- Trusted setup(p):
  Sample a random supersingular curve $E/\mathbb{F}_p$.
- Setup(p, N, E, T):
  Get an isogeny walk $\phi : E \to E'$ of degree ℓ^T.
  Compute a point $P \in E(\mathbb{F}_P)$ of order N.
  Output (ϕ, E, P, $\phi(P)$).
- Evaluation($Q \in E'[N]$):
  Evaluate $\hat{\phi}(Q)$.
- Verification($P, \phi(P), Q$):
  Check $e_N(P, \hat{\phi}(Q)) = e'_N(\phi(P), Q)$.

$$E = E_0 \longrightarrow E_1 \dashrightarrow E_i \dashrightarrow E_T = E'$$

Isogeny Δ-shortcut game

Security of the isogeny-based VDF is defined by the following game.

- Precomputation:
  The adversary receives (N, p, E, E', ϕ) and outputs an algorithm S.
- Challenge:
  The challenger outputs a random point $Q \in E'[N]$.
- Guess:
  The adversary wins if $S(Q)$ halts in time less that Δ and $S(Q) = \hat{\phi}(Q)$.



Isogeny-based VDF security

- There are at least 4 ways to evaluate an isogeny chain faster:
  1. Parallelization
  2. Specialized hardware
  3. Optimized formulas
  4. Find a shorter isogeny

- It is possible to convince the verifier without evaluating $\hat{\phi}(Q)$.
  The verification is $e_N(P, Q') = e'_N(\phi(P), Q)$.
  Let $Q_0 \in E[N]$ such that $e_N(P, Q_0)$ generates μ_N.
  Compute $x = \log_{e_N(P,Q_0)} e'_N(\phi(P), Q)$ then $Q' = xQ_0$.

Isogeny-based VDF security

- If $\mathrm{cl}(\mathrm{End}_{\mathbb{F}_p}(E))$ is known, a short basis $B = (\mathcal{I}_{\ell_1}, \mathcal{I}_{\ell_2}, ..., \mathcal{I}_{\ell_n})$ can be computed so every ideal has a representative with small ℓ_∞-norm.
- If $\mathrm{End}(E)$ or $\mathrm{End}(E')$ is known then the isogeny ϕ could be translated into an ideal and converted into an ideal of small norm and finally translated to an isogeny of small degree.
- A quantum adversary can compute the class group in polynomial time and a classical adversary can do it in sub-exponential time.
- Random Walks in the full isogeny graph.

Summery of shortcut attacks

	Classical over $\mathbb{F}_p$	Classical over $\mathbb{F}_{p^2}$	Quantum over $\mathbb{F}_p$	Quantum over $\mathbb{F}_{p^2}$
Computing shortcut	$L_p(1/2)$	$O(p^{1/2})$	$polylog(p)$	$O(p^{1/4})$
Pairing inversion	$L_p(1/3)$	$L_p(1/3)$	$polylog(p)$	$polylog(p)$

To achieve λ bits of security, N should be a prime with 2λ bits and p a prime with λ^3 bits of the form $p = Nf - 1$.

isogeny-based delay encryption

- Trusted setup(λ):
  Generate a random supersingular curve $E/\mathbb{F}_p$.
- Untrusted setup(E, T):
  1. Start from E, get an ℓ^T-isogeny $\phi : E \to E'$.
  2. Choose a random point $P \in E[N]$ and evaluate $\phi(P)$.
  3. Publish ek = (E', ϕ), pk = (E', P, $\phi(P)$).
- Extract(E, E', ϕ, id):
  1. Output $\hat{\phi}(Q = H_1(id))$.
- Encaps$(E, E', P, \phi(P), id)$:
  1. Sample uniformly $r \in \mathbb{Z}_N$.
  2. Compute $k = e'_N(\phi(P), Q)^r$.
  3. Output (rP, k).
- Decaps$(E, E', \hat{\phi}(Q), rP)$:
  1. Compute $k = e_N(rP, \hat{\phi}(Q))$.

