Asiacrypt 2021

Cryptanalysis of an Oblivious PRF from Supersingular Isogenies

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

Oblivious Pseudorandom Functions (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
- if the OPRF is *verifable*, the server always uses the same key k

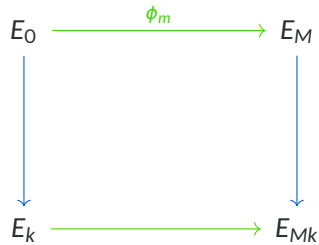
Applications:

- password-authenticated key exchanges,
- private-set interesection,
- privacy-preserving CAPTCHA

Oblivious Pseudorandom Functions from Isogenies [BKW20]

Client

Server



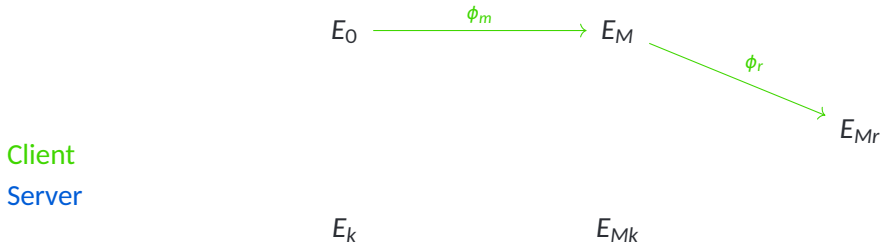
Oblivious Pseudorandom Functions from Isogenies [BKW20]

Client

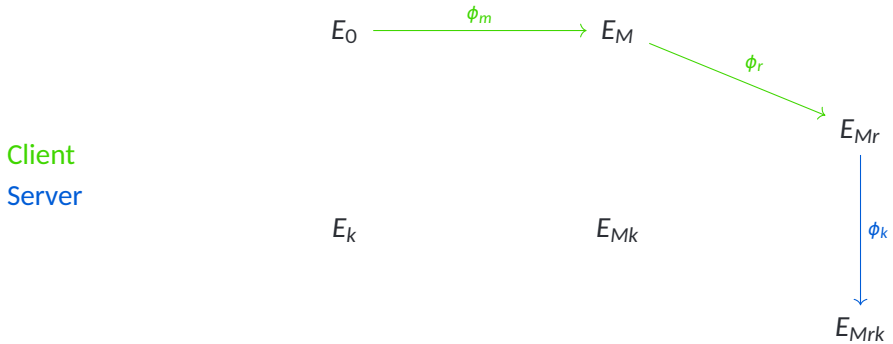
Server

E_k E_{Mk}

Oblivious Pseudorandom Functions from Isogenies [BKW20]



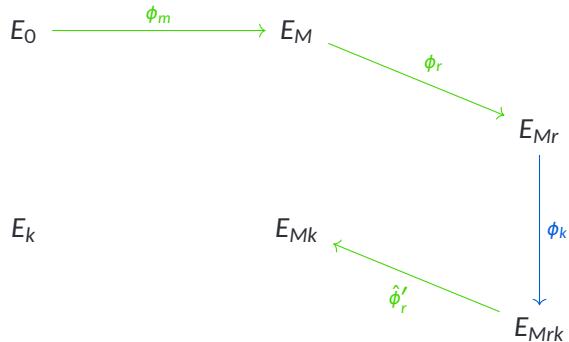
Oblivious Pseudorandom Functions from Isogenies [BKW20]



Oblivious Pseudorandom Functions from Isogenies [BKW20]

Client

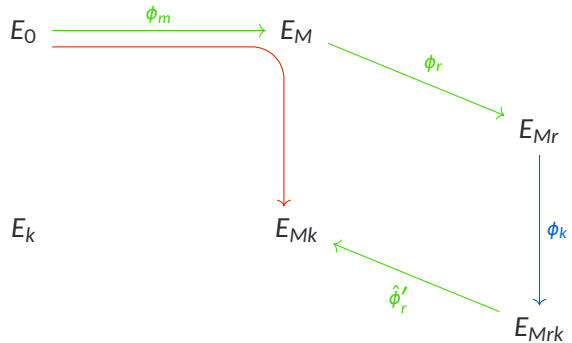
Server



Oblivious Pseudorandom Functions from Isogenies [BKW20]

Client

Server



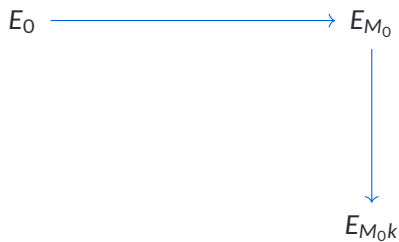
$$f(k, m) = H(m, j(E_{Mk}), \text{pk})$$

