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

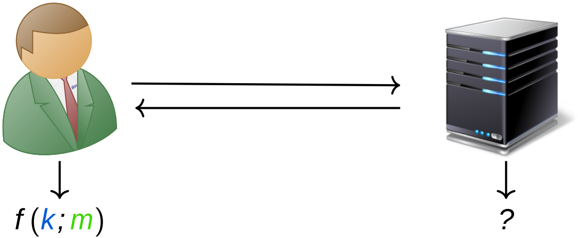
Content

- 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:

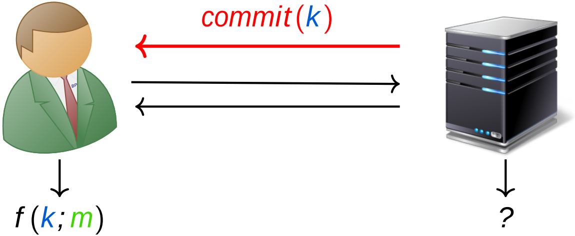
- The client learns $f(k; m)$, one evaluation of a PRF on a chosen input
- The server learns nothing about m



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



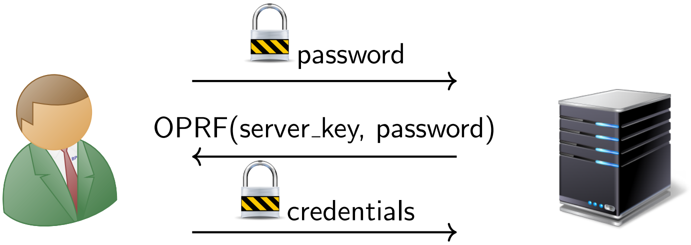
- An OPRF is called *verifable*, if the server proves to the client that output was computed using the key k

OPAQUE: OPRF + PAKE

- Use passwords that never leave your device

How to check a password that you have never seen?

Registration Phase:

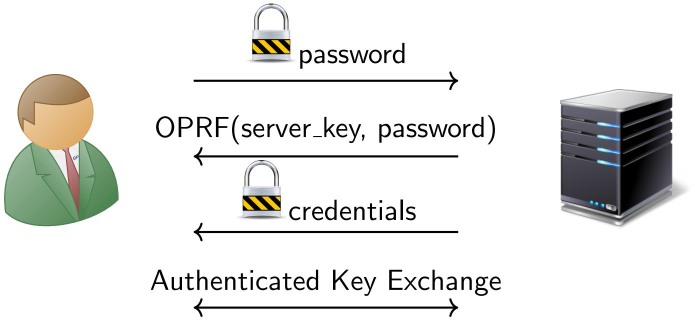


OPAQUE: OPRF + PAKE

- 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 $\varphi(O_E) = O_{E'}$

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 $\varphi(O_E) = O_{E'}$

- 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

De nition (Universal property)

Let ' : E ! E^0 be an isogeny. If P 2 ker('), then there exist isogenies ; such that ker( ) = hPi and

$$' =$$

$$\text{with } \deg(') = \deg(\ ) \ \deg(\ )$$

Definition (Universal property)

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

$$\varphi = $$

$$\text{with } \deg(\varphi) = \deg(\ )\ \deg(\ )$$

- 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

De nition (`-isogeny graph)

The supersingular `-isogeny graph over F_{p^2} consists of

- vertices are j-invariants of supersingular elliptic curves defined over F_{p^2}
- edges between j and j^0 correspond to an `-isogeny between two elliptic curves with j-invariants j and j^0.

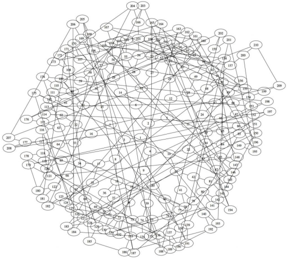


Figure: Image by D. Charles

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

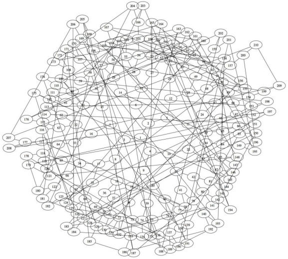


Figure: Image by D. Charles

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 $p/12$ vertices

Figure: Image by D. Charles

Supersingular isogeny graphs

De nition (`-isogeny graph)

The supersingular `-isogeny graph over F_{p^2} consists of

- vertices are j-invariants of supersingular elliptic curves defined over F_{p^2}
- edges between j and j^0 correspond to an `-isogeny between two elliptic curves with j-invariants j and j^0.

