

LeOPaRd: Towards Practical Post-Quantum Oblivious PRFs via 2HashDH Paradigm

Muhammed F. Esgin Ron Steinfeld Erkan Tairi Jie Xu



CRYPTO 2026

Oblivious PRFs [JKK14; CHL22]

A PRF F_k evaluated **between two parties**: the server holds the secret key k and the client holds the input x .



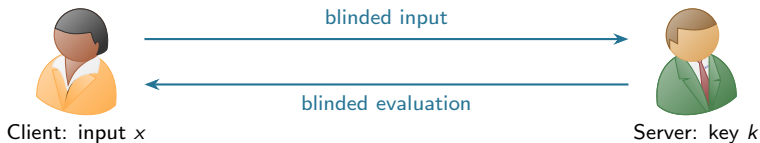
Client: input x



Server: key k

Oblivious PRFs [JKK14; CHL22]

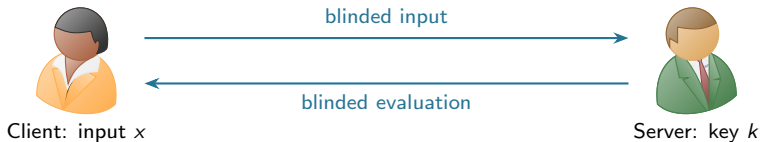
A PRF F_k evaluated **between two parties**: the server holds the secret key k and the client holds the input x .



Oblivious PRFs [JKK14; CHL22]

A PRF F_k evaluated **between two parties**: the server holds the secret key k and the client holds the input x .

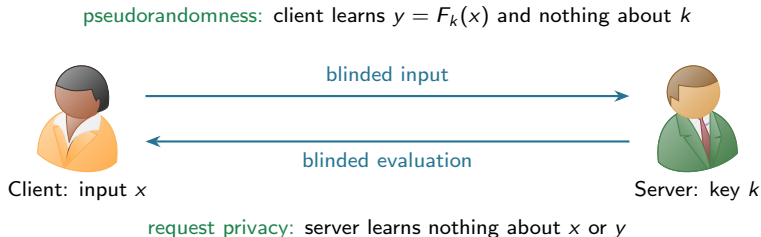
pseudorandomness: client learns $y = F_k(x)$ and nothing about k



request privacy: server learns nothing about x or y

Oblivious PRFs [JKK14; CHL22]

A PRF F_k evaluated **between two parties**: the server holds the secret key k and the client holds the input x .

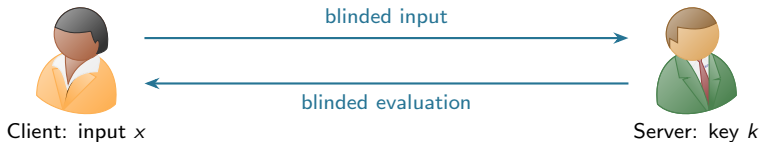


- **Verifiable** (VOPRF): the client is convinced y was computed under the *committed* key.

Oblivious PRFs [JKK14; CHL22]

A PRF F_k evaluated **between two parties**: the server holds the secret key k and the client holds the input x .

pseudorandomness: client learns $y = F_k(x)$ and nothing about k

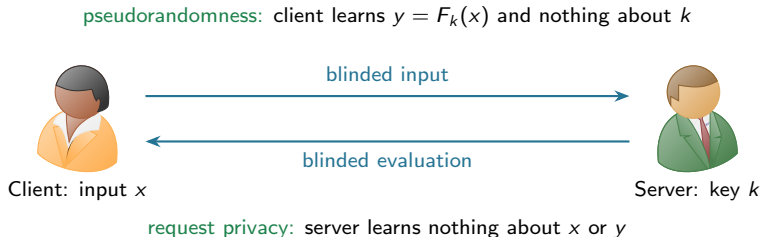


request privacy: server learns nothing about x or y

- **Verifiable** (VOPRF): the client is convinced y was computed under the *committed* key.
- **Partial** (POPRF): input splits into a public tag t and a private x ; $y = F_k(t, x)$.

Oblivious PRFs [JKK14; CHL22]

A PRF F_k evaluated **between two parties**: the server holds the secret key k and the client holds the input x .



- **Verifiable** (VOPRF): the client is convinced y was computed under the *committed* key.
- **Partial** (POPRF): input splits into a public tag t and a private x ; $y = F_k(t, x)$.

Applications: Privacy Pass [Dav+18], PAKE/OPAQUE [JKX18], compromised-credential checking [Li+19], password-protected secret sharing [Jar+16], ...

Post-quantum (V)(P)OPRFs

Scheme	Assumption	Rounds	Communication	Model
[Alb+21]	RLWE + SIS	2	> 128 GB	strong, QROM
[Alb+24]	heuristic (broken [CJ25])	2	15 MB + 633 KB	strong, ROM
[Beu+25]	Legendre PRF	9	911 KB	strong, ROM
[Yan+25]	Power-Residue PRF	≥ 5	$\geq 1,600$ KB	strong, ROM
[AG24]	RLWE	2	108 KB + 189 KB	weak, ROM
LeOPaRd	MLWE-PRF-RU + MLWE + MSIS	2	(10 + 31) KB + 97 KB	strong, ROM

Communication = offline + online per query; for LeOPaRd the offline part is one-time + per-query.

“strong” = simulation-based model of [Tya+22]; “weak” = one-more-style [FLÖ25].

Post-quantum (V)(P)OPRFs

Scheme	Assumption	Rounds	Communication	Model
[Alb+21]	RLWE + SIS	2	> 128 GB	strong, QROM
[Alb+24]	heuristic (broken [CJ25])	2	15 MB + 633 KB	strong, ROM
[Beu+25]	Legendre PRF	9	911 KB	strong, ROM
[Yan+25]	Power-Residue PRF	≥ 5	$\geq 1,600$ KB	strong, ROM
[AG24]	RLWE	2	108 KB + 189 KB	weak, ROM
LeOPaRd	MLWE-PRF-RU + MLWE + MSIS	2	(10 + 31) KB + 97 KB	strong, ROM

Communication = offline + online per query; for LeOPaRd the offline part is one-time + per-query.

“strong” = simulation-based model of [Tya+22]; “weak” = one-more-style [FLÖ25].

≈ 138 KB total and $2\times$ less online communication than the state of the art.

