Round-optimal Verifiable Oblivious Pseudorandom Functions from Ideal Lattices

Martin R. Albrecht

joint work with Alex Davidson, Amit Deo and Nigel Smart.

Outline

Motivation

A VOPRF from Lattices

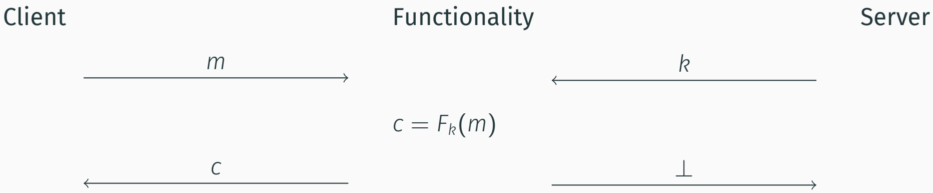
Security

Parameters

Alternative Constructions

Motivation

VOPRF



Client

Functionality

Server

m

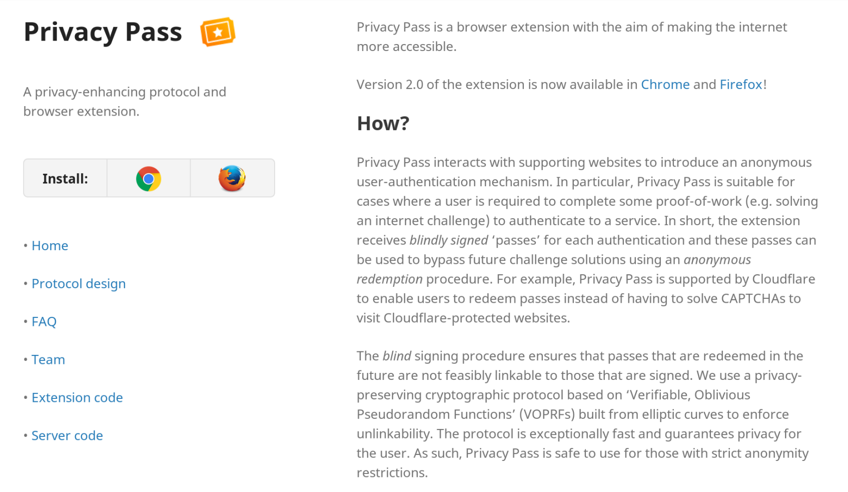
k

$c = F_k(m)$

c

$\perp$

Applications: Privacy Pass I



Privacy Pass

A privacy-enhancing protocol and browser extension.

Privacy Pass is a browser extension with the aim of making the internet more accessible.

Version 2.0 of the extension is now available in Chrome and Firefox!

How?

Privacy Pass interacts with supporting websites to introduce an anonymous user-authentication mechanism. In particular, Privacy Pass is suitable for cases where a user is required to complete some proof-of-work (e.g. solving an internet challenge) to authenticate to a service. In short, the extension receives *blindly signed* ‘passes’ for each authentication and these passes can be used to bypass future challenge solutions using an *anonymous redemption* procedure. For example, Privacy Pass is supported by Cloudflare to enable users to redeem passes instead of having to solve CAPTCHAs to visit Cloudflare-protected websites.

The *blind* signing procedure ensures that passes that are redeemed in the future are not feasibly linkable to those that are signed. We use a privacy-preserving cryptographic protocol based on ‘Verifiable, Oblivious Pseudorandom Functions’ (VOPRFs) built from elliptic curves to enforce unlinkability. The protocol is exceptionally fast and guarantees privacy for the user. As such, Privacy Pass is safe to use for those with strict anonymity restrictions.

Applications: Privacy Pass II

Problem:

- Tor users are having a hard time on Cloudflare protected sites
- They’re constantly asked to solve CAPTCHAs to prove that they’re not bots
- Want a privacy-preserving way of running reverse Turing test once and re-use later

Idea:

- Solve CAPTCHA
- Evaluate a VOPRF on a bunch of random points to produce tokens $F_k(x_i)$
- Redeem token by sending $(x_i, F_k(x_i))$

Alex Davidson et al. Privacy Pass: Bypassing Internet Challenges Anonymously. In: *PoPETs* 2018.3 (July 2018), pp. 164–180. DOI: `10.1515/popets-2018-0026`

Applications: OPAQUE

Problem:

- Passwords are everywhere,
- but servers know passwords, e.g.
  - phishing exploits that password are sent to server in clear
  - server breach
  - ...

Idea:

Registration Client stores $Env_C = Enc_{rwd}(sk_C, pk_S)$ on server with $rwd = F_k(pwd)$

Login run OPRF $rwd = F_k(pwd)$, client decrypts Env_C and runs key-exchange with S.