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

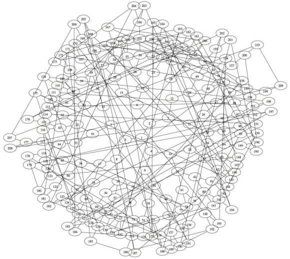


Figure: Image by D. Charles

Supersingular isogeny graphs

De nition (`-isogeny graph)

The supersingular `-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^0 correspond to an `-isogeny between two elliptic curves with j-invariants j and j^0.

- connected, ` + 1-regular graph
- graph has $p/12$ 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

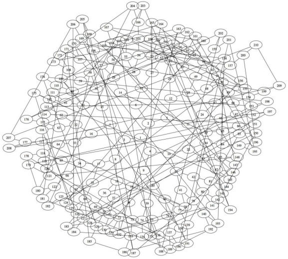
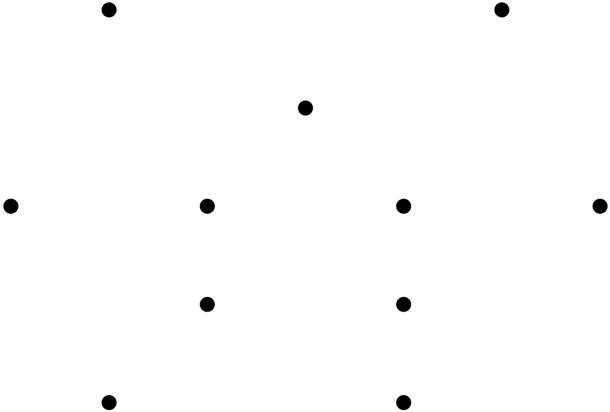


Figure: Image by D. Charles

SIDH [JD11]

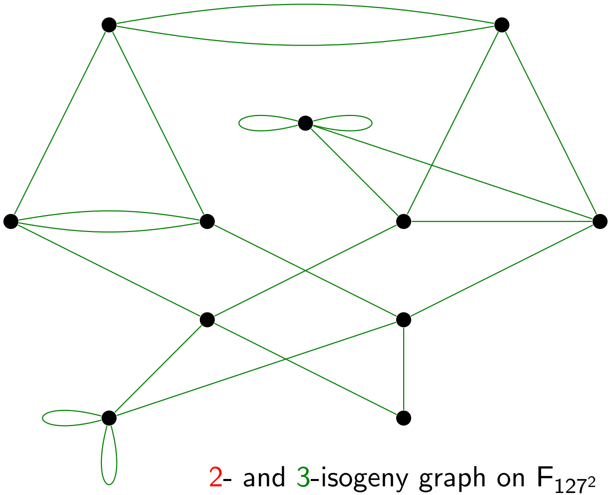
Idea: Alice and Bob walk in two *di erent* isogeny graphs on the *same* vertex set.



2- and 3-isogeny graph on F_{127^2}

SIDH [JD11]

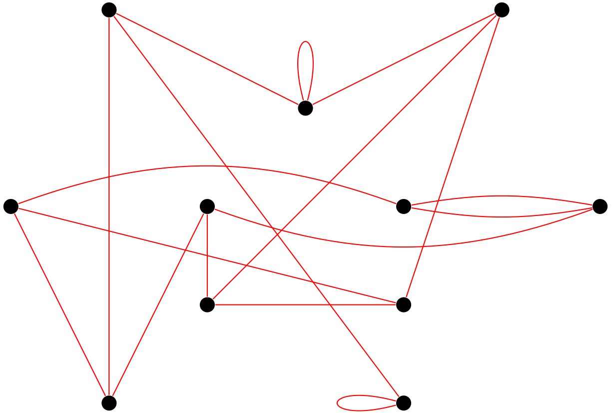
Idea: Alice and Bob walk in two *di erent* isogeny graphs on the *same* vertex set.



2- and 3-isogeny graph on F_{127^2}

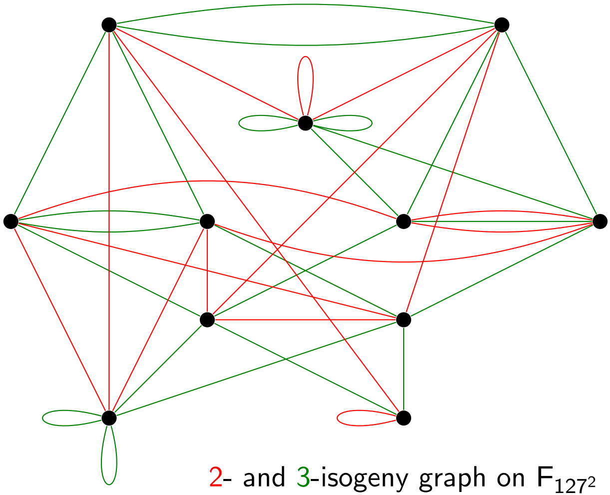
SIDH [JD11]

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



2- and 3-isogeny graph on F_{127^2}

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



2- and 3-isogeny graph on 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 $\langle P_A ; Q_A \rangle = E_0[N_1]$, $\langle P_B ; Q_B \rangle = E_0[N_2]$

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
  $A := P_A + [sk_A]Q_A$
- Bob's secret is
  $B := P_B + [sk_B]Q_B$

SIDH [JD11] (cont.)