The 2HashDH blueprint [JKK14]

In a group (multiplicative blinding [Bol03]):

- public: g , $pk := g^k$
- $b_x := H_1(x)$, client sends $c_x := b_x \cdot g^r$
- server returns $u_x := c_x^k$
- client unblinds $z := u_x \cdot pk^{-r} = H_1(x)^k$
- output $y := H_2(x, z)$

blinding = a statistical pad: g^r uniform
— stripped **exactly** via the homomorphism

The 2HashDH blueprint [JKK14]

In a group (multiplicative blinding [Bol03]):

- public: g , $pk := g^k$
- $b_x := H_1(x)$, client sends $c_x := b_x \cdot g^r$
- server returns $u_x := c_x^k$
- client unblinds $z := u_x \cdot pk^{-r} = H_1(x)^k$
- output $y := H_2(x, z)$

blinding = a statistical pad: g^r uniform
— stripped **exactly** via the homomorphism

Over lattices: $\mathbf{B}_x := H(x)$

- public: $\mathbf{A}_r$, $\mathbf{v}_k := \mathbf{A}_r \mathbf{k} + \mathbf{e}_s$
- client sends $\mathbf{C}_x := \mathbf{R} \mathbf{A}_r + \mathbf{B}_x$
- server returns $\mathbf{u}_x := \mathbf{C}_x \mathbf{k} + \mathbf{e}'_s$
- client unblinds $\mathbf{u}_x - \mathbf{R} \mathbf{v}_k = \mathbf{B}_x \mathbf{k} + \underbrace{\mathbf{e}'_s - \mathbf{R} \mathbf{e}_s}_{\mathbf{e}_f}$
- round: $z := \lfloor \mathbf{B}_x \mathbf{k} + \mathbf{e}_f \rfloor_p = \lfloor \mathbf{B}_x \mathbf{k} \rfloor_p$

blinding = a computational pad (Knapsack MLWE)
 $\mathbf{R} \mathbf{A}_r \approx$ uniform — stripped only **approximately**

The 2HashDH blueprint [JKK14]

In a group (multiplicative blinding [Bol03]):

- public: g , $pk := g^k$
- $b_x := H_1(x)$, client sends $c_x := b_x \cdot g^r$
- server returns $u_x := c_x^k$
- client unblinds $z := u_x \cdot pk^{-r} = H_1(x)^k$
- output $y := H_2(x, z)$

blinding = a statistical pad: g^r uniform
— stripped **exactly** via the homomorphism

$g^r \mapsto \mathbf{RA}_r$, $pk^{-r} \mapsto -\mathbf{Rv}_k$, outer hash $\mapsto$ **rounding** — correct as long as $\mathbf{e}_f$ stays **small**.

Over lattices: $\mathbf{B}_x := H(x)$

- public: $\mathbf{A}_r$, $\mathbf{v}_k := \mathbf{A}_r \mathbf{k} + \mathbf{e}_s$
- client sends $\mathbf{C}_x := \mathbf{RA}_r + \mathbf{B}_x$
- server returns $\mathbf{u}_x := \mathbf{C}_x \mathbf{k} + \mathbf{e}'_s$
- client unblinds $\mathbf{u}_x - \mathbf{Rv}_k = \mathbf{B}_x \mathbf{k} + \underbrace{\mathbf{e}'_s - \mathbf{Re}_s}_{\mathbf{e}_f}$
- round: $z := \lfloor \mathbf{B}_x \mathbf{k} + \mathbf{e}_f \rfloor_p = \lfloor \mathbf{B}_x \mathbf{k} \rfloor_p$

blinding = a computational pad (Knapsack MLWE)
 $\mathbf{RA}_r \approx$ uniform — stripped only **approximately**

The obstacle: client controls $\mathbf{C}_x$

The server's answer $\mathbf{u}_x = \mathbf{C}_x \mathbf{k} + \mathbf{e}'_s$ *looks like* an MLWE sample, but $\mathbf{C}_x$ is chosen by a malicious client:



client can plant a trapdoored, structured $\mathbf{C}_x$ that leaks $\mathbf{k}$

The obstacle: client controls $\mathbf{C}_x$

The server's answer $\mathbf{u}_x = \mathbf{C}_x \mathbf{k} + \mathbf{e}'_s$ looks like an MLWE sample, but $\mathbf{C}_x$ is chosen by a malicious client:



client can plant a trapdoored, structured $\mathbf{C}_x$ that leaks $\mathbf{k}$

- [Alb+21]: noise flooding — take $\|\mathbf{e}'_s\| = O(2^\lambda)$ so that $\mathbf{u}_x$ statistically hides everything.
 $\Rightarrow \mathbf{e}_f$ is also $O(2^\lambda) \Rightarrow q = O(2^{2\lambda})$, and communication is quadratic in $\log q$.

The obstacle: client controls $\mathbf{C}_x$

The server's answer $\mathbf{u}_x = \mathbf{C}_x \mathbf{k} + \mathbf{e}'_s$ looks like an MLWE sample, but $\mathbf{C}_x$ is chosen by a malicious client:



client can plant a trapdoored, structured $\mathbf{C}_x$ that leaks $\mathbf{k}$

- [Alb+21]: **noise flooding** — take $\|\mathbf{e}'_s\| = O(2^\lambda)$ so that $\mathbf{u}_x$ statistically hides everything.
 $\Rightarrow \mathbf{e}_f$ is also $O(2^\lambda) \Rightarrow q = O(2^{2\lambda})$, and communication is **quadratic in $\log q$** .
- [AG24]: replace flooding by a **Rényi divergence** argument — much better parameters, but
 $\Rightarrow$ forces a **weaker, one-more-style** security model (no indistinguishability-based proof) [FLÖ25];
 $\Rightarrow$ loses $\sqrt{Q}$ in the *total* number of queries Q across *all* inputs (global bound).

The obstacle: client controls $\mathbf{C}_x$

The server's answer $\mathbf{u}_x = \mathbf{C}_x \mathbf{k} + \mathbf{e}'_s$ looks like an MLWE sample, but $\mathbf{C}_x$ is chosen by a malicious client:



client can plant a trapdoored, structured $\mathbf{C}_x$ that leaks $\mathbf{k}$

- [Alb+21]: **noise flooding** — take $\|\mathbf{e}'_s\| = O(2^\lambda)$ so that $\mathbf{u}_x$ statistically hides everything.
 $\Rightarrow \mathbf{e}_f$ is also $O(2^\lambda) \Rightarrow q = O(2^{2\lambda})$, and communication is **quadratic in $\log q$** .
- [AG24]: replace flooding by a **Rényi divergence** argument — much better parameters, but
 $\Rightarrow$ forces a **weaker, one-more-style** security model (no indistinguishability-based proof) [FLÖ25];
 $\Rightarrow$ loses $\sqrt{Q}$ in the *total* number of queries Q across *all* inputs (global bound).

