Cryptanalysis of an Oblivious PRF from Supersingular Isogenies

Andrea Basso, Péter Kutas, Simon-Philipp Merz, Christophe Petit and Antonio Sanso

ROYAL HOLLOWAY UNIVERSITY OF LONDON

March 2022

CWI Student Seminar

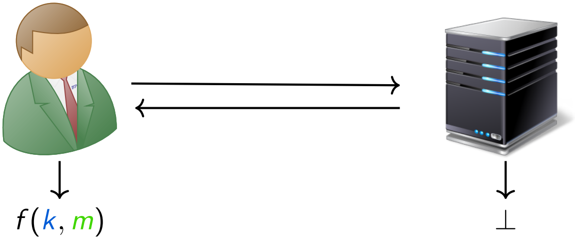
Content

- Definition of (V)OPRFs
- Applications
  - OPAQUE
  - PrivacyPass
- Isogenies and SIDH
- OPRF from isogenies
- Cryptanalytic results
  - Polytime and subexponential attacks
  - Requirement for trusted setup

Oblivious Pseudorandom Function (OPRF)

An OPRF is a two-party protocol to evaluate a PRF $f(k, m)$ where:

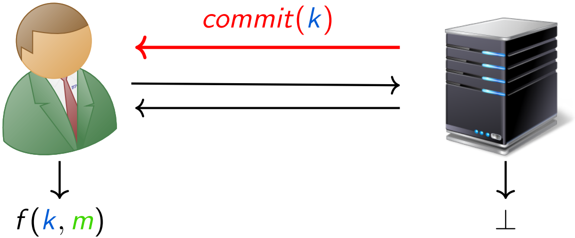
- The client learns $f(k, m)$, one evaluation of a PRF on a chosen input
- The server learns nothing about m



Oblivious Pseudorandom Function (OPRF)

An OPRF is a two-party protocol to evaluate a PRF $f(k, m)$ where:

- The client learns $f(k, m)$, one evaluation of a PRF on a chosen input
- The server learns nothing about m



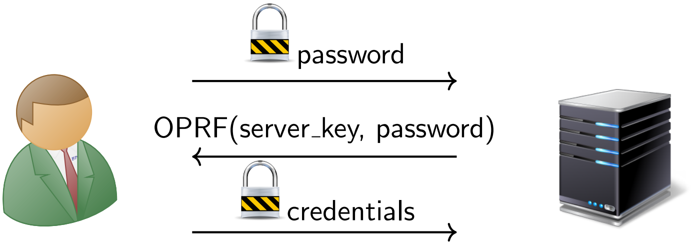
- An OPRF is called *verifable*, if the server proves to the client that output was computed using the key k

OPAQUE: OPRF + PAKE

- Use passwords that never leave your device

How to check a password that you have never seen?

Registration Phase:

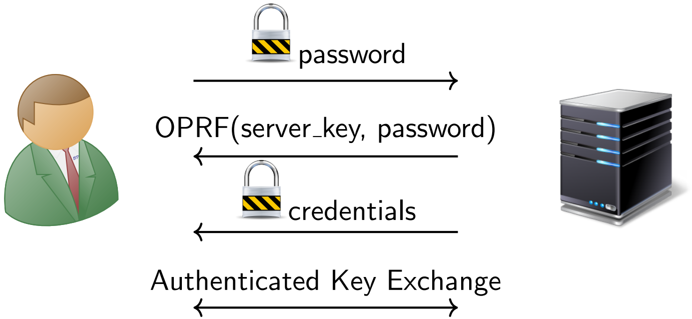


OPAQUE: OPRF + PAKE

- Use passwords that never leave your device

How to check a password that you have never seen?

Login Phase:



PrivacyPass

- Generate cryptographically ‘blinded’ tokens that can be signed by server after client authenticates themselves (e.g. CAPTCHA solution)
- Security properties:
  1. Unlinkability
  2. Unforgeability
- Construction:
  - VOPRF for issuance of tokens during blind signing phase
  - Verification of anonymous tokens during redemption phase

Existing Constructions

Parameters: group $\mathbb{G}$ of order q, hash functions H_1, H_2 onto $\mathbb{G}$ and $\{0,1\}^\ell$ resp.

Client $C(m)$		Server $S(k)$
Pick $r \leftarrow_R \mathbb{Z}_q$ Set $a \leftarrow (H_1(m))^r$	$\xrightarrow{a}$	
		If $a \in \mathbb{G}$, set $b \leftarrow a^k$
	$\xleftarrow{b}$	
If $b \in \mathbb{G}$, set $v \leftarrow b^{1/r}$ Output $H_2(m, v)$		

Existing Constructions

Parameters: group $\mathbb{G}$ of order q, hash functions H_1, H_2 onto $\mathbb{G}$ and $\{0,1\}^\ell$ resp.