- Fix a prime p such that $p = N_1 N_2 \pm 1$, $E_0 / \mathbb{F}_{p^2}$ 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
  $A := P_A + [sk_A]Q_A$
- Bob's secret is
  $B := P_B + [sk_B]Q_B$

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

SIDH [JD11] (cont.)

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

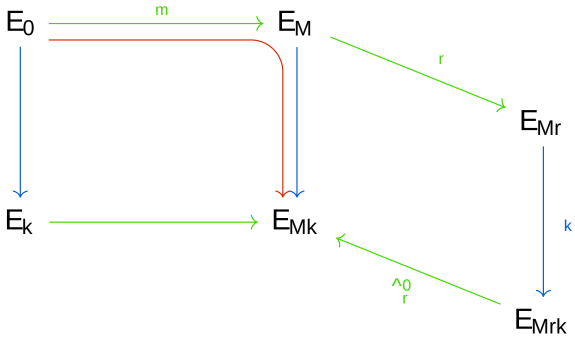
- Alice's secret is
  $A := P_A + [sk_A]Q_A$
- Bob's secret is
  $B := P_B + [sk_B]Q_B$

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

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

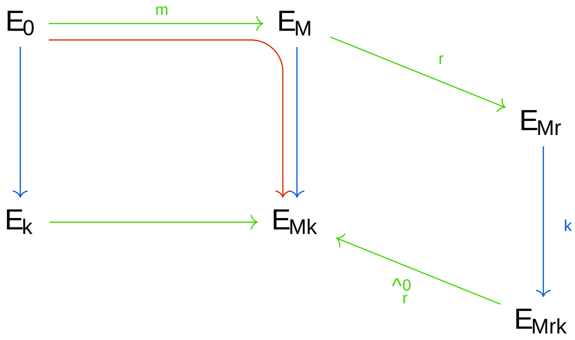
Oblivious Pseudorandom Functions from Isogenies [BKW20]

Client
Server



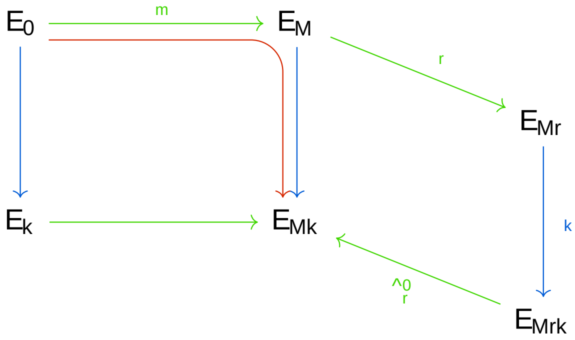
Oblivious Pseudorandom Functions from Isogenies [BKW20]

Client
Server



Oblivious Pseudorandom Functions from Isogenies [BKW20]

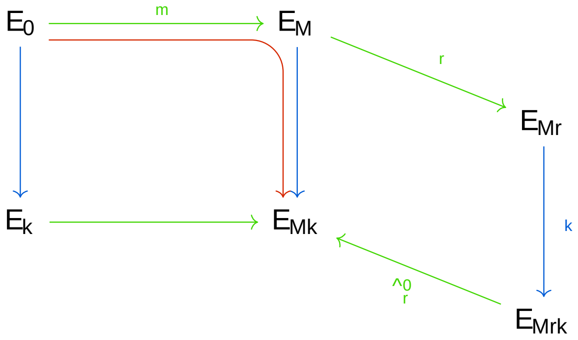
Client
Server



Oblivious Pseudorandom Functions from Isogenies [BKW20]

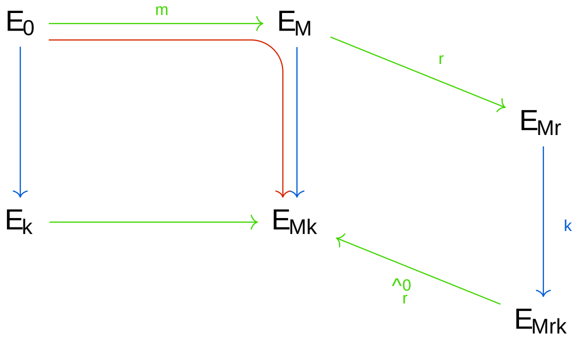
Client

Server

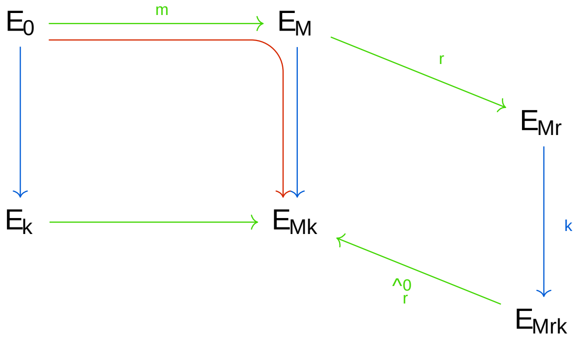


Oblivious Pseudorandom Functions from Isogenies [BKW20]