Can we keep the strong model *and* a small modulus (with no flooding and no FHE)?

Our idea: commit-then-sample

Key change

The client **commits to $\mathbf{R}$** first, $\mathbf{c}_r \leftarrow \text{Com}(\mathbf{R})$, and only then gets $\mathbf{A}_r := \text{RO}_r(\mathbf{c}_r)$.

Our idea: commit-then-sample

Key change

The client **commits to $\mathbf{R}$** first, $\mathbf{c}_r \leftarrow \text{Com}(\mathbf{R})$, and only then gets $\mathbf{A}_r := \text{RO}_r(\mathbf{c}_r)$.

- The client must fix **$\mathbf{R}$ before** seeing the MLWE sample $\mathbf{v}_k = \mathbf{A}_r \mathbf{k} + \mathbf{e}_s \Rightarrow$ it cannot bias $\mathbf{C}_x$ as a function of the sample.

Our idea: commit-then-sample

Key change

The client **commits to $\mathbf{R}$** first, $\mathbf{c}_r \leftarrow \text{Com}(\mathbf{R})$, and only then gets $\mathbf{A}_r := \text{RO}_r(\mathbf{c}_r)$.

- The client must fix **$\mathbf{R}$ before** seeing the MLWE sample $\mathbf{v}_k = \mathbf{A}_r \mathbf{k} + \mathbf{e}_s \Rightarrow$ it cannot bias $\mathbf{C}_x$ as a function of the sample.
- Given $(\mathbf{v}_k, \mathbf{z})$ with $\mathbf{z} = \mathbf{B}_{x,t} \mathbf{k} + \mathbf{e}'_s - \mathbf{R} \mathbf{e}_s$, the reply is *determined*: $\mathbf{u}_x = \mathbf{z} + \mathbf{R} \mathbf{v}_k$ — the security reduces to whether the pair $(\mathbf{v}_k, \mathbf{z})$ hides $\mathbf{k}$.

Our idea: commit-then-sample

Key change

The client **commits to $\mathbf{R}$** first, $\mathbf{c}_r \leftarrow \text{Com}(\mathbf{R})$, and only then gets $\mathbf{A}_r := \text{RO}_r(\mathbf{c}_r)$.

- The client must fix $\mathbf{R}$ **before** seeing the MLWE sample $\mathbf{v}_k = \mathbf{A}_r \mathbf{k} + \mathbf{e}_s \Rightarrow$ it cannot bias $\mathbf{C}_x$ as a function of the sample.
- Given $(\mathbf{v}_k, \mathbf{z})$ with $\mathbf{z} = \mathbf{B}_{x,t} \mathbf{k} + \mathbf{e}'_s - \mathbf{R} \mathbf{e}_s$, the reply is *determined*: $\mathbf{u}_x = \mathbf{z} + \mathbf{R} \mathbf{v}_k$ — the security reduces to whether the pair $(\mathbf{v}_k, \mathbf{z})$ hides $\mathbf{k}$.
- New observation: building $\mathbf{C}_x$ in **Knapsack form** $\mathbf{R} \mathbf{A}_r + \mathbf{B}_{x,t}$ (instead of $\mathbf{S} \mathbf{A}_r + \mathbf{E} + \mathbf{B}_{x,t}$) makes $\mathbf{e}_f = \mathbf{e}'_s - \mathbf{R} \mathbf{e}_s$ **independent of $\mathbf{k}$** $\Rightarrow$ a *large* PRF key is allowed.

Our idea: commit-then-sample

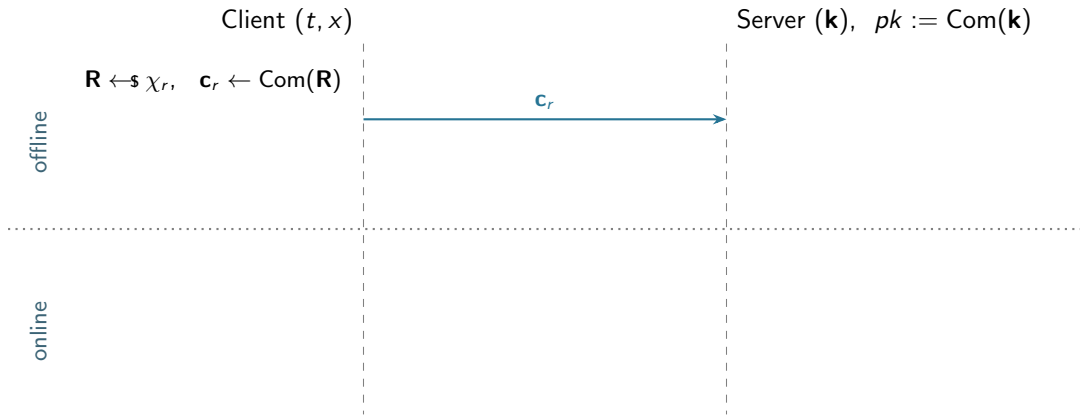
Key change

The client **commits to $\mathbf{R}$** first, $\mathbf{c}_r \leftarrow \text{Com}(\mathbf{R})$, and only then gets $\mathbf{A}_r := \text{RO}_r(\mathbf{c}_r)$.