Stanislaw Jarecki, Hugo Krawczyk, and Jiayu Xu. OPAQUE: An Asymmetric PAKE Protocol Secure Against Pre-computation Attacks. In: *EUROCRYPT 2018, Part III*. ed. by Jesper Buus Nielsen and Vincent Rijmen. Vol. 10822. LNCS. Springer, Heidelberg, 2018, pp. 456–486. DOI: 10.1007/978-3-319-78372-7_15

Applications: Other

- Secure keyword search[1]
- Private set intersection[2]
- Secure data de-duplication[3]
- Password-protected secret sharing[4]

[1] Michael J. Freedman, Yuval Ishai, Benny Pinkas, and Omer Reingold. Keyword Search and Oblivious Pseudorandom Functions. In: *TCC 2005*. Ed. by Joe Kilian. Vol. 3378. LNCS. Springer, Heidelberg, Feb. 2005, pp. 303–324. DOI: 10.1007/978-3-540-30576-7_17.

[2] Stanislaw Jarecki and Xiaomin Liu. Efficient Oblivious Pseudorandom Function with Applications to Adaptive OT and Secure Computation of Set Intersection. In: *TCC 2009*. Ed. by Omer Reingold. Vol. 5444. LNCS. Springer, Heidelberg, Mar. 2009, pp. 577–594. DOI: 10.1007/978-3-642-00457-5_34.

[3] Sriram Keelveedhi, Mihir Bellare, and Thomas Ristenpart. DupLESS: Server-Aided Encryption for Deduplicated Storage. In: *Presented as part of the 22nd USENIX Security Symposium (USENIX Security 13)*. Washington, D.C.: USENIX, 2013, pp. 179–194. ISBN: 978-1-931971-03-4.

[4] Stanislaw Jarecki, Aggelos Kiayias, Hugo Krawczyk, and Jiayu Xu. Highly-Efficient and Composable Password-Protected Secret Sharing (Or: How to Protect Your Bitcoin Wallet Online). In: *EuroS&P*. IEEE, 2016, pp. 276–291.

DH-Based VOPRF

Client | Server

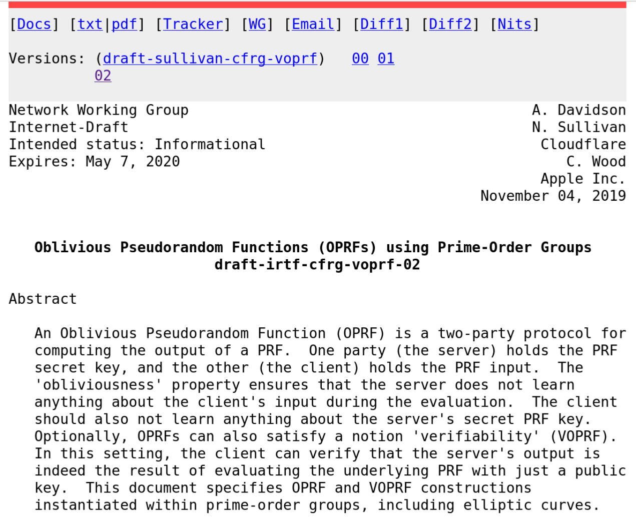
$$\xrightarrow{a = H(x) \cdot g^r}$$

$$\xleftarrow{b = a^k, v = g^k}$$

$H(x)^k = b/v^r$

$$b/v^r = a^k/v^r = (H(x) \cdot g^r)^k/(g^k)^r = H(x)^k$$

Standardisation



[Docs] [txt|pdf] [Tracker] [WG] [Email] [Diff1] [Diff2] [Nits]

Versions: (draft-sullivan-cfrg-voprf) 00 01
02

```
Network Working Group                                        A. Davidson
Internet-Draft                                               N. Sullivan
Intended status: Informational                                Cloudflare
Expires: May 7, 2020                                             C. Wood
                                                              Apple Inc.
                                                       November 04, 2019
```

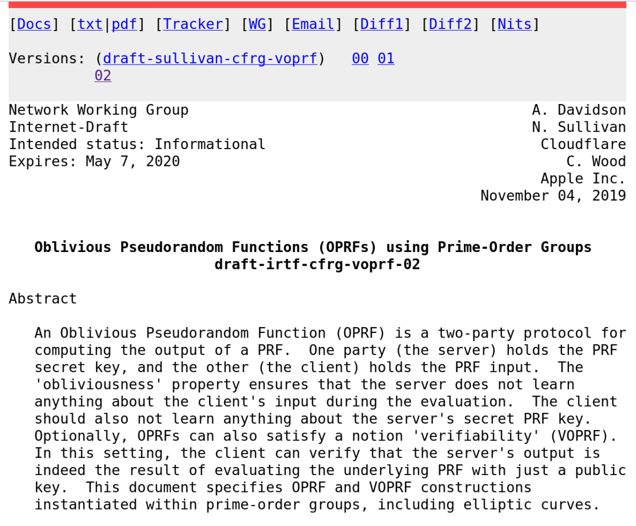
Oblivious Pseudorandom Functions (OPRFs) using Prime-Order Groups
draft-irtf-cfrg-voprf-02