Client
Server

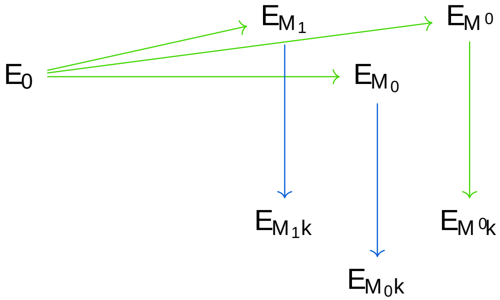


Client
Server

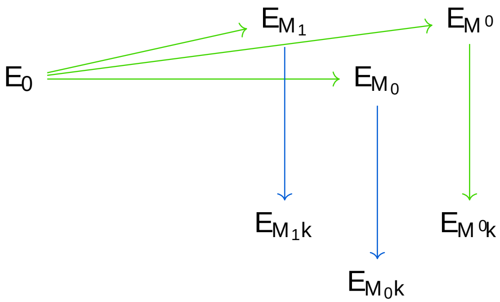


$$f(k; m) = H(m; j(E_{Mk}); pk)$$

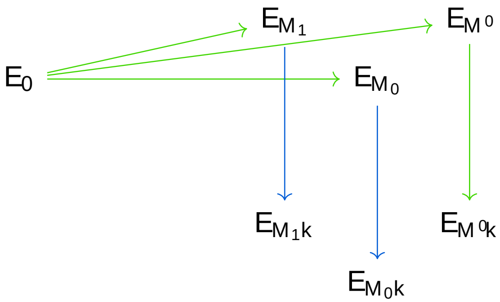
- An attacker should not be able to evaluate the OPRF without the server's help even after multiple queries



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

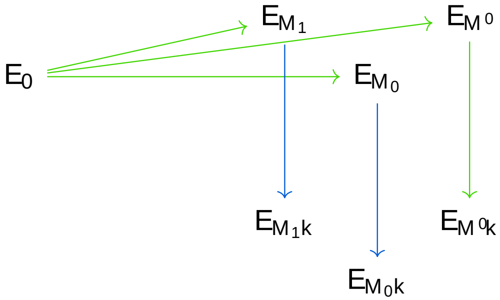


- An attacker should not be able to evaluate the OPRF without the server's help even after multiple queries