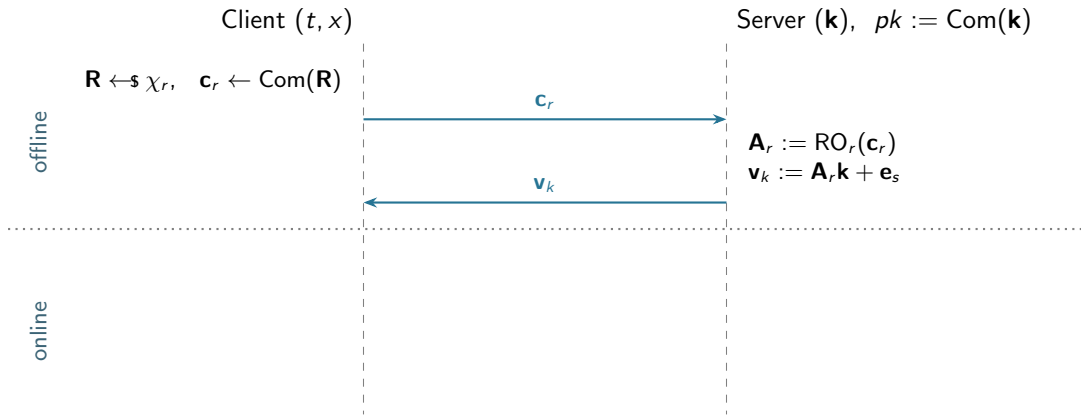
- The client must fix $\mathbf{R}$ **before** seeing the MLWE sample $\mathbf{v}_k = \mathbf{A}_r \mathbf{k} + \mathbf{e}_s \Rightarrow$ it cannot bias $\mathbf{C}_x$ as a function of the sample.
- Given $(\mathbf{v}_k, \mathbf{z})$ with $\mathbf{z} = \mathbf{B}_{x,t} \mathbf{k} + \mathbf{e}'_s - \mathbf{R} \mathbf{e}_s$, the reply is *determined*: $\mathbf{u}_x = \mathbf{z} + \mathbf{R} \mathbf{v}_k$ — the security reduces to whether the pair $(\mathbf{v}_k, \mathbf{z})$ hides $\mathbf{k}$.
- New observation: building $\mathbf{C}_x$ in **Knapsack form** $\mathbf{R} \mathbf{A}_r + \mathbf{B}_{x,t}$ (instead of $\mathbf{S} \mathbf{A}_r + \mathbf{E} + \mathbf{B}_{x,t}$) makes $\mathbf{e}_f = \mathbf{e}'_s - \mathbf{R} \mathbf{e}_s$ **independent of $\mathbf{k}$** $\Rightarrow$ a *large* PRF key is allowed.
- Now $\mathbf{e}'_s$ only has to be **polynomially** wider than $\mathbf{e}_s \Rightarrow$ no noise flooding.

MLWE-PRF-RU: “a mix of MLWE and deficient PRF samples, under a committed $\mathbf{R}$, hides $\mathbf{k}$ ”