Client $C(m)$		Server $S(k)$
Pick $r \leftarrow_R \mathbb{Z}_q$ Set $a \leftarrow (H_1(m))^r$	$\xrightarrow{a}$	
		If $a \in \mathbb{G}$, set $b \leftarrow a^k$
	$\xleftarrow{b}$	
If $b \in \mathbb{G}$, set $v \leftarrow b^{1/r}$ Output $H_2(m, v)$		

Post-quantum OPRF:

- Construction from lattices [ADDS19]
- Construction from isogenies [BKW20]

Isogenies

Definition

Let E, E' be two elliptic curves, and let $\varphi : E \to E'$ be a map between them. φ is called an *isogeny*, if

- φ is a surjective group homomorphism
- φ is a group homomorphism with finite kernel
- φ is a non-constant rational map with $\varphi(\mathcal{O}_E) = \mathcal{O}_{E'}$

Isogenies

Definition

Let E, E' be two elliptic curves, and let $\varphi : E \to E'$ be a map between them. φ is called an *isogeny*, if

- φ is a surjective group homomorphism
- φ is a group homomorphism with finite kernel
- φ is a non-constant rational map with $\varphi(\mathcal{O}_E) = \mathcal{O}_{E'}$

- For any finite subgroup $H \subset E$, there exists an isogeny $\varphi : E \to E' := E/H$ with kernel H
- For (separable) isogenies, $\# \ker(\varphi)$ is the degree of φ

Factoring isogenies

Definition (Universal property)

Let $\varphi : E \to E'$ be an isogeny. If $P \in \ker(\varphi)$, then there exist isogenies ψ, ϕ such that $\ker(\psi) = \langle P \rangle$ and

$$\varphi = \phi \circ \psi$$
$$\text{with } \deg(\varphi) = \deg(\phi) \cdot \deg(\psi)$$

Factoring isogenies

Definition (Universal property)

Let $\varphi : E \to E'$ be an isogeny. If $P \in \ker(\varphi)$, then there exist isogenies ψ, ϕ such that $\ker(\psi) = \langle P \rangle$ and

$$\varphi = \phi \circ \psi$$
$$\text{with } \deg(\varphi) = \deg(\phi) \cdot \deg(\psi)$$

- Factorisation is unique up to composition with isomorphisms
- Two elliptic curves are isomorphic if and only if they have the same j-invariant

Supersingular isogeny graphs

Definition (ℓ-isogeny graph)

The supersingular ℓ-isogeny graph over $\mathbb{F}_{p^2}$ consists of

- vertices are j-invariants of supersingular elliptic curves defined over $\mathbb{F}_{p^2}$
- edges between j and j' correspond to an ℓ-isogeny between two elliptic curves with j-invariants j and j'.

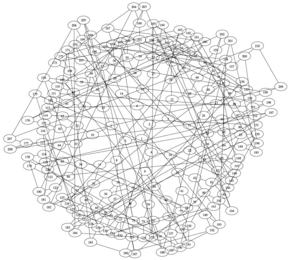


Figure: Image by D. Charles

Supersingular isogeny graphs

Definition (ℓ-isogeny graph)

The supersingular ℓ-isogeny graph over $\mathbb{F}_{p^2}$ consists of

- vertices are j-invariants of supersingular elliptic curves defined over $\mathbb{F}_{p^2}$
- edges between j and j' correspond to an ℓ-isogeny between two elliptic curves with j-invariants j and j'.

- connected, $\ell + 1$-regular graph

Figure: Image by D. Charles

Supersingular isogeny graphs

Definition (ℓ-isogeny graph)

The supersingular ℓ-isogeny graph over $\mathbb{F}_{p^2}$ consists of

- vertices are j-invariants of supersingular elliptic curves defined over $\mathbb{F}_{p^2}$
- edges between j and j' correspond to an ℓ-isogeny between two elliptic curves with j-invariants j and j'.

- connected, $\ell + 1$-regular graph
- graph has $\approx p/12$ vertices

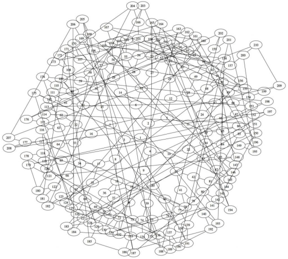


Figure: Image by D. Charles

Supersingular isogeny graphs

Definition (ℓ-isogeny graph)

The supersingular ℓ-isogeny graph over $\mathbb{F}_{p^2}$ consists of

- vertices are j-invariants of supersingular elliptic curves defined over $\mathbb{F}_{p^2}$
- edges between j and j' correspond to an ℓ-isogeny between two elliptic curves with j-invariants j and j'.

- connected, $\ell + 1$-regular graph
- graph has $\approx p/12$ vertices
- expander property: random walk of $\log(p)$ steps is almost as good as uniformly sampling the vertices

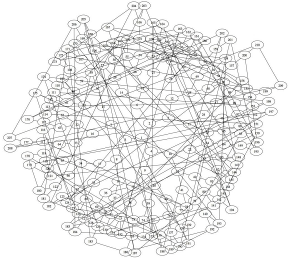


Figure: Image by D. Charles

Supersingular isogeny graphs

Definition (ℓ-isogeny graph)

The supersingular ℓ-isogeny graph over $\mathbb{F}_{p^2}$ consists of

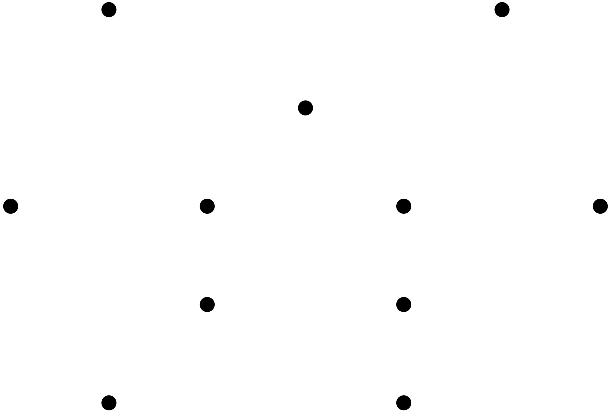
- vertices are j-invariants of supersingular elliptic curves defined over $\mathbb{F}_{p^2}$
- edges between j and j' correspond to an ℓ-isogeny between two elliptic curves with j-invariants j and j'.