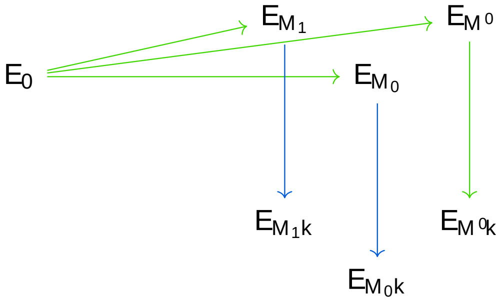
Pseudorandomness of an Oblivious PRF

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

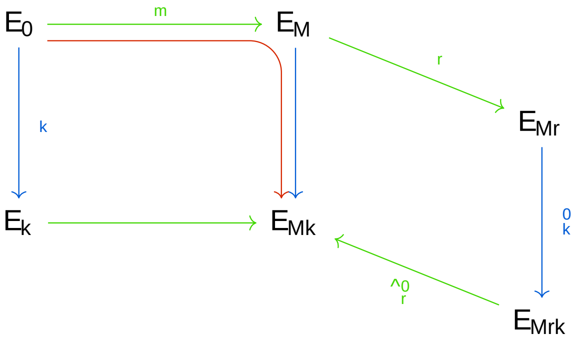


Pseudorandomness of an Oblivious PRF

- 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

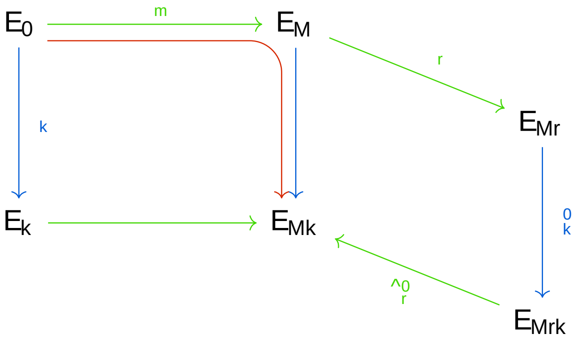


Attacking the `one-more' Assumption

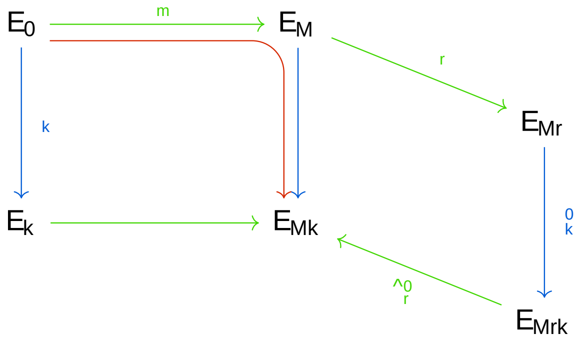


- Find E_k and $\langle \phi_k(M) \rangle$ for some point $M \in E_0[2^n]$

Attacking the `one-more' Assumption



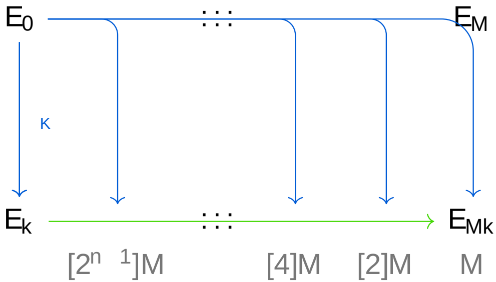
- Find E_k and h $_k$(M)i for some point M 2 $E_0[2^n]$
- Combine multiple points to obtain $_k(E_0[2^n])$ up to scalar multiplication



- Find E_k and h $_k$(M)i for some point M 2 $E_0[2^n]$
- Combine multiple points to obtain $_k(E_0[2^n])$ up to scalar multiplication
- Given point P 2 $E_0[2^n]$, compute h $_k$(P)i and nally E_k=h $_k$(P)i = E_{Pk}