Abstract

An Oblivious Pseudorandom Function (OPRF) is a two-party protocol for computing the output of a PRF. One party (the server) holds the PRF secret key, and the other (the client) holds the PRF input. The 'obliviousness' property ensures that the server does not learn anything about the client's input during the evaluation. The client should also not learn anything about the server's secret PRF key. Optionally, OPRFs can also satisfy a notion 'verifiability' (VOPRF). In this setting, the client can verify that the server's output is indeed the result of evaluating the underlying PRF with just a public key. This document specifies OPRF and VOPRF constructions instantiated within prime-order groups, including elliptic curves.

- OPAQUE is currently being considered for standardisation
- DH-based VOPRFs are currently being considered for standardisation

Standardisation



[Docs] [txt|pdf] [Tracker] [WG] [Email] [Diff1] [Diff2] [Nits]

Versions: (draft-sullivan-cfrg-voprf) 00 01
02

```
Network Working Group                                        A. Davidson
Internet-Draft                                               N. Sullivan
Intended status: Informational                                Cloudflare
Expires: May 7, 2020                                             C. Wood
                                                              Apple Inc.
                                                       November 04, 2019
```

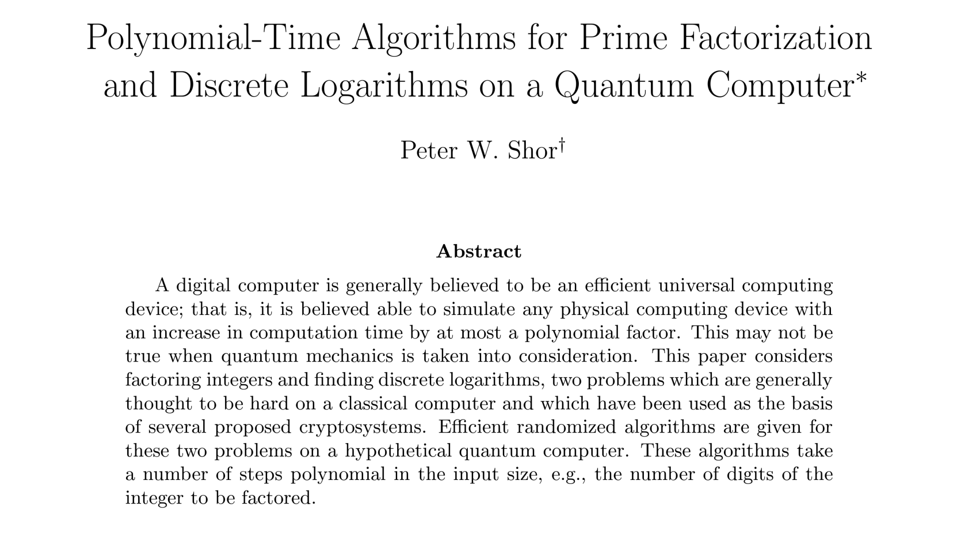
Oblivious Pseudorandom Functions (OPRFs) using Prime-Order Groups
draft-irtf-cfrg-voprf-02

Abstract

An Oblivious Pseudorandom Function (OPRF) is a two-party protocol for computing the output of a PRF. One party (the server) holds the PRF secret key, and the other (the client) holds the PRF input. The 'obliviousness' property ensures that the server does not learn anything about the client's input during the evaluation. The client should also not learn anything about the server's secret PRF key. Optionally, OPRFs can also satisfy a notion 'verifiability' (VOPRF). In this setting, the client can verify that the server's output is indeed the result of evaluating the underlying PRF with just a public key. This document specifies OPRF and VOPRF constructions instantiated within prime-order groups, including elliptic curves.

- OPAQUE is currently being considered for standardisation
- DH-based VOPRFs are currently being considered for standardisation