- connected, $\ell + 1$-regular graph
- graph has $\approx p/12$ vertices
- expander property: random walk of $\log(p)$ steps is almost as good as uniformly sampling the vertices
- path finding is postulated to be exponentially hard both classically and quantumly

Figure: Image by D. Charles

SIDH [JD11]

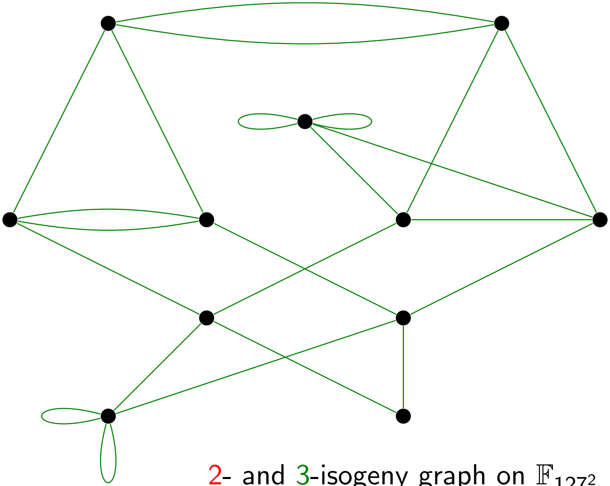
Idea: Alice and Bob walk in two *different* isogeny graphs on the *same* vertex set.



2- and 3-isogeny graph on $\mathbb{F}_{127^2}$

SIDH [JD11]

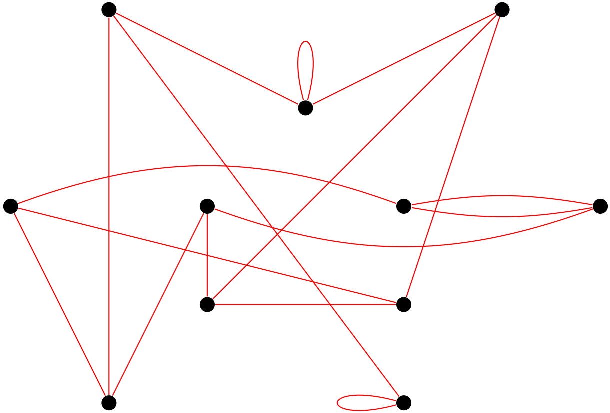
Idea: Alice and Bob walk in two *different* isogeny graphs on the *same* vertex set.



2- and 3-isogeny graph on $\mathbb{F}_{127^2}$

SIDH [JD11]

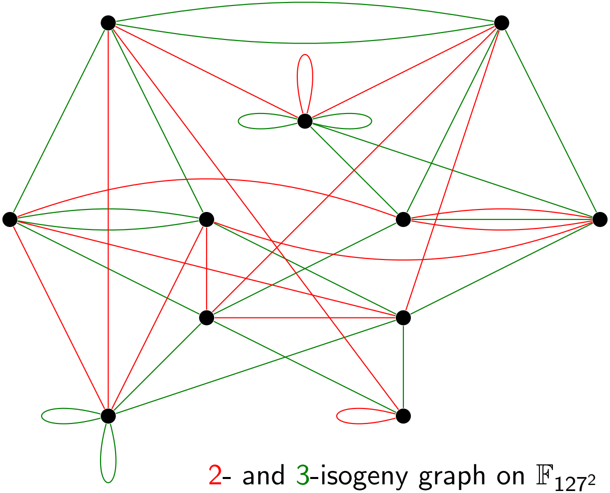
Idea: Alice and Bob walk in two *different* isogeny graphs on the *same* vertex set.



2- and 3-isogeny graph on $\mathbb{F}_{127^2}$

SIDH [JD11]

Idea: Alice and Bob walk in two *different* isogeny graphs on the *same* vertex set.



2- and 3-isogeny graph on $\mathbb{F}_{127^2}$

SIDH [JD11] (cont.)