A Polytime Attack

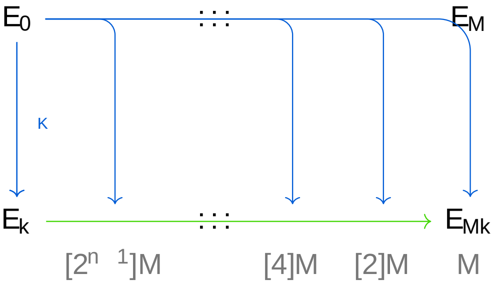
Recovering points on E_k



ker = h κ (M)i

A Polytime Attack

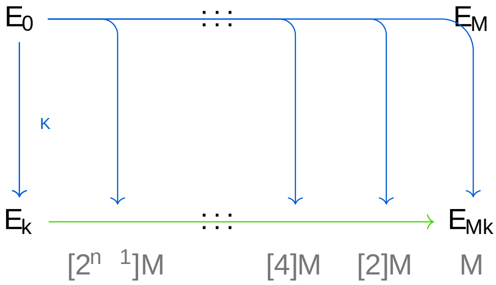
Recovering points on E_k



ker = h K (M)i

A Polytime Attack

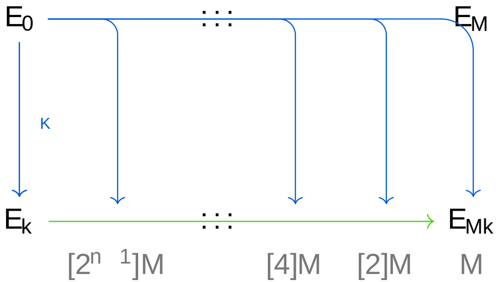
Recovering points on E_k



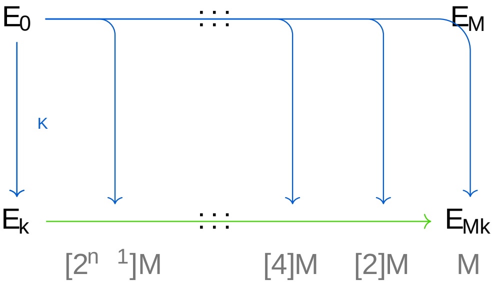
ker = h K (M)i

A Polytime Attack

Recovering points on E_k



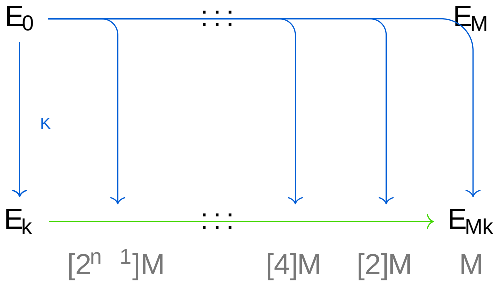
ker = h κ (M) i



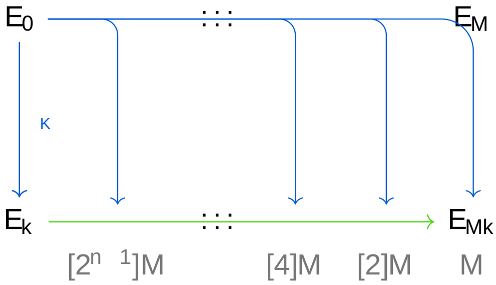
ker = h K (M)i

A Polytime Attack

Recovering points on E_k



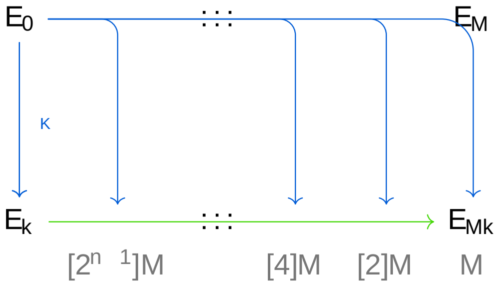
ker = h K (M)i



ker = h K (M)i

A Polytime Attack

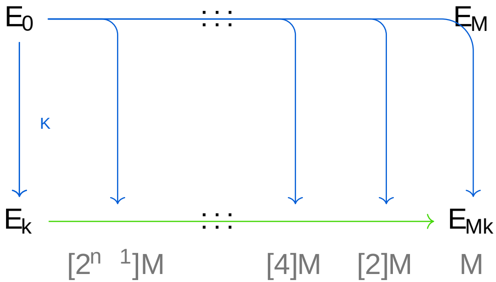
Recovering points on E_k



ker = h K (M)i

A Polytime Attack

Recovering points on E_k



ker = h K (M)i

A Polytime Attack

Combining the points

GivenM on $E_0[2^n]$, we can recoveh $_K$(M)i

A Polytime Attack

Combining the points

Given M on $E_0[2^n]$, we can recover $h_K(M)i$) we can recover [] $_K(M)$

A Polytime Attack

Combining the points

Given M on $E_0[2^n]$, we can recover $\langle \; {}_K(M) \rangle$) we can recover $[\,]\; {}_K(M)$

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

$$M' = [\quad]\; {}_K(M)$$
$$N' = [\quad]\; {}_K(N)$$
$$R' = [\quad]\; {}_K(M + N) = [a]M' + [b]N'$$

Given M on $E_0[2^n]$, we can recoveh $_K(M)$i) we can recover [] $_K(M)$

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

$$M^0 = [\ \ ]\ _K(M)$$
$$N^0 = [\ \ ]\ _K(N)$$
$$R^0 = [\ \ ]\ _K(M + N) = [a]M^0 + [b]N^0$$

9 ≥ > ,) — = $\frac{b}{a}$

A Polytime Attack

Combining the points

Given M on $E_0[2^n]$, we can recover $h\ _K(M)i$) we can recover $[\,]\ _K(M)$

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

$$
\begin{aligned}
M^0 &= [\quad]\ _K(M) \\
N^0 &= [\quad]\ _K(N) \\
R^0 &= [\quad]\ _K(M + N) = [a]M^0 + [b]N^0
\end{aligned}
\quad 9 \geq\ >,\quad) \quad - = \frac{b}{a}
$$

Breaking the assumption

Given any $P = [x]M + [y]N$, we can compute $h\ _K(P)i = h[x]M^0 + [y]\,-\,N^0 i$

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

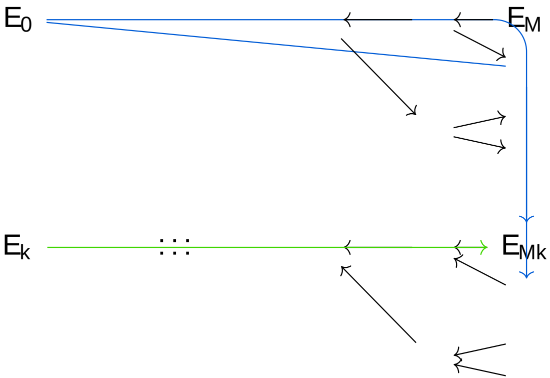
- O( ) queries recover $h_K(M)i$ for any M in $E_0[2^n]$
- With three subgroups, we can compute $h_K(P)i$ for any P without further interactions
- This breaks the `one-more' assumption