VOPRFs in a Post-Quantum World



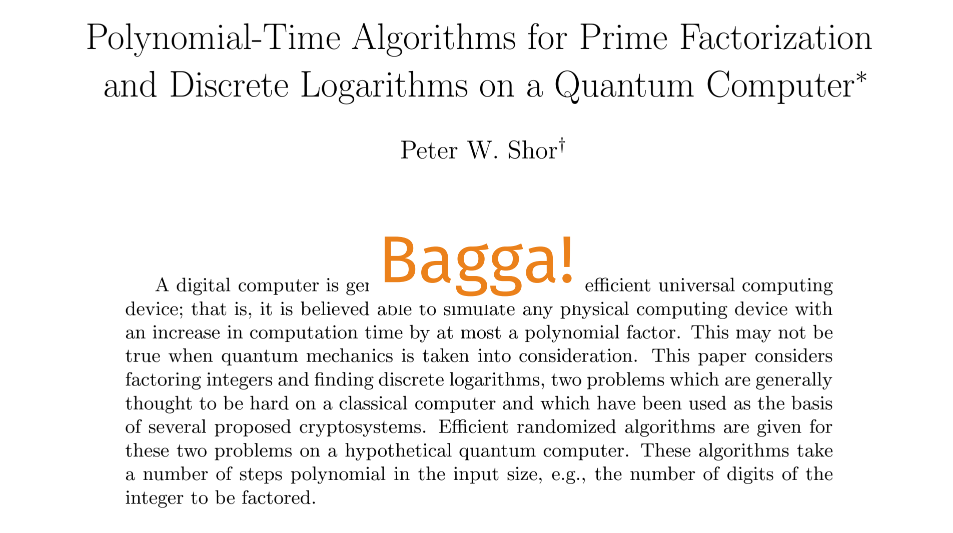
Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*

Peter W. Shor[†]

Abstract

A digital computer is generally believed to be an efficient universal computing device; that is, it is believed able to simulate any physical computing device with an increase in computation time by at most a polynomial factor. This may not be true when quantum mechanics is taken into consideration. This paper considers factoring integers and finding discrete logarithms, two problems which are generally thought to be hard on a classical computer and which have been used as the basis of several proposed cryptosystems. Efficient randomized algorithms are given for these two problems on a hypothetical quantum computer. These algorithms take a number of steps polynomial in the input size, e.g., the number of digits of the integer to be factored.

VOPRFs in a Post-Quantum World



Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*

Peter W. Shor[†]

A digital computer is gen efficient universal computing device; that is, it is believed able to simulate any physical computing device with an increase in computation time by at most a polynomial factor. This may not be true when quantum mechanics is taken into consideration. This paper considers factoring integers and finding discrete logarithms, two problems which are generally thought to be hard on a classical computer and which have been used as the basis of several proposed cryptosystems. Efficient randomized algorithms are given for these two problems on a hypothetical quantum computer. These algorithms take a number of steps polynomial in the input size, e.g., the number of digits of the integer to be factored.

Bagga!

A VOPRF from Lattices

Ring-LWE/Polynomial-LWE

Definition

Let $q, n, \sigma > 0$ depend on λ (q, n are integers). The **decision-RLWE problem**[5] is to distinguish between:

$$(a_i,\ a_i \cdot s + e_i) \in (R_q)^2 \quad \text{and} \quad (a_i, u_i) \in (R_q)^2$$

for $a_i, u_i \leftarrow_\$ R_q$; $s, e_i \leftarrow_\$ R(\chi_\sigma)$

- Think $R_q = \mathbb{Z}_q[x]/(x^n + 1)$ where n is a power of two.
- $\leftarrow_\$ R(\chi_\sigma)$ returns elements with small coefficients.

[5] Damien Stehlé, Ron Steinfeld, Keisuke Tanaka, and Keita Xagawa. Efficient Public Key Encryption Based on Ideal Lattices. In: *ASIACRYPT 2009*. Ed. by Mitsuru Matsui. Vol. 5912. LNCS. Springer, Heidelberg, Dec. 2009, pp. 617–635. DOI: 10.1007/978-3-642-10366-7_36; Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On Ideal Lattices and Learning with Errors over Rings. In: *EUROCRYPT 2010*. Ed. by Henri Gilbert. Vol. 6110. LNCS. Springer, Heidelberg, 2010, pp. 1–23. DOI: 10.1007/978-3-642-13190-5_1.

1D-SIS

Definition

Let q, m, t depend on λ. The **one-dimensional SIS problem**[6] is: Given a uniform $\mathbf{v} \leftarrow_{\$} \mathbb{Z}_q^m$, find $\mathbf{z} \in \mathbb{Z}^m$ such that $\|\mathbf{z}\|_\infty \leq t$ and $\langle \mathbf{v}, \mathbf{z} \rangle \in [-t, t] + q\mathbb{Z}$.

- Informally, the problem asks for a short element producing a short element when multiplied with a random vector.
- The problem can be instantiated with vectors over the ring R_q.

[6] Zvika Brakerski and Vinod Vaikuntanathan. Constrained Key-Homomorphic PRFs from Standard Lattice Assumptions - Or: How to Secretly Embed a Circuit in Your PRF. In: *TCC 2015, Part II*. ed. by Yevgeniy Dodis and Jesper Buus Nielsen. Vol. 9015. LNCS. Springer, Heidelberg, Mar. 2015, pp. 1–30. DOI: 10.1007/978-3-662-46497-7_1.