Pseudorandomness of an Oblivious PRF

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

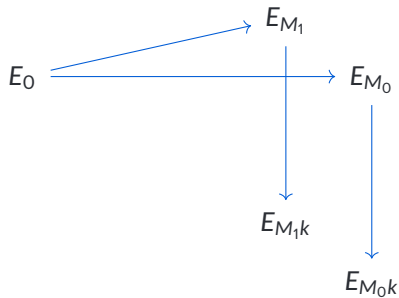
Pseudorandomness of an Oblivious PRF

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



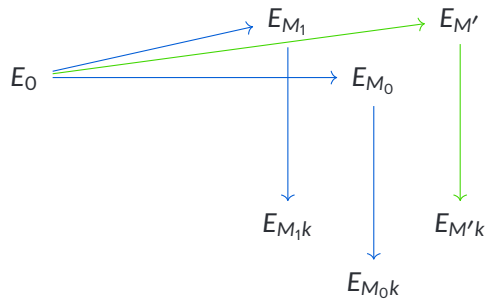
Pseudorandomness of an Oblivious PRF

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



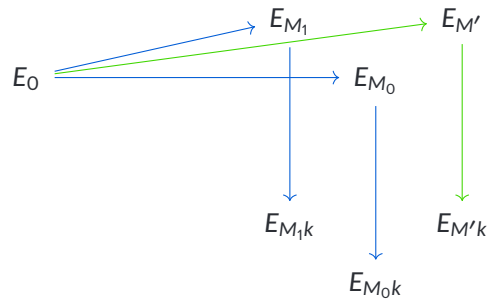
Pseudorandomness of an Oblivious PRF

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

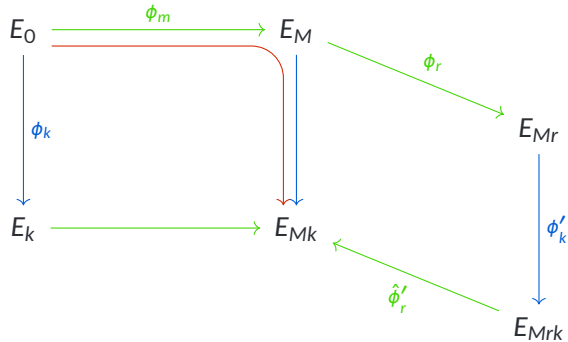


Pseudorandomness of an Oblivious PRF

- 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 $M \in E_0[2^n]$
- Combine multiple points to obtain $\phi_k(E_0[2^n])$