- Fix a prime p such that $p = N_1 N_2 - 1$, $E_0/\mathbb{F}_p^2$ and bases $\langle P_A, Q_A \rangle = E_0[N_1]$, $\langle P_B, Q_B \rangle = E_0[N_2]$

$$
\begin{tikzcd}
 & E_0 \arrow[dl, "\varphi_A"'] \arrow[dr, "\varphi_B"] & \\
\begin{array}{c} E_A := E/\langle A \rangle \\ \varphi_A(P_B), \varphi_A(Q_B) \end{array} \arrow[dr, "\varphi'_B"'] & & \begin{array}{c} E_B := E/\langle B \rangle \\ \varphi_B(P_A), \varphi_B(Q_A) \end{array} \arrow[dl, "\varphi'_A"] \\
 & E_{AB} = E/\langle A, B \rangle &
\end{tikzcd}
$$

- Fix a prime p such that $p = N_1 N_2 - 1$, $E_0/\mathbb{F}_p^2$ and bases $\langle P_A, Q_A \rangle = E_0[N_1]$, $\langle P_B, Q_B \rangle = E_0[N_2]$

- Alice's secret is $A := P_A + [\mathrm{sk}_A]Q_A$
- Bob's secret is $B := P_B + [\mathrm{sk}_B]Q_B$

$$
\begin{tikzcd}
 & E_0 \arrow[dl, "\varphi_A"'] \arrow[dr, "\varphi_B"] & \\
\begin{array}{c} E_A := E/\langle A \rangle \\ \varphi_A(P_B), \varphi_A(Q_B) \end{array} \arrow[dr, "\varphi'_B"'] & & \begin{array}{c} E_B := E/\langle B \rangle \\ \varphi_B(P_A), \varphi_B(Q_A) \end{array} \arrow[dl, "\varphi'_A"] \\
 & E_{AB} = E/\langle A, B \rangle &
\end{tikzcd}
$$

SIDH [JD11] (cont.)

- Fix a prime p such that $p = N_1 N_2 - 1$, $E_0/\mathbb{F}_p^2$ and bases $\langle P_A, Q_A \rangle = E_0[N_1]$, $\langle P_B, Q_B \rangle = E_0[N_2]$

- Alice's secret is $A := P_A + [\mathrm{sk}_A]Q_A$
- Bob's secret is $B := P_B + [\mathrm{sk}_B]Q_B$

- Alice sends E_A, $\varphi_A(P_B)$, $\varphi_A(Q_B)$
- Bob sends E_B, $\varphi_B(P_A)$, $\varphi_B(Q_A)$

$$
\begin{tikzcd}
 & E_0 \arrow[dl, "\varphi_A"] \arrow[dr, "\varphi_B"] & \\
\begin{array}{c} E_A := E/\langle A \rangle \\ \varphi_A(P_B), \varphi_A(Q_B) \end{array} \arrow[dr, "\varphi'_B"] & & \begin{array}{c} E_B := E/\langle B \rangle \\ \varphi_B(P_A), \varphi_B(Q_A) \end{array} \arrow[dl, "\varphi'_A"] \\
 & E_{AB} = E/\langle A, B \rangle &
\end{tikzcd}
$$

SIDH [JD11] (cont.)

- Fix a prime p such that $p = N_1 N_2 - 1$, $E_0/\mathbb{F}_p^2$ and bases $\langle P_A, Q_A \rangle = E_0[N_1]$, $\langle P_B, Q_B \rangle = E_0[N_2]$

- Alice's secret is $A := P_A + [\mathrm{sk}_A]Q_A$
- Bob's secret is $B := P_B + [\mathrm{sk}_B]Q_B$

- Alice sends E_A, $\varphi_A(P_B)$, $\varphi_A(Q_B)$
- Bob sends E_B, $\varphi_B(P_A)$, $\varphi_B(Q_A)$

$$\begin{tikzcd}
& E_0 \arrow[dl, "\varphi_A"] \arrow[dr, "\varphi_B"] & \\
\begin{array}{c} E_A := E/\langle A \rangle \\ \varphi_A(P_B), \varphi_A(Q_B) \end{array} \arrow[dr, "\varphi'_B"] & & \begin{array}{c} E_B := E/\langle B \rangle \\ \varphi_B(P_A), \varphi_B(Q_A) \end{array} \arrow[dl, "\varphi'_A"] \\
& E_{AB} = E/\langle A, B \rangle &
\end{tikzcd}$$

- The shared secret is the j-invariant of E_{AB}

Oblivious Pseudorandom Functions from Isogenies [BKW20]

Client
Server

$$\begin{tikzcd} E_0 \arrow[r, "\phi_m"] \arrow[d] & E_M \arrow[d] \\ E_k \arrow[r] & E_{Mk} \end{tikzcd}$$

Oblivious Pseudorandom Functions from Isogenies [BKW20]

$$\begin{tikzcd} E_0 \arrow[r, "\phi_m"] & E_M \\ E_k & E_{Mk} \end{tikzcd}$$

Client
Server

Oblivious Pseudorandom Functions from Isogenies [BKW20]

$$\begin{array}{ccccc} E_0 & \xrightarrow{\phi_m} & E_M & & \\ & & & \searrow^{\phi_r} & \\ & & & & E_{Mr} \\ E_k & & E_{Mk} & & \end{array}$$

Client
Server

Oblivious Pseudorandom Functions from Isogenies [BKW20]

$$
\begin{tikzcd}
E_0 \arrow[r, "\phi_m"] & E_M \arrow[dr, "\phi_r"] & \\
 & & E_{Mr} \arrow[dd, "\phi_k"] \\
E_k & E_{Mk} & \\
 & & E_{Mrk}
\end{tikzcd}
$$

Client
Server

Oblivious Pseudorandom Functions from Isogenies [BKW20]

$$\begin{tikzcd}
E_0 \arrow[r, "\phi_m"] & E_M \arrow[dr, "\phi_r"] & \\
 & & E_{Mr} \arrow[dd, "\phi_k"] \\
E_k & E_{Mk} & \\
 & & E_{Mrk} \arrow[ul, "\hat{\phi}'_r"]
\end{tikzcd}$$