“Over Ideal Lattices”

We have: If

- **Ring-LWE** is easy then finding short vectors in ideal lattices is easy on a quantum computer and if
- **1D-SIS** is easy over rings then finding short vectors in ideal lattices is easy.

[7] Ronald Cramer, Léo Ducas, and Benjamin Wesolowski. Short Stickelberger Class Relations and Application to Ideal-SVP. In: *EUROCRYPT 2017, Part I*. ed. by Jean-Sébastien Coron and Jesper Buus Nielsen. Vol. 10210. LNCS. Springer, Heidelberg, 2017, pp. 324–348. DOI: 10.1007/978-3-319-56620-7_12.

"Over Ideal Lattices"

We have: If

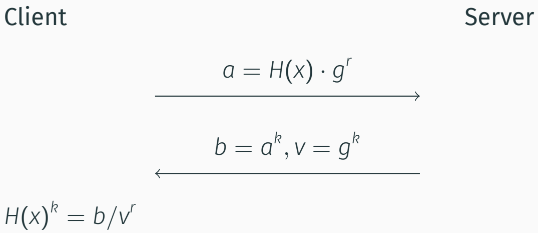
- **Ring-LWE** is easy then finding short vectors in ideal lattices is easy on a quantum computer and if
- **1D-SIS** is easy over rings then finding short vectors in ideal lattices is easy.

Ideal-SVP

At this point, we might have more confidence in Ring-LWE/Ring-SIS being hard on a quantum computer than Ideal-SVP.[7]

[7] Ronald Cramer, Léo Ducas, and Benjamin Wesolowski. Short Stickelberger Class Relations and Application to Ideal-SVP. In: *EUROCRYPT 2017, Part I*. ed. by Jean-Sébastien Coron and Jesper Buus Nielsen. Vol. 10210. LNCS. Springer, Heidelberg, 2017, pp. 324–348. DOI: `10.1007/978-3-319-56620-7_12`.

VOPRF Blueprint



Client

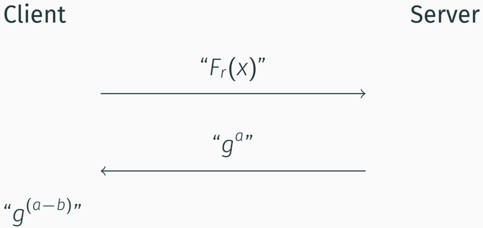
Server

$$\xrightarrow{a = H(x) \cdot g^r}$$

$$\xleftarrow{b = a^k, v = g^k}$$

$H(x)^k = b/v^r$

VOPRF Blueprint



Client

Server

“$F_r(x)$”

“g^a”

“$g^{(a-b)}$”

DH to Ring-LWE Dictionary

DH Land	Ring-LWE Land
g	a
g^x	$a \cdot s + e$
$g^x \cdot g^y = g^{x+y}$	$(a \cdot s + e_0) + (a \cdot t + e_1) = a \cdot (s + t) + e'$
$(g^a)^b = (g^b)^a$	$(a \cdot s + e) \cdot t = (a \cdot s \cdot t + e \cdot t)$
	$\approx a \cdot s \cdot t \approx (a \cdot t + e) \cdot s$
	assuming s and t are small
(g, g^a, g^b, g^{ab})	$(a,\ a \cdot s + e,\ a \cdot t + d,\ a \cdot s \cdot t + e')$
$\approx_c (g, g^a, g^b, u)$	$\approx_c (a,\ a \cdot s + e,\ a \cdot t + d,\ u)$

(Ring-)LWR: Derandomised (Ring-)LWE

Ring-LWE effectively overwrites the lower order bits of $a \cdot s$ with e. Ring-LWR simply drops those bits.

Definition

Let q, n, p depend on λ be integers and $p \mid q$. The **decision-RLWR problem** is to distinguish between:

$$\left(a_i,\ \left\lfloor \frac{p}{q} \cdot a_i \cdot s \right\rceil\right) \in (R_q, R_p) \quad \text{and} \quad (a_i, u_i) \in (R_q, R_p)$$

for $a_i \leftarrow_{\$} R_q$, $s \leftarrow_{\$} R(\chi_\sigma)$, $u_i \leftarrow_{\$} R_p$.

The security of LWR can be reduced to LWE.

LWR-Based PRF: BP14 I

For a particular function $\mathbf{a}^F : \{0,1\}^L \to R_q^{1\times\ell}$ we set out to design a VOPRF for the PRF

$$F_k(x) = \left\lfloor \frac{p}{q} \cdot \mathbf{a}^F(x) \cdot k \right\rceil$$

where the key $k \in R_q$ is small.