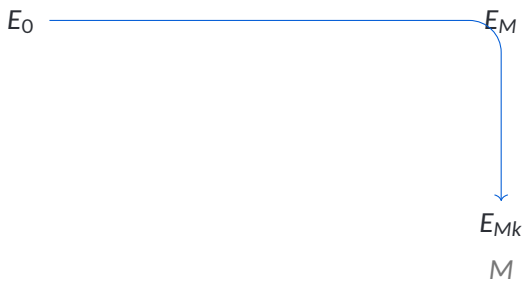
A Polytime Attack

Recovering points on E_k

E_0

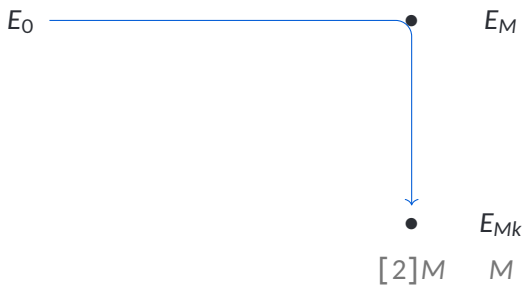
A Polytime Attack

Recovering points on E_k



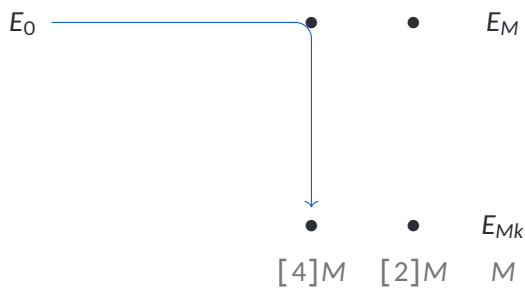
A Polytime Attack

Recovering points on E_k



A Polytime Attack

Recovering points on E_k

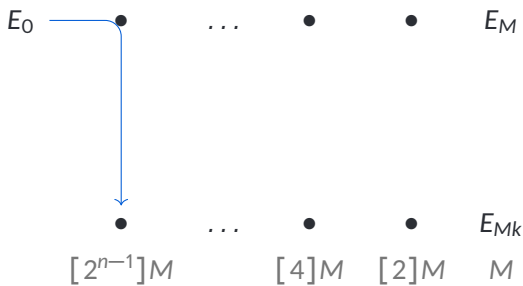


A Polytime Attack

Recovering points on E_k

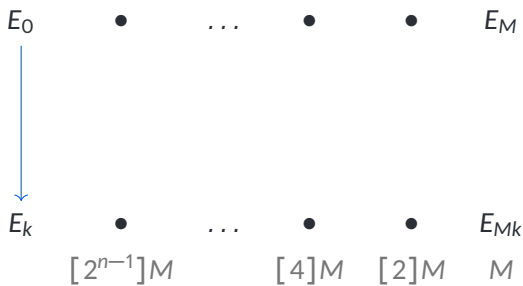
A Polytime Attack

Recovering points on E_k



A Polytime Attack

Recovering points on E_k



A Polytime Attack

Recovering points on E_k

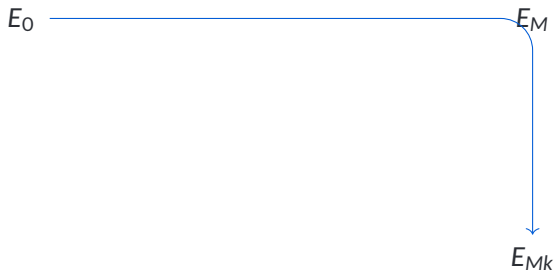
A Polytime Attack

Recovering points on E_k

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

A Subexponential Attack

Using full-order queries



A Subexponential Attack

Using full-order queries

E_0

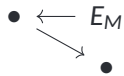
$\bullet \longleftarrow E_M$

E_{Mk}

A Subexponential Attack

Using full-order queries

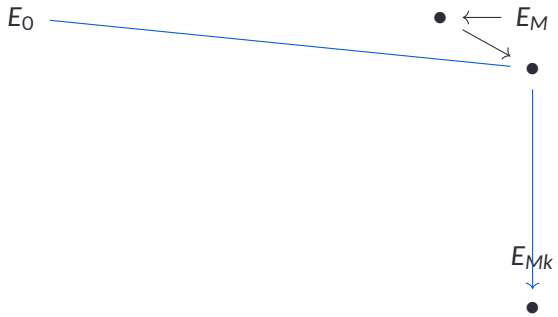
E_0



E_{Mk}

A Subexponential Attack

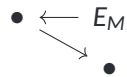
Using full-order queries



A Subexponential Attack

Using full-order queries

E_0



A Subexponential Attack

Using full-order queries

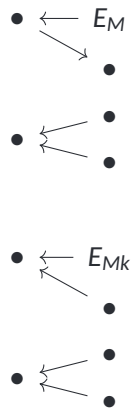
A Subexponential Attack

Using full-order queries

A Subexponential Attack

Using full-order queries

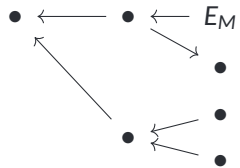
E_0

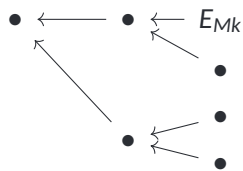


A Subexponential Attack

Using full-order queries

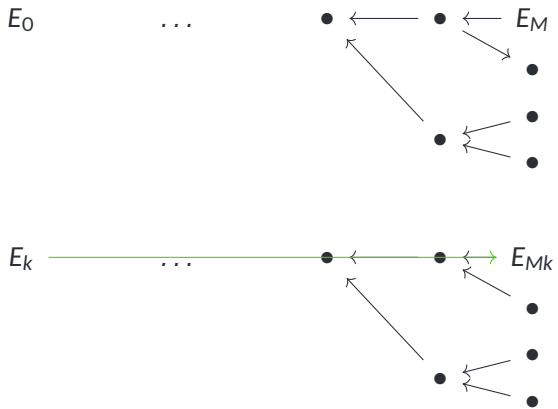
E_0





A Subexponential Attack

Using full-order queries



Conclusion

- Two attacks on the ‘one-more’ assumption and the pseudorandomness of the OPRF
- A proof of concept implementation of the attack
- Need for a trusted setup

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

References I

[BKW20] Dan Boneh, Dmitry Kogan, and Katharine Woo.
Oblivious pseudorandom functions from isogenies.
In *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part II*, pages 520–550, 2020.