Client
Server

Oblivious Pseudorandom Functions from Isogenies [BKW20]

Client
Server

$$
\begin{tikzcd}
E_0 \arrow[r, "\phi_m"] \arrow[ddr] & E_M \arrow[dr, "\phi_r"] & \\
 & & E_{Mr} \arrow[dd, "\phi_k"] \\
E_k & E_{Mk} & \\
 & & E_{Mrk} \arrow[ul, "\hat{\phi}'_r"]
\end{tikzcd}
$$

$$f(k, m) = H(m, j(E_{Mk}), \mathrm{pk})$$

Pseudorandomness of an Oblivious PRF

- An attacker should not be able to evaluate the OPRF without the server's help even after multiple queries

Pseudorandomness of an Oblivious PRF

- An attacker should not be able to evaluate the OPRF without the server's help even after multiple queries

$$\begin{tikzcd} E_0 \arrow[rr] & & E_{M_0} \arrow[d] \\ & & E_{M_0 k} \end{tikzcd}$$

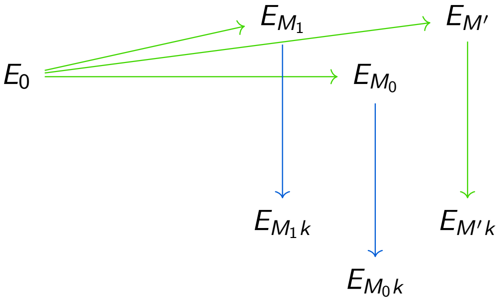
Pseudorandomness of an Oblivious PRF

- An attacker should not be able to evaluate the OPRF without the server's help even after multiple queries

$$\begin{tikzcd}
 & E_{M_1} \arrow[dd] & \\
E_0 \arrow[ur] \arrow[rr] & & E_{M_0} \arrow[dd] \\
 & E_{M_1 k} & \\
 & & E_{M_0 k}
\end{tikzcd}$$

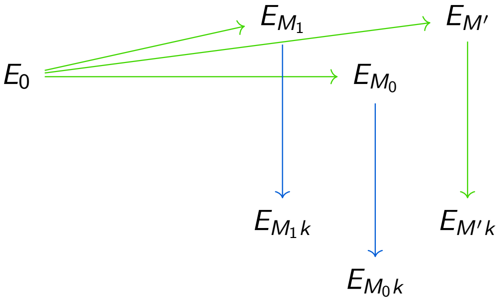
Pseudorandomness of an Oblivious PRF

- An attacker should not be able to evaluate the OPRF without the server's help even after multiple queries



Pseudorandomness of an Oblivious PRF

- An attacker should not be able to evaluate the OPRF without the server's help even after multiple queries
- Pseudorandomness of [BKW20] is based on a new 'auxiliary one-more' assumption



Attacking the 'one-more' Assumption

$$
\begin{tikzcd}
E_0 \arrow[r, "\phi_m"] \arrow[dd, "\phi_k"] \arrow[ddr, bend left] & E_M \arrow[dr, "\phi_r"] \arrow[dd] & \\
& & E_{Mr} \arrow[dd, "\phi'_k"] \\
E_k \arrow[r] & E_{Mk} & \\
& & E_{Mrk} \arrow[ul, "\hat{\phi}'_r"]
\end{tikzcd}
$$

- Find E_k and $\langle \phi_k(M) \rangle$ for some point $M \in E_0[2^n]$

Attacking the 'one-more' Assumption

$$
\begin{tikzcd}
E_0 \arrow[r, "\phi_m"] \arrow[dd, "\phi_k"] \arrow[ddr] & E_M \arrow[dr, "\phi_r"] \arrow[dd] & \\
 & & E_{Mr} \arrow[dd, "\phi'_k"] \\
E_k \arrow[r] & E_{Mk} & \\
 & & E_{Mrk} \arrow[ul, "\hat{\phi}'_r"]
\end{tikzcd}
$$

- Find E_k and $\langle \phi_k(M) \rangle$ for some point $M \in E_0[2^n]$
- Combine multiple points to obtain $\phi_k(E_0[2^n])$ up to scalar multiplication

$$
\begin{tikzcd}
E_0 \arrow[r, "\phi_m"] \arrow[dd, "\phi_k"] \arrow[ddr, bend left] & E_M \arrow[dd] \arrow[dr, "\phi_r"] & \\
 & & E_{Mr} \arrow[dd, "\phi'_k"] \\
E_k \arrow[r] & E_{Mk} & \\
 & & E_{Mrk} \arrow[ul, "\hat{\phi}'_r"]
\end{tikzcd}
$$

- Find E_k and $\langle \phi_k(M) \rangle$ for some point $M \in E_0[2^n]$
- Combine multiple points to obtain $\phi_k(E_0[2^n])$ up to scalar multiplication
- Given point $P \in E_0[2^n]$, compute $\langle \phi_k(P) \rangle$ and finally $E_k / \langle \phi_k(P) \rangle = E_{Pk}$