Bilinear isogeny shortcut game

Bilinear isogeny shortcut game:

- Precomputation:
  The adversary receives (N, p, E, E', ϕ) and outputs an algorithm S.
- Challenge:
  The challenger samples random $P \in E[N], Q \in E'[N]$.
- Guess:
  Run $S(P, Q)$. The adversary wins of S halts in time less that Δ and $S(P, Q) = e_N(P, \hat{\phi}(Q)) = e'_N(\phi(P), Q)$.

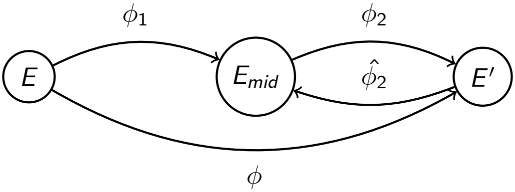
If the bilinear isogeny shortcut game is Δ'-hard, delay encryption is Δ-CPA IND where $\Delta \in \Delta' - o(\Delta')$ and H_1 is a random oracles.

Watermarking

Watermarking I

- Tie the evaluation of a VDF to the evaluator.
- A watermarking method is **complete** if the honest evaluator always convinces the honest verifier.
- A Watermarking method has **soundness error** ϵ if there is an adversary that given $\hat{\phi}(Q)$ and a watermarking, can generate a new valid watermarking in time $(1-\epsilon)T$.
- When the VDF evaluation has a proof, the proof can be signed to be tied to the evaluator.
- The isogeny-based VDF doesn't have a proof.

Watermarking II



1. $(E, E', E_{mid}, P, P_{mid} = \phi_1(P), \phi(P))$ is published as the setup.
2. Participant i selects a random element $s_i \in \mathbb{Z}_N$ as secret key and publishes $S_i = s_i\phi(P)$ as their public key.
3. i publishes a proof of knowledge of the secret key s_i.
4. i publishes $Q^i_{mid} = s_i Q_{mid}$ as her proof.
5. Verification: $e^{mid}_N(P_{mid}, Q^i_{mid}) = e'_N(S_i, Q)$.

Watermarking III

- Just compute $Q_{mid} = \hat{\phi}_2(Q)$ or $Q_{mid} = \phi_1^{-1}(\hat{\phi}(Q))$.

- Soundness error is $\frac{1}{2}$.

- Split the walk into n segments. Soundness error is $\frac{1}{n}$ and proof length is n

$$\begin{tikzcd} E \arrow[r, bend left, "\phi_1"] & E_1 \arrow[r, dashed, bend left] & E_i \arrow[r, dashed, bend left] & E' \end{tikzcd}$$

New watermarking

1. The setup for the delay encryption doesn't change.

$$E \xleftarrow{\hat{\phi}} E'$$

2. Participant i chooses a secret key $s_i \in \mathbb{Z}_N^*$ and publishes her public key
$S_i = (vk_i, ek_i) = (s_i\phi(P), s_i^{-1}(rP))$.
3. i publishes a non-interactive zero-knowledge proof of knowledge for s_i.
4. i publishes $Q_i = s_i\hat{\phi}(Q)$ as her watermarked evaluation.
5. Verify correctness and identity: $e_N(P, Q_i) = e'_N(vk_i, Q)$.
6. The session key is $k = e_N(s_i^{-1}rP, Q_i)$.

Random supersingular elliptic curves

Distributed trust

1. Start from $E_0 : y^2 = x^3 - x$.
2. Participant i checks all previous proofs.
3. Perform a random walk of length $c \log(p)$ to get isogeny $\phi_i : E_{i-1} \to E_i$.
4. Publish a proof of knowledge for ϕ_i.

$$\begin{tikzcd} E_0 \arrow[r, "\phi_1"'] & E_1 \arrow[r, dashed] & E_{i-1} \arrow[r, "\phi_i"'] & E_i \arrow[r, dashed] & E_n \end{tikzcd}$$

The proof has to be knowledge-sound when one of $\mathrm{End}(E_{i-1})$ or $\mathrm{End}(E_i)$ is known.

Proof of isogeny knowledge

Let F be a deterministic function that takes two curves E, E' as input and outputs two points $P \in E[(N, \pi - 1)]^o$, $Q \in E'[(N, \pi + 1)]^o$.