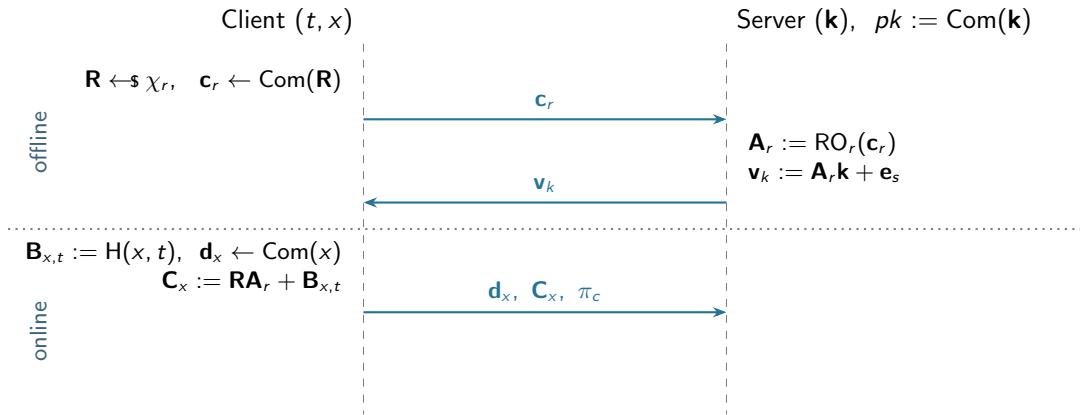
LeOPaRd



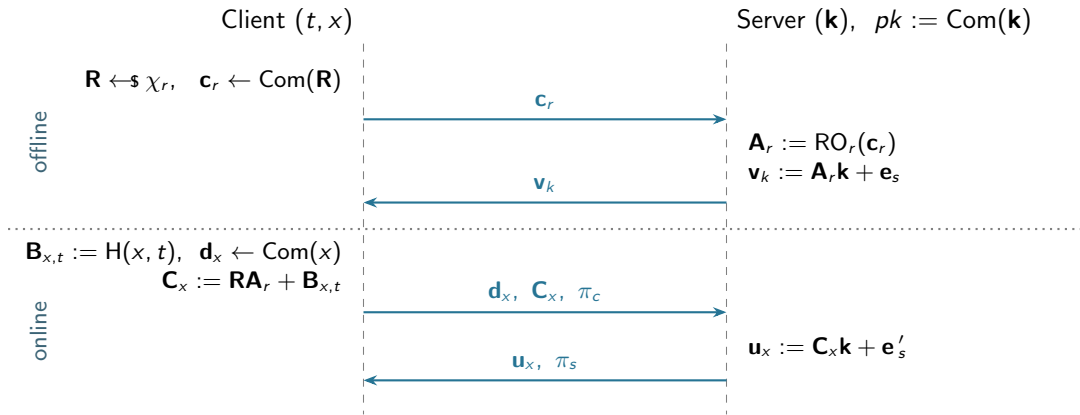
LeOPaRd



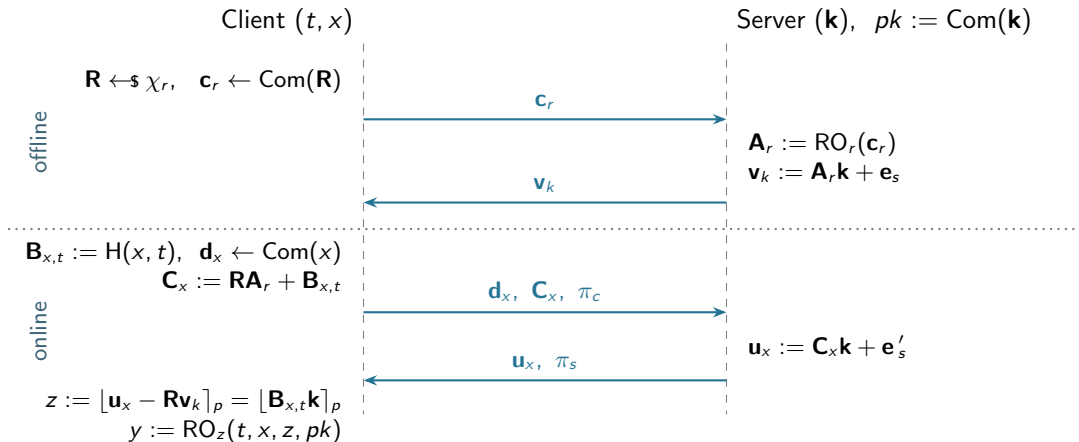
LeOPaRd



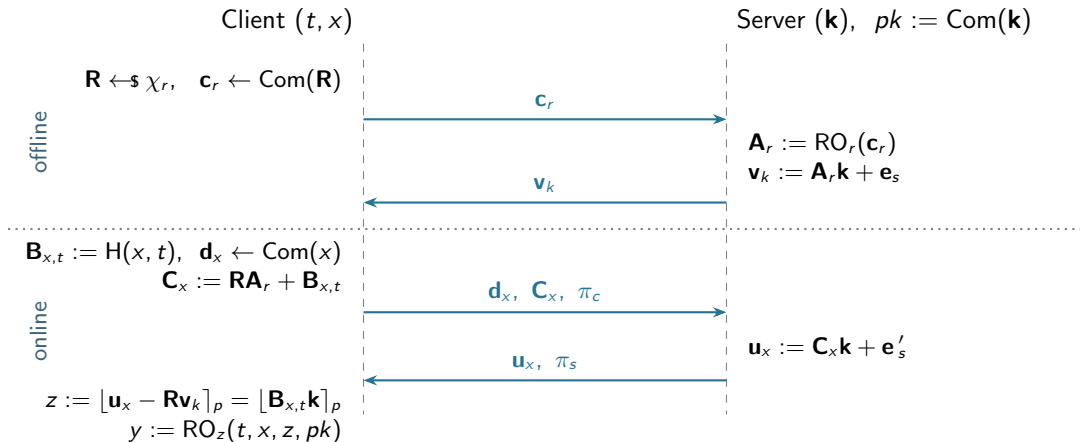
LeOPaRd



LeOPaRd



LeOPaRd



- π_c, π_s : well-formedness NIZKs (LaBRADOR [BS23] + [LNP22]); **no straight-line extractability needed.**

New assumption: MLWE-PRF-RU

A mix of MLWE and PRF samples in which the PRF error is **deficient** — exactly what the client sees.

MLWE-PRF-RU^b ($\mathbf{k} \leftarrow \$ \chi_k^m$; adversary queries adaptively, $\|\mathbf{R}\|_\infty \leq \beta_r, \leq Q_x^\infty$ queries per x)

SampMLWEcom($\mathbf{R}$): $\mathbf{A} \leftarrow \$ R_q^{N \times m}, \mathbf{e} \leftarrow \$ \chi^N$

$\mathbf{c} := \mathbf{A}\mathbf{k} + \mathbf{e}$ ($b = 1$) or $\mathbf{c} \leftarrow \$ R_q^N$ ($b = 0$); store $(\mathbf{c}, \mathbf{e}, \mathbf{R})$; return $(\mathbf{A}, \mathbf{c})$

SampPRF-RU($\mathbf{A}, x$): look up $(\mathbf{e}, \mathbf{R})$ for $\mathbf{A}$, $\mathbf{e}' \leftarrow \$ \chi_1^h, \mathbf{B}_x := H(x)$

return $\mathbf{z} := \mathbf{B}_x \mathbf{k} + \mathbf{e}' - \mathbf{R}\mathbf{e}$ ($b = 1$) or $\mathbf{z} := \mathbf{y}_x + \mathbf{e}' - \mathbf{R}\mathbf{e}, \mathbf{y}_x \leftarrow \$ R_q^h$ ($b = 0$)

- Correspondence: $\mathbf{c} \leftrightarrow \mathbf{v}_k, \mathbf{z} \leftrightarrow \mathbf{u}_x - \mathbf{R}\mathbf{v}_k$

New assumption: MLWE-PRF-RU

A mix of MLWE and PRF samples in which the PRF error is **deficient** — exactly what the client sees.

MLWE-PRF-RU^b ($\mathbf{k} \leftarrow \$ \chi_k^m$; adversary queries adaptively, $\|\mathbf{R}\|_\infty \leq \beta_r, \leq Q_x^\infty$ queries per x)

SampMLWEcom($\mathbf{R}$): $\mathbf{A} \leftarrow \$ R_q^{N \times m}, \mathbf{e} \leftarrow \$ \chi^N$

$\mathbf{c} := \mathbf{A}\mathbf{k} + \mathbf{e}$ ($b = 1$) or $\mathbf{c} \leftarrow \$ R_q^N$ ($b = 0$); store $(\mathbf{c}, \mathbf{e}, \mathbf{R})$; return $(\mathbf{A}, \mathbf{c})$

SampPRF-RU($\mathbf{A}, x$): look up $(\mathbf{e}, \mathbf{R})$ for $\mathbf{A}$, $\mathbf{e}' \leftarrow \$ \chi_1^h, \mathbf{B}_x := H(x)$

return $\mathbf{z} := \mathbf{B}_x \mathbf{k} + \mathbf{e}' - \mathbf{R}\mathbf{e}$ ($b = 1$) or $\mathbf{z} := \mathbf{y}_x + \mathbf{e}' - \mathbf{R}\mathbf{e}, \mathbf{y}_x \leftarrow \$ R_q^h$ ($b = 0$)

- Correspondence: $\mathbf{c} \leftrightarrow \mathbf{v}_k, \mathbf{z} \leftrightarrow \mathbf{u}_x - \mathbf{R}\mathbf{v}_k$
- Re-use: the *same* MLWE error $\mathbf{e}$ appears in $\mathbf{c}$ and in $\mathbf{z}$ — the PRF error $\mathbf{e}' - \mathbf{R}\mathbf{e}$ is **deficient**: correlated, and not a clean Gaussian.

New assumption: MLWE-PRF-RU

A mix of MLWE and PRF samples in which the PRF error is **deficient** — exactly what the client sees.