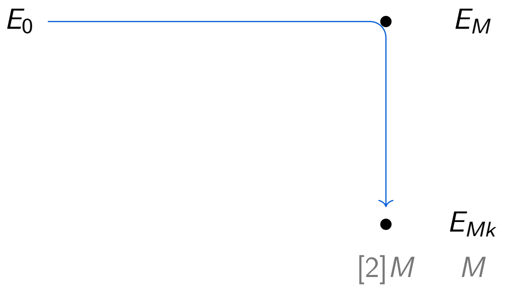
E_0

A Polytime Attack

Recovering points on E_k

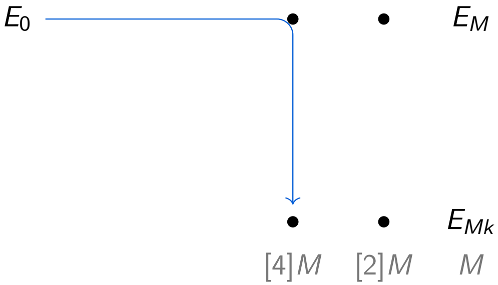
$$\begin{tikzcd} E_0 \arrow[r, dash] & E_M \arrow[d] \\ & E_{Mk} \\ & M \end{tikzcd}$$

A Polytime Attack
Recovering points on E_k



A Polytime Attack

Recovering points on E_k



A Polytime Attack

Recovering points on E_k

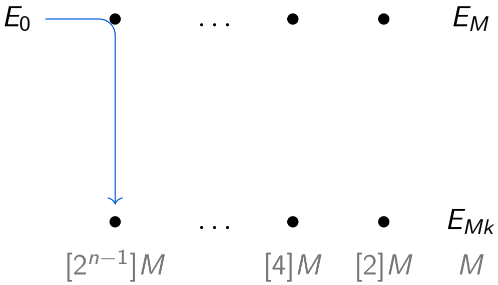
E_0 $\dots$ $\bullet$ $\bullet$ E_M

$\dots$ $\bullet$ $\bullet$ E_{Mk}

$[4]M$ $[2]M$ M

A Polytime Attack

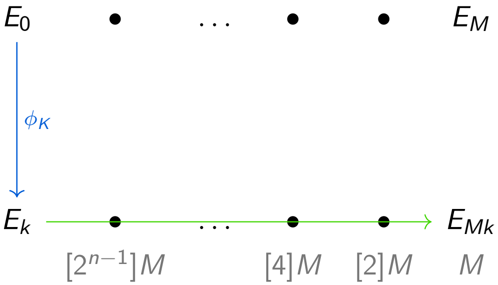
Recovering points on E_k



$$
\begin{tikzcd}
E_0 \arrow[d] & \bullet & \cdots & \bullet & \bullet & E_M \\
E_k & \bullet & \cdots & \bullet & \bullet & E_{Mk} \\
 & [2^{n-1}]M & & [4]M & [2]M & M
\end{tikzcd}
$$

$$
\begin{tikzcd}
E_0 \arrow[d] & \bullet & \cdots & \bullet & \bullet & E_M \\
E_k \arrow[rrrrr] & \bullet & \cdots & \bullet & \bullet & E_{Mk} \\
 & {[2^{n-1}]M} & & {[4]M} & {[2]M} & M
\end{tikzcd}
$$

A Polytime Attack
Recovering points on E_k



$$\ker \phi = \langle \phi_K(M) \rangle$$

A Polytime Attack

Combining the points

Given M on $E_0[2^n]$, we can recover $\langle \phi_K(M) \rangle$

A Polytime Attack

Combining the points

Given M on $E_0[2^n]$, we can recover $\langle \phi_K(M) \rangle \quad \Rightarrow \quad$ we can recover $[\alpha]\phi_K(M)$

A Polytime Attack

Combining the points

Given M on $E_0[2^n]$, we can recover $\langle \phi_K(M) \rangle \quad \Rightarrow \quad$ we can recover $[\alpha]\phi_K(M)$

We query on $M, N, M+N$ and obtain

$$\begin{aligned} M' &= [\alpha]\phi_K(M) \\ N' &= [\beta]\phi_K(N) \\ R' &= [\gamma]\phi_K(M+N) = [a]M' + [b]N' \end{aligned}$$

A Polytime Attack

Combining the points

Given M on $E_0[2^n]$, we can recover $\langle \phi_K(M) \rangle \quad \Rightarrow \quad$ we can recover $[\alpha]\phi_K(M)$

We query on $M, N, M+N$ and obtain

$$\left.\begin{aligned} M' &= [\alpha]\phi_K(M) \\ N' &= [\beta]\phi_K(N) \\ R' &= [\gamma]\phi_K(M+N) = [a]M' + [b]N' \end{aligned}\right\} \Rightarrow \frac{\alpha}{\beta} = \frac{b}{a}$$

A Polytime Attack

Combining the points

Given M on $E_0[2^n]$, we can recover $\langle \phi_K(M) \rangle \quad \Rightarrow \quad$ we can recover $[\alpha]\phi_K(M)$

We query on $M, N, M+N$ and obtain

$$\left.\begin{aligned} M' &= [\alpha]\phi_K(M) \\ N' &= [\beta]\phi_K(N) \\ R' &= [\gamma]\phi_K(M+N) = [a]M' + [b]N' \end{aligned}\right\} \Rightarrow \frac{\alpha}{\beta} = \frac{b}{a}$$