But

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

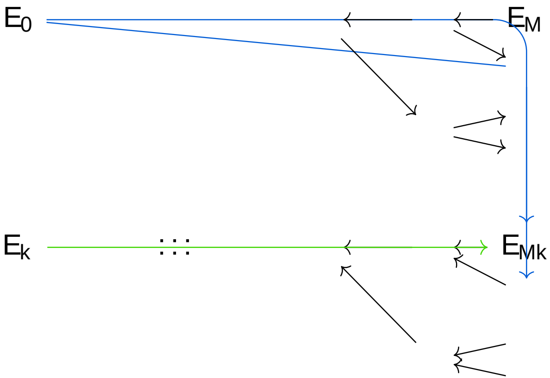
A Subexponential Attack

Using full-order queries



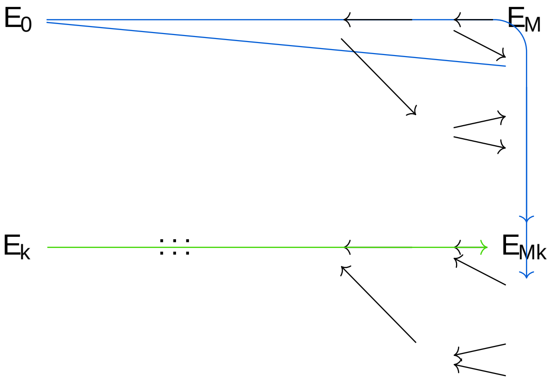
A Subexponential Attack

Using full-order queries



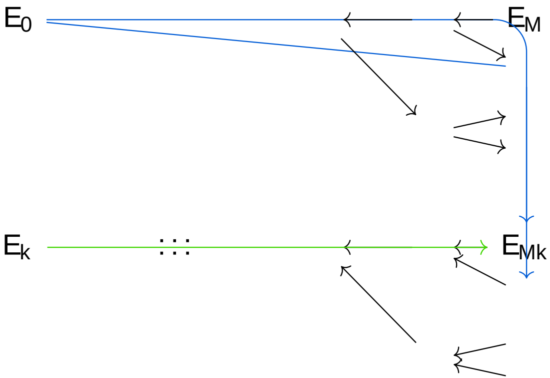
A Subexponential Attack

Using full-order queries



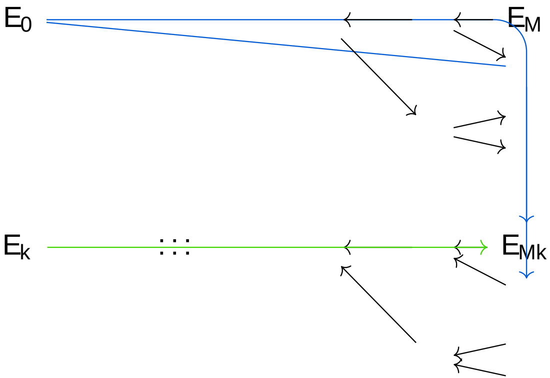
A Subexponential Attack

Using full-order queries

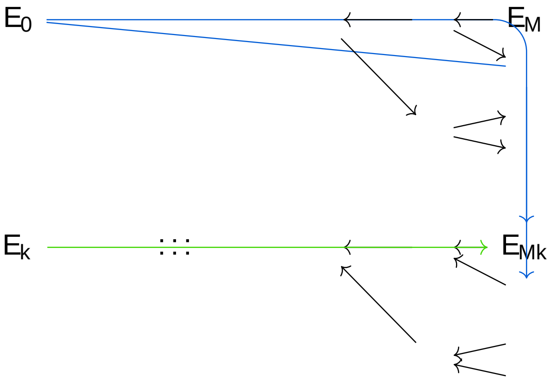


A Subexponential Attack

Using full-order queries

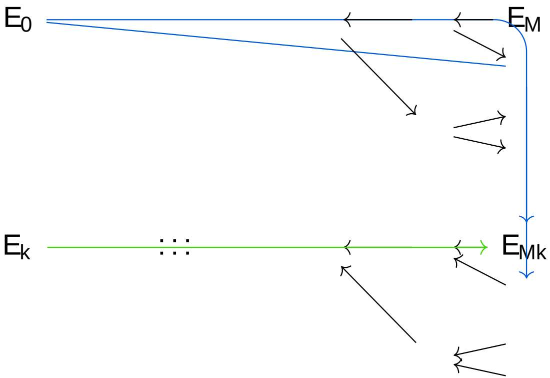


A Subexponential Attack
Using full-order queries



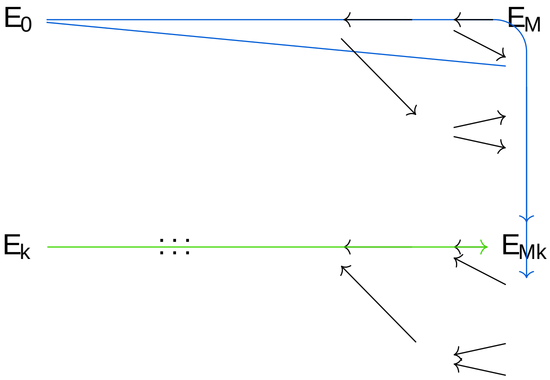
A Subexponential Attack

Using full-order queries



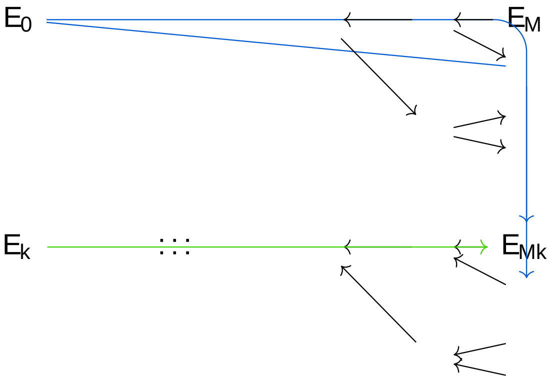
A Subexponential Attack

Using full-order queries



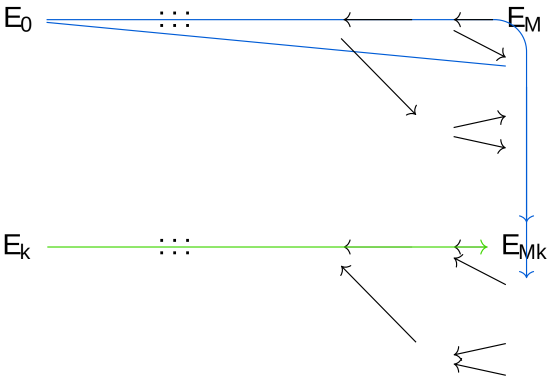
A Subexponential Attack

Using full-order queries



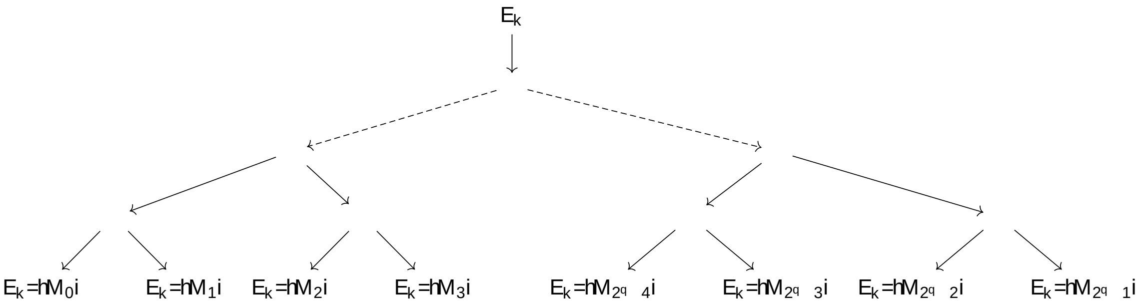
A Subexponential Attack

Using full-order queries



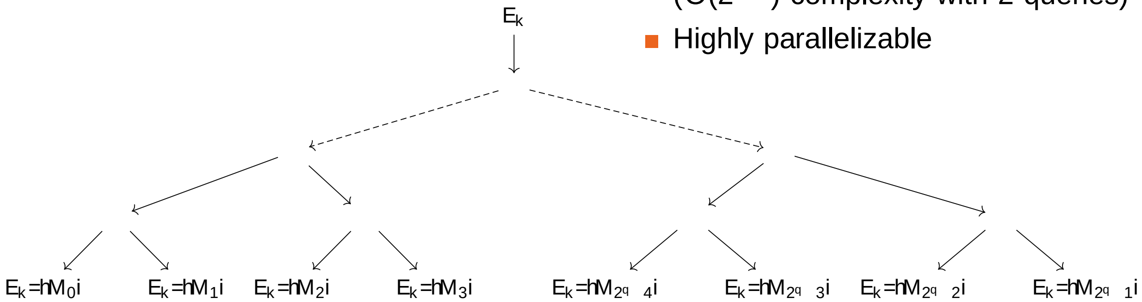
A Subexponential Attack

Building a tree



A Subexponential Attack

Building a tree



- Queries/complexity trade-o s ($O(2^{= 3})$ 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-o s

Countermeasures:

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

Implementation Results

Parameters			MITM		Running Time
log p		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 athttps://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

} can backdoor E_0) key-recovery attack on the server

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

Who chooses E_0?

- The client
- A third-party

} can backdoor E_0) key-recovery attack on the server

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

} breaks the *Supersingular Isogeny Collision* assumption

- Trusted setup

The Starting Curve

Who chooses E_0?

- The client
- A third-party

} can backdoor E_0) key-recovery attack on the server

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

} 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`