MLWE-PRF-RU^b ($\mathbf{k} \leftarrow \$ \chi_k^m$; adversary queries adaptively, $\|\mathbf{R}\|_\infty \leq \beta_r, \leq Q_x^\infty$ queries per x)

SampMLWEcom($\mathbf{R}$): $\mathbf{A} \leftarrow \$ R_q^{N \times m}, \mathbf{e} \leftarrow \$ \chi^N$

$\mathbf{c} := \mathbf{A}\mathbf{k} + \mathbf{e}$ ($b = 1$) or $\mathbf{c} \leftarrow \$ R_q^N$ ($b = 0$); store $(\mathbf{c}, \mathbf{e}, \mathbf{R})$; return $(\mathbf{A}, \mathbf{c})$

SampPRF-RU($\mathbf{A}, x$): look up $(\mathbf{e}, \mathbf{R})$ for $\mathbf{A}$, $\mathbf{e}' \leftarrow \$ \chi_1^h, \mathbf{B}_x := \mathbf{H}(x)$

return $\mathbf{z} := \mathbf{B}_x \mathbf{k} + \mathbf{e}' - \mathbf{R}\mathbf{e}$ ($b = 1$) or $\mathbf{z} := \mathbf{y}_x + \mathbf{e}' - \mathbf{R}\mathbf{e}, \mathbf{y}_x \leftarrow \$ R_q^h$ ($b = 0$)

- Correspondence: $\mathbf{c} \leftrightarrow \mathbf{v}_k, \mathbf{z} \leftrightarrow \mathbf{u}_x - \mathbf{R}\mathbf{v}_k$
- Re-use: the *same* MLWE error $\mathbf{e}$ appears in $\mathbf{c}$ and in $\mathbf{z}$ — the PRF error $\mathbf{e}' - \mathbf{R}\mathbf{e}$ is **deficient**: correlated, and not a clean Gaussian.
- Commit-then-sample: $\mathbf{R}$ is an *input* to SampMLWEcom — fixed *before* $(\mathbf{A}, \mathbf{c})$ is seen.
Without this, the adversary could correlate $\mathbf{R}$ with the sample and plant a trapdoor.

Reduction to MLWE

MLWE-PRF-RU (what LeOPaRd's proof needs)

$$\mathbf{c}_i = \mathbf{A}_i \mathbf{k} + \mathbf{e}_i, \quad \mathbf{z}_{x,i} = \mathbf{B}_x \mathbf{k} + \mathbf{e}'_i - \mathbf{R}_i \mathbf{e}_i$$

Reduction to MLWE

MLWE-PRF-RU (what LeOPaRd's proof needs)

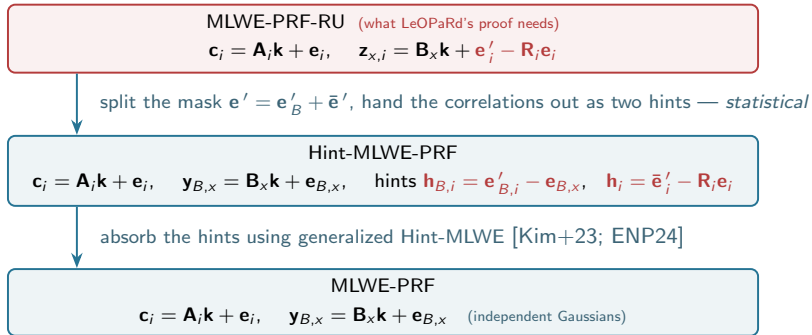
$$\mathbf{c}_i = \mathbf{A}_i \mathbf{k} + \mathbf{e}_i, \quad \mathbf{z}_{x,i} = \mathbf{B}_x \mathbf{k} + \mathbf{e}'_i - \mathbf{R}_i \mathbf{e}_i$$

split the mask $\mathbf{e}' = \mathbf{e}'_B + \bar{\mathbf{e}}'$, hand the correlations out as two hints — *statistical*

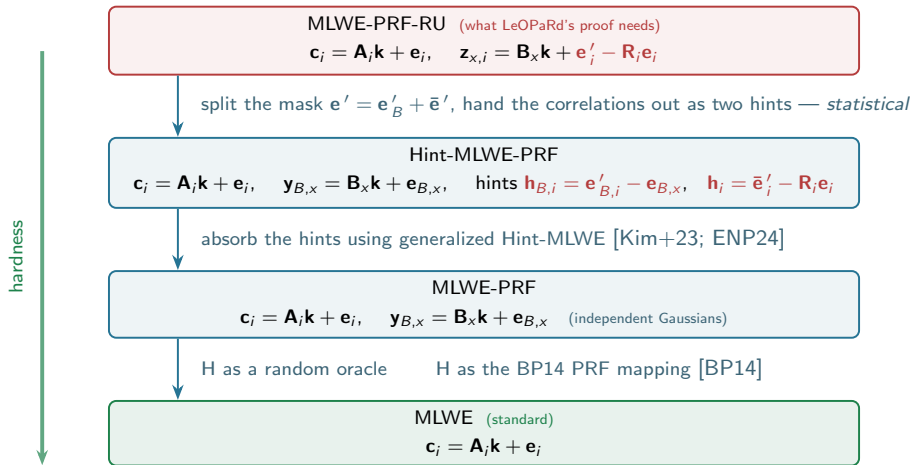
Hint-MLWE-PRF

$$\mathbf{c}_i = \mathbf{A}_i \mathbf{k} + \mathbf{e}_i, \quad \mathbf{y}_{B,x} = \mathbf{B}_x \mathbf{k} + \mathbf{e}_{B,x}, \quad \text{hints } \mathbf{h}_{B,i} = \mathbf{e}'_{B,i} - \mathbf{e}_{B,x}, \quad \mathbf{h}_i = \bar{\mathbf{e}}'_i - \mathbf{R}_i \mathbf{e}_i$$

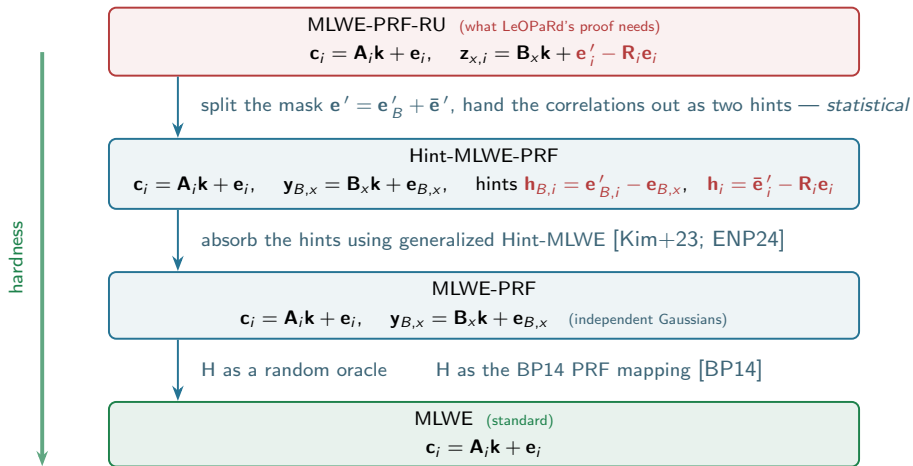
Reduction to MLWE



Reduction to MLWE



Reduction to MLWE



Every step is **tight** up to statistical losses — no rewinding, no complexity leveraging.