Abhishek Banerjee and Chris Peikert. New and Improved Key-Homomorphic Pseudorandom Functions. In: *CRYPTO 2014, Part I*. ed. by Juan A. Garay and Rosario Gennaro. Vol. 8616. LNCS. Springer, Heidelberg, Aug. 2014, pp. 353–370. DOI: 10.1007/978-3-662-44371-2_20

LWR-Based PRF: BP14 II

As an example, consider the PRF for 2-bit inputs.
We define $\mathbf{a}^F(x) = \mathbf{a}_{x_0} \cdot G^{-1}(\mathbf{a}_{x_1})$ where

- $\mathbf{a}_0, \mathbf{a}_1 \in R_q^{1\times\ell}$ are uniform,
- $G^{-1}(\mathbf{a}_2) \in R_2^{\ell\times\ell}$ is binary decomposition,
- $G = (1, 2, \ldots, 2^{\ell-1})$.

Example:

- $x = 5 \bmod 8$
- $G^{-1}(5) = (1, 0, 1)$
- $G \cdot (1, 0, 1) = (1, 2, 4) \cdot (1, 0, 1) = 1 + 4 = 5$

LWR-Based PRF: BP14 III

We can write

$$\begin{aligned}
\left\lfloor \frac{p}{q} \cdot \mathbf{a}^F(x) \cdot k \right\rceil &= \left\lfloor \frac{p}{q} \cdot k \cdot \mathbf{a}_{x_0} \cdot G^{-1}(\mathbf{a}_{x_1}) \right\rceil = \left\lfloor \frac{p}{q} \cdot (k \cdot \mathbf{a}_{x_0} + \mathbf{e}) \cdot G^{-1}(\mathbf{a}_{x_1}) \right\rceil \\
&\approx_c \left\lfloor \frac{p}{q} \cdot \mathbf{u} \cdot G^{-1}(\mathbf{a}_{x_1}) \right\rceil \text{ (RLWE)} \\
&= \left\lfloor \frac{p}{q} (u' G + \mathbf{e}') \cdot G^{-1}(\mathbf{a}_{x_1}) \right\rceil = \left\lfloor \frac{p}{q} (u' \mathbf{a}_{x_1} + \mathbf{e}'') + \frac{p}{q} \mathbf{e}' \cdot G^{-1}(\mathbf{a}_{x_1}) \right\rceil \\
&\approx_c \left\lfloor \frac{p}{q} \cdot \mathbf{u}'' + \frac{p}{q} \cdot \mathbf{e}' \cdot G^{-1}(\mathbf{a}_{x_1}) \right\rceil \text{ (RLWE)} \\
&= \left\lfloor \frac{p}{q} \cdot \tilde{\mathbf{u}} \right\rceil
\end{aligned}$$

where $\mathbf{u}, \mathbf{u}'', \tilde{\mathbf{u}}$ are uniform in $R_q^{1\times\ell}$, u' is uniform in R_q and uniform $\mathbf{e}' \in R_q^{1\times\ell}/(R_q \cdot G)$.

A First Attempt

Client Server

$$\xrightarrow{\mathbf{c}_x = \mathbf{a}^F(x) \cdot r + \mathbf{e}}$$

$$\xleftarrow{\mathbf{d}_x = \mathbf{c}_x \cdot k + \mathbf{e}'}$$

$$\left\lfloor \frac{p}{q} \cdot \mathbf{d}_x \cdot r^{-1} \right\rceil$$

We **would like to say** that

$$\left\lfloor \frac{p}{q} \cdot \mathbf{d}_x \cdot r^{-1} \right\rceil = \left\lfloor \frac{p}{q} \cdot \mathbf{a}^F(x) \cdot k + \frac{p}{q} \left(\mathbf{e} \cdot k \cdot r^{-1} + \mathbf{e}' \cdot r^{-1} \right) \right\rceil = \left\lfloor \frac{p}{q} \cdot \mathbf{a}^F(x) \cdot k \right\rceil .$$

Problem 1

- This simplified protocol cannot be realised using standard RLWE secret distributions.
- The problem is that there is no standard RLWE secret distribution where samples from the distribution are guaranteed to have small inverses in R_q.

Secret distribution:

uniform	fine
error distribution	fine
small	fine, small loss
s^{-1} small	maybe fine, but not proven

“Every problem in computer science can be solved by adding another layer of indirection”

1. Sample small ring elements s and t.
2. Run the extended GCD algorithm to compute some $u' \cdot s + v' \cdot t = 1$.
3. Observe that

$$(u' - r \cdot t) \cdot s + (v' + r \cdot s) \cdot t = u' \cdot s - r \cdot t \cdot s + v' \cdot t + r \cdot s \cdot t = 1$$

4. Use Babai’s rounding algorithm to find r s.t. $u = u' - r \cdot s$ and $v = v' + r$ are small.

Result

We end up with $u \cdot s + v \cdot t = 1 \bmod R_q$ where u, s, v, t are all small.[8]

[8] Jeffrey Hoffstein et al. NTRUSIGN: Digital Signatures Using the NTRU Lattice. In: *CT-RSA 2003*. Ed. by Marc Joye. Vol. 2612. LNCS. Springer, Heidelberg, Apr. 2003, pp. 122–140. DOI: `10.1007/3-540-36563-X_9`; Thomas Pornin and Thomas Prest. More Efficient Algorithms for the NTRU Key Generation using the Field Norm. Cryptology ePrint Archive, Report 2019/015. `https://eprint.iacr.org/2019/015`. 2019.