Breaking the assumption

Given any $P = [x]M + [y]N$, we can compute $\langle \phi_K(P) \rangle = \langle [x]M' + [y]\frac{\alpha}{\beta}N' \rangle$

A Polytime Attack
Results

- $O(\lambda)$ queries recover $\langle \phi_K(M) \rangle$ for any M in $E_0[2^n]$
- With three subgroups, we can compute $\langle \phi_K(P) \rangle$ for any P without further interactions
- This breaks the 'one-more' assumption

A Polytime Attack

Results

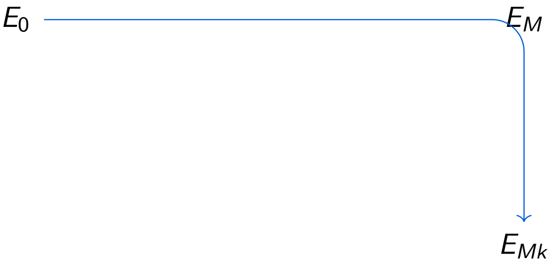
- $O(\lambda)$ queries recover $\langle \phi_K(M) \rangle$ for any M in $E_0[2^n]$
- With three subgroups, we can compute $\langle \phi_K(P) \rangle$ for any P without further interactions
- This breaks the 'one-more' assumption