Proof sketch: the deficiency are hints

Step 1. The PRF error $\mathbf{e}' - \mathbf{Re}$ is *correlated* with the MLWE error $\mathbf{e}$. Split $\mathbf{e}' = \mathbf{e}'_B + \bar{\mathbf{e}}'$ and give the correlations away:

$$\mathbf{h}_B := \mathbf{e}'_B - \mathbf{e}_B, \quad \mathbf{h} := \bar{\mathbf{e}}' - \mathbf{Re} \quad \implies \quad \underbrace{\mathbf{y}_B + \mathbf{h}_B + \mathbf{h}}_{\text{simulated by } \mathcal{B}} = \mathbf{B}_x \mathbf{k} + \mathbf{e}' - \mathbf{Re}.$$

Proof sketch: the deficiency are hints

Step 1. The PRF error $\mathbf{e}' - \mathbf{Re}$ is *correlated* with the MLWE error $\mathbf{e}$. Split $\mathbf{e}' = \mathbf{e}'_B + \bar{\mathbf{e}}'$ and *give* the correlations away:

$$\mathbf{h}_B := \mathbf{e}'_B - \mathbf{e}_B, \quad \mathbf{h} := \bar{\mathbf{e}}' - \mathbf{Re} \quad \implies \quad \underbrace{\mathbf{y}_B + \mathbf{h}_B + \mathbf{h}}_{\text{simulated by } \mathcal{B}} = \mathbf{B}_x \mathbf{k} + \mathbf{e}' - \mathbf{Re}.$$

Step 2. Conditioned on a hint $\mathbf{h} = \bar{\mathbf{e}}' - \mathbf{Re}$, the error $\mathbf{e}$ is *still* a discrete Gaussian — re-centred, with covariance $\Sigma_0 = (\Sigma_1^{-1} + \frac{1}{s_1^2} \mathbf{R}^\top \mathbf{R})^{-1} \Rightarrow$ the reduction *samples the hints itself*, then corrects its MLWE-PRF samples.

Proof sketch: the deficiency are hints

Step 1. The PRF error $\mathbf{e}' - \mathbf{Re}$ is *correlated* with the MLWE error $\mathbf{e}$. Split $\mathbf{e}' = \mathbf{e}'_B + \bar{\mathbf{e}}'$ and *give* the correlations away:

$$\mathbf{h}_B := \mathbf{e}'_B - \mathbf{e}_B, \quad \mathbf{h} := \bar{\mathbf{e}}' - \mathbf{Re} \quad \implies \quad \underbrace{\mathbf{y}_B + \mathbf{h}_B + \mathbf{h}}_{\text{simulated by } \mathcal{B}} = \mathbf{B}_x \mathbf{k} + \mathbf{e}' - \mathbf{Re}.$$

Step 2. Conditioned on a hint $\mathbf{h} = \bar{\mathbf{e}}' - \mathbf{Re}$, the error $\mathbf{e}$ is *still* a discrete Gaussian — re-centred, with covariance $\Sigma_0 = (\Sigma_1^{-1} + \frac{1}{s_1^2} \mathbf{R}^\top \mathbf{R})^{-1} \Rightarrow$ the reduction *samples the hints itself*, then corrects its MLWE-PRF samples.

- New vs. Hint-MLWE [Kim+23; ENP24]: *non-square, adaptively chosen* hint matrices $\mathbf{R}_i$, and PRF samples w.r.t. *non-random* $\mathbf{B}_x = \mathbf{H}(x)$ — all Q_x^∞ hints on the same x hit the *same* error $\mathbf{e}_B$, a fixed identity-block hint matrix.

Proof sketch: the deficiency are hints

Step 1. The PRF error $\mathbf{e}' - \mathbf{R}\mathbf{e}$ is *correlated* with the MLWE error $\mathbf{e}$. Split $\mathbf{e}' = \mathbf{e}'_B + \bar{\mathbf{e}}'$ and *give* the correlations away:

$$\mathbf{h}_B := \mathbf{e}'_B - \mathbf{e}_B, \quad \mathbf{h} := \bar{\mathbf{e}}' - \mathbf{R}\mathbf{e} \quad \implies \quad \underbrace{\mathbf{y}_B + \mathbf{h}_B + \mathbf{h}}_{\text{simulated by } \mathcal{B}} = \mathbf{B}_x \mathbf{k} + \mathbf{e}' - \mathbf{R}\mathbf{e}.$$

Step 2. Conditioned on a hint $\mathbf{h} = \bar{\mathbf{e}}' - \mathbf{R}\mathbf{e}$, the error $\mathbf{e}$ is *still* a discrete Gaussian — re-centred, with covariance $\Sigma_0 = (\Sigma_1^{-1} + \frac{1}{\bar{s}_1^2} \mathbf{R}^\top \mathbf{R})^{-1} \Rightarrow$ the reduction *samples the hints itself*, then corrects its MLWE-PRF samples.

- New vs. Hint-MLWE [Kim+23; ENP24]: *non-square, adaptively chosen* hint matrices $\mathbf{R}_i$, and PRF samples w.r.t. *non-random* $\mathbf{B}_x = \mathbf{H}(x)$ — all Q_x^∞ hints on the same x hit the *same* error $\mathbf{e}_B$, a fixed identity-block hint matrix.