A Second Attempt

Client Server

$$\mathbf{c}_x^1 = \mathbf{a}^F(x) \cdot s + \mathbf{e}_1, \quad \mathbf{c}_x^2 = \mathbf{a}^F(x) \cdot t + \mathbf{e}_2$$

$$\xrightarrow{\hspace{6cm}}$$

$$\mathbf{d}_x^1 = \mathbf{c}_x^1 \cdot k + \mathbf{e}_1', \quad \mathbf{d}_x^2 = \mathbf{c}_x^2 \cdot k + \mathbf{e}_2'$$

$$\xleftarrow{\hspace{6cm}}$$

$$\left\lfloor \frac{p}{q} \cdot \left(u \cdot \mathbf{d}_x^1 + v \cdot \mathbf{d}_x^2 \right) \right\rceil$$

We **can** that

$$\left\lfloor \frac{p}{q} \cdot (u \cdot \mathbf{d}_x^1 + v \cdot \mathbf{d}_x^2) \right\rceil = \left\lfloor \frac{p}{q} \cdot (u \cdot \mathbf{a}^F(x) \cdot s \cdot k + v \cdot \mathbf{a}^F(x) \cdot t \cdot k) + \frac{p}{q}\mathbf{e}' \right\rceil = \left\lfloor \frac{p}{q} \cdot \mathbf{a}^F(x) \cdot k \right\rceil$$

Problem 2

Definition

Let $q, n, \sigma > 0$ depend on λ (q, n are integers). The **NTRU problem** is to distinguish between:

$$f/g \in R_q \quad \text{and} \quad u \in R_q$$

for $f, g \leftarrow\!\!\$\, R(\chi_\sigma)$, g invertible in R_q and $u \leftarrow\!\!\$\, R_q$.

Attack:

- Client sends $c_x^1 = \gamma \cdot f/g$ for some scalar γ.
- Server sends $d_x^1 = c_x^1 \cdot k + e_1'$
- Client computes $d_x^1 \cdot g = (\gamma \cdot f/g \cdot k + e_1') \cdot g = \gamma \cdot f \cdot k + e_1' \cdot g$

Our Construction

Client **Server**

$$\xrightarrow{\mathbf{c}_x^1 = \mathbf{a}^F(x) \cdot s + \mathbf{e}_1, \quad \mathbf{c}_x^2 = \mathbf{a}^F(x) \cdot t + \mathbf{e}_2}$$

$$\xrightarrow{\text{“proof” } \pi_1 \text{ that } \mathbf{c}_x^1, \mathbf{c}_x^2 \text{ are well-formed.}}$$

Check π_1

$$\xleftarrow{\mathbf{d}_x^1 = \mathbf{c}_x^1 \cdot k + \mathbf{e}_1', \quad \mathbf{d}_x^2 = \mathbf{c}_x^1 \cdot k + \mathbf{e}_2'}$$

$$\xleftarrow{\text{“proof” } \pi_2 \text{ that } \mathbf{d}_x^1, \mathbf{d}_x^2 \text{ are well-formed.}}$$

Check π_2

$$\left\lfloor \frac{p}{q} \cdot \left(u \cdot \mathbf{d}_x^1 + v \cdot \mathbf{d}_x^2 \right) \right\rceil$$

Security

Notion

A protocol Π is a verifiable oblivious pseudorandom function if all of the following hold:

Correctness the protocol outputs the correct evaluation with overwhelming probability

Malicious server a malicious server cannot tell if it s talking to the ideal functionality or Π

Average case malicious client **for a random** k a malicious client cannot tell if it is talking to the ideal functionality or Π

Client Security: RLWE & 1D-SIS

- The messages $\mathbf{c}_x^1 = \mathbf{a}^F(x) \cdot s + \mathbf{e}_1$ and $\mathbf{c}_x^2 = \mathbf{a}^F(x) \cdot t + \mathbf{e}_2$ are indistinguishable from uniform by RLWE assumption.
- Correctness holds by the 1D-SIS assumption.