Prover		Verifier
Let $(P, Q) = F(E, E')$.		
		Choose a random $r \in \mathbb{Z}_{\mathbb{N}}{}^*$.
	$\xleftarrow{\text{Send } r}$	
Let $R = r\phi(P)$ and $S = r\hat{\phi}(Q)$.		
	$\xrightarrow{\text{Send } (R, S).}$	
		Let $(P, Q) = F(E, E')$.
		Check $R \in E'[(N, \pi - 1)]^o$.
		Check $S \in E[(N, \pi + 1)]^o$.
		Check $e_N(P, S) = e'_N(R, Q)$.

Notice that r is not used in verification

How to cheat?

1. $(P, Q) = F(E, E')$ is fixed (for fixed E and E').
2. The prover picks $P' \in E'[(N, \pi - 1)]$ and $Q' \in E[(N, \pi + 1)]$. This is done once and P' and Q' are fixed.
3. Let $\alpha = e_N(P, Q')$ and $\beta = e'_N(P', Q)$ and $x = \log_\alpha \beta$ then $\alpha^x = \beta$ and $e_N(P, xQ') = e'_N(P', Q)$.
4. Now if the prover sets $R = P'$ and $S = xQ'$ the verifier will be convinced $e_N(P, S) = e'_N(R, Q)$.
5. (aR, aS) is a valid response for any $a \in \mathbb{Z}_N$, so the malicious prover can generate many responses.

Sketch of a proof

- Prover:
  - choose random $P \in E[(N, \pi - 1)]$ and $r_1, r_2, r, r' \in \mathbb{Z}_N^*$.
  - Send $C_P = Com(P, r_1), C_{\phi(P)} = Com(\phi(P), r_2), rP, r'\phi(P)$
  - Send ZKPoK* of r and r'.
- Verifier:
  - check $rP \in E[(N, \pi - 1)] \wedge r'\phi(P) \in E'[(N, \pi - 1)]$ and proofs.
  - Choose a random bit $b \in \{0, 1\}$ and send it to the prover.
  - $b = 0$: Sample $Q \in E'[(N, \pi + 1)]$ and send it to Prover.
  - If $b = 1$: Sample $Q \in E[(N, \pi + 1)]$ and send it to Prover.
- Prover:
  - If $b = 0$ check $Q \in E'[(N, \pi + 1)]$ o.w. $Q \in E[(N, \pi + 1)]$.
  - If $b = 0$: Send $(P, r_1, r'\hat{\phi}(Q))$.
  - $b = 1$: Send $(\phi(P), r_2, \frac{r}{d}\phi(Q))$, where $d = \deg \phi$.
- Verifier:
  - $b = 0$: Check $Com(P, r_1) = C_P$, $e_N(P, r'\hat{\phi}(Q)) = e'_N(r'\phi(P), Q)$.
  - $b = 1$: $Com(\phi(P), r_2) = C_{\phi(P)}$, $e_N(rP, Q) = e'_N(\phi(P), \frac{r}{d}\phi(Q))$.

Is this enough?

- Not proving knowledge of isogeny, but knowledge of action on N-torsion and degree mod N.
- $e_N(P, \deg(\phi)Q) = e'_N(\phi(P), \phi(Q))$.
- Knowledge of degree is important, because knowledge of action is trivial.
- To recover an isogeny from torsion information we need $\sqrt{\deg(\phi)}$ points.
- $N \simeq 2^{2\lambda} << 2^{\lambda^3/2} \simeq \sqrt{\deg(\phi)}$
- Maybe less points are sufficient for oriented isogenies.

Remarks

- Basso et al. gave a proof of isogeny knowledge based on SIDH squares.
- Their proof reveals the degree of the secret isogeny. No problem over $\mathbb{F}_{p^2}$.
- For CSIDH, $|e_i|$ can be computed. For CSIDH-512 there are 2^{74} possibilities.
- Use SeaSign, but it has very high computation time.
- CSI-FiSh requires the class group to be known which breaks sequentiality.
- Jao and Mokrani propose an interactive proof.
- Their proof has linear interaction in the number of participants.
- (Quantum) CGL-SNARG based VDF
- (Quantum) High degree isogenies and Kani's criterion

Questions