But

- It is easy to check that query points have full order

A Subexponential Attack

Using full-order queries



A Subexponential Attack

Using full-order queries

E_0 $\qquad\qquad\qquad$ $\bullet \longleftarrow E_M$

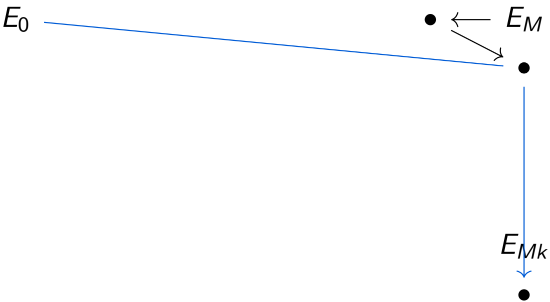
E_{Mk}

A Subexponential Attack

Using full-order queries

A Subexponential Attack

Using full-order queries

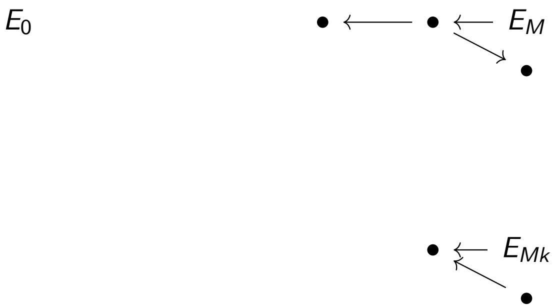


A Subexponential Attack

Using full-order queries

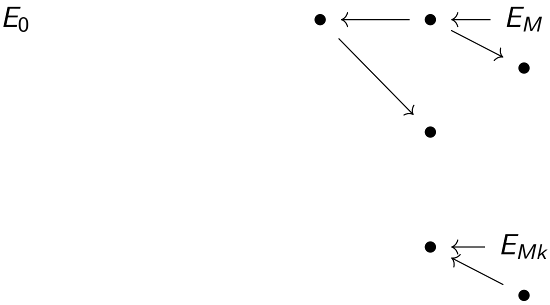
A Subexponential Attack

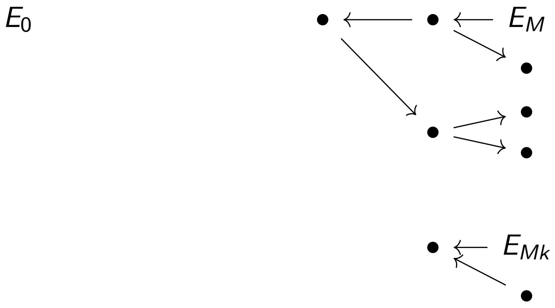
Using full-order queries



A Subexponential Attack

Using full-order queries

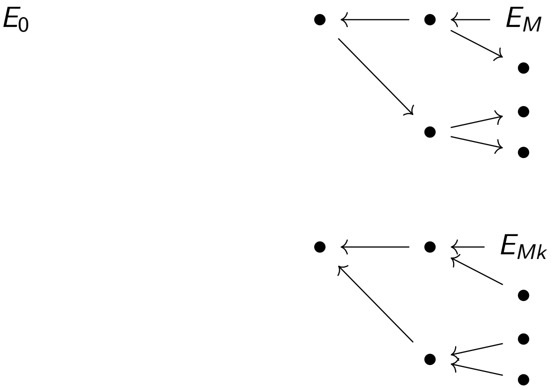




E_0
E_M
E_{Mk}

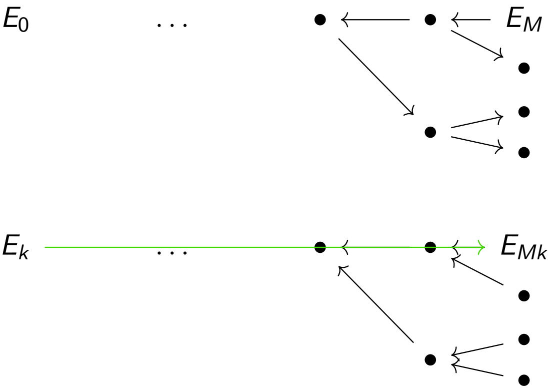
A Subexponential Attack

Using full-order queries



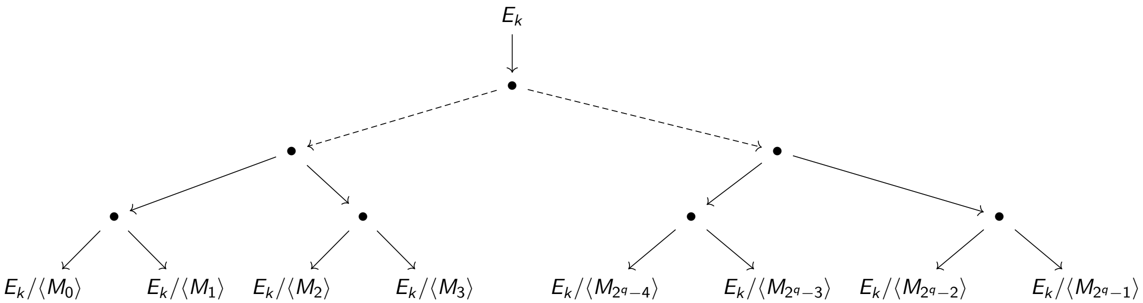
A Subexponential Attack

Using full-order queries



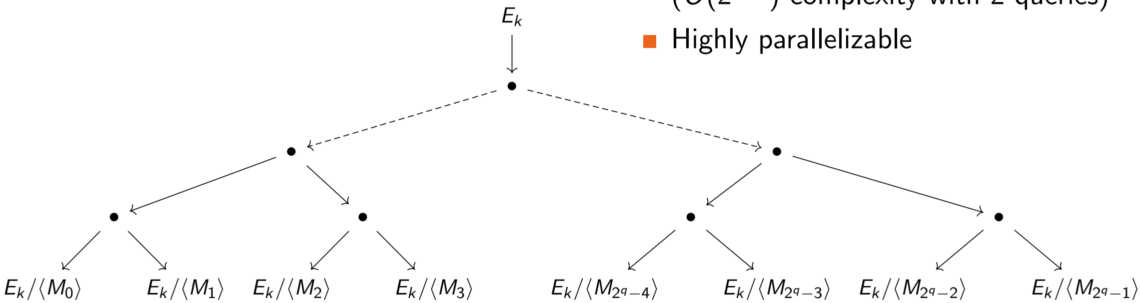
A Subexponential Attack

Building a tree



A Subexponential Attack

Building a tree



- Queries/complexity trade-offs ($O(2^{\lambda/3})$ complexity with 2 queries)
- Highly parallelizable

A Subexponential Attack

The full attack:

- Use the binary tree to recover points on E_k
- Second part of the attack same as polytime attack
- Subexponential complexity for balanced trade-offs

A Subexponential Attack

The full attack:

- Use the binary tree to recover points on E_k
- Second part of the attack same as polytime attack
- Subexponential complexity for balanced trade-offs

Countermeasures:

- No obvious countermeasures
- Increase the parameter size? $\Rightarrow$ very large degrees
- New efficient solutions?

Implementation Results

Parameters			MITM		Running Time
$\log p$	λ	q	Distance	Memory (kB)	(s)
112	8	3	8	3.5	15
216	16	6	10	13.8	212 (3.53 m)
413	32	8	16	211.4	1,371 (22.85 m)
859	67	11	26	14,073	163,869 (1.89 d)
1,614	128	18	40	3,384,803	*174,709,440 (5.54 y)*

Available at `https://github.com/isogenists/isogeny-OPRF`

The Starting Curve

Who chooses E_0?

- The client
- A third-party
- The server
- Known curve ($j(E_0) = 1728$)
- Trusted setup

The Starting Curve

Who chooses E_0?

- The client
- A third-party

$\left.\begin{array}{l}\text{The client}\\ \text{A third-party}\end{array}\right\}$ can backdoor E_0 $\Rightarrow$ key-recovery attack on the server

- The server
- Known curve ($j(E_0) = 1728$)
- Trusted setup

The Starting Curve

Who chooses E_0?

$$\left.\begin{array}{l}\blacksquare \text{ The client} \\ \blacksquare \text{ A third-party}\end{array}\right\} \text{can backdoor } E_0 \quad \Rightarrow \quad \text{key-recovery attack on the server}$$

$$\left.\begin{array}{l}\blacksquare \text{ The server} \\ \blacksquare \text{ Known curve } (j(E_0) = 1728)\end{array}\right\} \text{breaks the } \textit{Supersingular Isogeny Collision} \text{ assumption}$$

- Trusted setup

The Starting Curve

Who chooses E_0?

- The client
- A third-party

$\Big\}$ can backdoor E_0 $\Rightarrow$ key-recovery attack on the server

- The server
- Known curve ($j(E_0) = 1728$)

$\Big\}$ breaks the *Supersingular Isogeny Collision* assumption

- **Trusted setup**

Conclusion

- Two attacks on ‘one-more’ assumption and the pseudorandomness of Boneh et al.’s OPRF
- A proof of concept implementation of the attack
- Need for a trusted setup
- CSIDH-based OPRF construction is not affected by the attack

Paper available at `https://ia.cr/2021/706`