Server Security: RLWE & Drowning

- Note that

$$\mathbf{d}_x^1 = \mathbf{a}^F(x) \cdot s \cdot k + \mathbf{e}_1 \cdot k + \mathbf{e}_1'$$

- If we pick $\mathbf{e}_1'$ from a distribution that hides addition of terms $\mathbf{e} \cdot k$ and $\mathbf{e}_s \cdot s$ (where $\mathbf{e}_s$ is identically distributed to $\mathbf{e}$) then
- from the perspective of the client, the server might as well have sent

$$\mathbf{d}_x^1 = (\mathbf{a}^F(x) \cdot k + \mathbf{e}_s) \cdot s + \mathbf{e}_1'.$$

- The term in brackets $\mathbf{a}^F(x) \cdot k + \mathbf{e}_s$ computationally indistinguishable from uniform random under a RLWE assumption
- Thus, the message $\mathbf{d}_x^1$ leaks nothing about the server's key k.

Parameters

They're Disgusting!

- We need super-polynomial q for BP14 and we need super-polynomial q for drowning
  - $\lambda = 128 \Rightarrow q = 2^{256}$
- Need[9] LWE dimension $n = 2^{14}$
  - 2^{22} bits per RLWE sample: 0.5MB, need two samples per direction

ZK Cost

This is ignoring the cost of sending the zero-knowledge arguments

But can tune parameters, round more aggressively, perhaps remove drowning . . .

[9] Martin R. Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of Learning with Errors. In: *Journal of Mathematical Cryptology* 9.3 (2015), pp. 169–203.

Alternative Constructions

Alternative VOPRF Candidate: Prove a Hash Function

- Let $H(\cdot)$ be a zk-friendly hash function[10]
- Prove $\mathrm{seed} = H(x)$ instead of proving $\mathbf{a}^F(x) \cdot s + e$.
- Let $\mathbf{a}_{\mathrm{seed}}$ is the output of some sampler[11] of elements in R_q when given seed as input
- Send $(\mathrm{seed}, \mathbf{a}_{\mathrm{seed}} \cdot s + e)$
- Still need to prove $\mathbf{a}_{\mathrm{seed}} \cdot s + e$ but this is easier/cheaper.

[10] Martin R. Albrecht et al. Ciphers for MPC and FHE. In: *EUROCRYPT 2015, Part I*. ed. by Elisabeth Oswald and Marc Fischlin. Vol. 9056. LNCS. Springer, Heidelberg, Apr. 2015, pp. 430–454. DOI: 10.1007/978-3-662-46800-5_17; Martin R. Albrecht et al. MiMC: Efficient Encryption and Cryptographic Hashing with Minimal Multiplicative Complexity. In: *ASIACRYPT 2016, Part I*. ed. by Jung Hee Cheon and Tsuyoshi Takagi. Vol. 10031. LNCS. Springer, Heidelberg, Dec. 2016, pp. 191–219. DOI: 10.1007/978-3-662-53887-6_7; Abdelrahaman Aly et al. Design of Symmetric-Key Primitives for Advanced Cryptographic Protocols. Cryptology ePrint Archive, Report 2019/426. https://eprint.iacr.org/2019/426. 2019.

[11] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum Key Exchange - A New Hope. In: *USENIX Security 2016*. Ed. by Thorsten Holz and Stefan Savage. USENIX Association, Aug. 2016, pp. 327–343.

Alternative VOPRF Candidate: FHE

1. Client encrypts x under an FHE scheme
2. Sever computes $Eval(F_k, x)$ homomorphically using an FHE friendly PRF[12]
3. Client decrypts $F_k(x)$.

[12] Martin R. Albrecht et al. Ciphers for MPC and FHE. In: *EUROCRYPT 2015, Part I*. ed. by Elisabeth Oswald and Marc Fischlin. Vol. 9056. LNCS. Springer, Heidelberg, Apr. 2015, pp. 430–454. DOI: 10.1007/978-3-662-46800-5_17; Martin R. Albrecht et al. MiMC: Efficient Encryption and Cryptographic Hashing with Minimal Multiplicative Complexity. In: *ASIACRYPT 2016, Part I*. ed. by Jung Hee Cheon and Tsuyoshi Takagi. Vol. 10031. LNCS. Springer, Heidelberg, Dec. 2016, pp. 191–219. DOI: 10.1007/978-3-662-53887-6_7; Abdelrahaman Aly et al. Design of Symmetric-Key Primitives for Advanced Cryptographic Protocols. Cryptology ePrint Archive, Report 2019/426. https://eprint.iacr.org/2019/426. 2019.

Alternative Applications aka "We will miss DH"

- NIST PQ Competition Process only covers ephemeral key exchange and digital signature schemes
- VOPRFs are just one example of DH-based constructions that need translation in a post-quantum world
- We cannot even do an efficient post-quantum NIKE

Fin

Thank You

PS: We are hiring a lecturer/assistant professor in the ISG.

`https://jobs.royalholloway.ac.uk/0120-023`

Application deadline: 15 April

PPS: We are looking for PhD students.

`https://royalholloway.ac.uk/CDT`