Two masking conditions

$$\bar{s}_1/s \geq \eta \beta_r d \sqrt{hN}$$

$$s_{1,B}/s_B \geq \eta \sqrt{Q_x^\infty}$$

widths: $\mathbf{e} \sim s$, $\mathbf{e}_B \sim s_B$, $\bar{\mathbf{e}}' \sim \bar{s}_1$, $\mathbf{e}'_B \sim s_{1,B}$; $\eta \approx 1$ (smoothing); $\|\mathbf{R}\|_\infty \leq \beta_r$, $\mathbf{R} \in R^{h \times N}$, $d = \text{ring dim.}$

Proof sketch: the deficiency are hints

Step 1. The PRF error $\mathbf{e}' - \mathbf{Re}$ is *correlated* with the MLWE error $\mathbf{e}$. Split $\mathbf{e}' = \mathbf{e}'_B + \bar{\mathbf{e}}'$ and *give* the correlations away:

$$\mathbf{h}_B := \mathbf{e}'_B - \mathbf{e}_B, \quad \mathbf{h} := \bar{\mathbf{e}}' - \mathbf{Re} \quad \implies \quad \underbrace{\mathbf{y}_B + \mathbf{h}_B + \mathbf{h}}_{\text{simulated by } \mathcal{B}} = \mathbf{B}_x \mathbf{k} + \mathbf{e}' - \mathbf{Re}.$$

Step 2. Conditioned on a hint $\mathbf{h} = \bar{\mathbf{e}}' - \mathbf{Re}$, the error $\mathbf{e}$ is *still* a discrete Gaussian — re-centred, with covariance $\Sigma_0 = (\Sigma_1^{-1} + \frac{1}{s_1^2} \mathbf{R}^\top \mathbf{R})^{-1} \Rightarrow$ the reduction *samples the hints itself*, then corrects its MLWE-PRF samples.

- New vs. Hint-MLWE [Kim+23; ENP24]: *non-square, adaptively chosen* hint matrices $\mathbf{R}_i$, and PRF samples w.r.t. *non-random* $\mathbf{B}_x = \mathbf{H}(x)$ — all Q_x^∞ hints on the same x hit the *same* error $\mathbf{e}_B$, a fixed identity-block hint matrix.

Two masking conditions

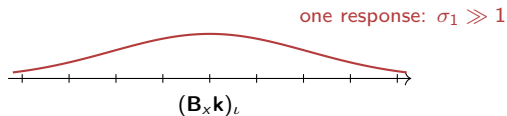
$$\bar{s}_1/s \geq \eta \beta_r d \sqrt{hN} \qquad s_{1,B}/s_B \geq \eta \sqrt{Q_x^\infty}$$

widths: $\mathbf{e} \sim s$, $\mathbf{e}_B \sim s_B$, $\bar{\mathbf{e}}' \sim \bar{s}_1$, $\mathbf{e}'_B \sim s_{1,B}$; $\eta \approx 1$ (smoothing); $\|\mathbf{R}\|_\infty \leq \beta_r$, $\mathbf{R} \in R^{h \times N}$, $d = \text{ring dim.}$

The mask $\mathbf{e}'$ must dominate (i) the hint matrix and (ii) the number of hints on the *same* x .

Is the $\sqrt{Q_x^\infty}$ condition necessary? (yes, an averaging attack)

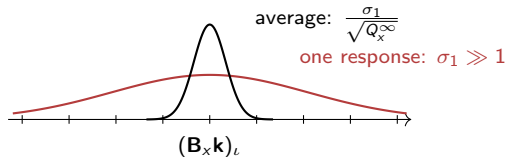
- Query the *same* x a total of Q_x^∞ times, unblind:
 $\mathbf{z}_j = \mathbf{B}_x \mathbf{k} + \mathbf{e}'_j - \mathbf{R}_j \mathbf{e}_j.$



Is the $\sqrt{Q_x^\infty}$ condition necessary? (yes, an averaging attack)

- Query the *same* x a total of Q_x^∞ times, unblind:
 $\mathbf{z}_j = \mathbf{B}_x \mathbf{k} + \mathbf{e}'_j - \mathbf{R}_j \mathbf{e}_j$.
- Average over the *rationals*:

$$\bar{\mathbf{z}} = \frac{1}{Q_x^\infty} \sum_j \mathbf{z}_j = \mathbf{B}_x \mathbf{k} + \bar{\mathbf{e}}, \quad \sigma(\bar{\mathbf{e}}) \approx \frac{\sigma_1}{\sqrt{Q_x^\infty}}.$$

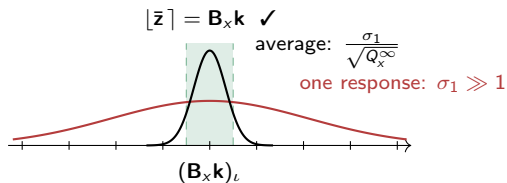


Is the $\sqrt{Q_x^\infty}$ condition necessary? (yes, an averaging attack)

- Query the *same* x a total of Q_x^∞ times, unblind:
 $\mathbf{z}_j = \mathbf{B}_x \mathbf{k} + \mathbf{e}'_j - \mathbf{R}_j \mathbf{e}_j$.
- Average over the *rationals*:

$$\bar{\mathbf{z}} = \frac{1}{Q_x^\infty} \sum_j \mathbf{z}_j = \mathbf{B}_x \mathbf{k} + \bar{\mathbf{e}}, \quad \sigma(\bar{\mathbf{e}}) \approx \frac{\sigma_1}{\sqrt{Q_x^\infty}}.$$

- Once $\sigma(\bar{\mathbf{e}}) < \frac{1}{2}$: rounding $\bar{\mathbf{z}}$ coordinate-wise reveals $\mathbf{B}_x \mathbf{k}$ **exactly** — then solve for $\mathbf{k}$.

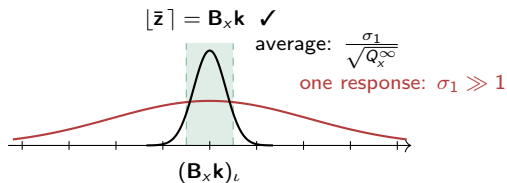


Is the $\sqrt{Q_x^\infty}$ condition necessary? (yes, an averaging attack)

- Query the *same* x a total of Q_x^∞ times, unblind:
 $\mathbf{z}_j = \mathbf{B}_x \mathbf{k} + \mathbf{e}'_j - \mathbf{R}_j \mathbf{e}_j$.
- Average over the rationals:

$$\bar{\mathbf{z}} = \frac{1}{Q_x^\infty} \sum_j \mathbf{z}_j = \mathbf{B}_x \mathbf{k} + \bar{\mathbf{e}}, \quad \sigma(\bar{\mathbf{e}}) \approx \frac{\sigma_1}{\sqrt{Q_x^\infty}}.$$

- Once $\sigma(\bar{\mathbf{e}}) < \frac{1}{2}$: rounding $\bar{\mathbf{z}}$ coordinate-wise reveals $\mathbf{B}_x \mathbf{k}$ *exactly* — then solve for $\mathbf{k}$.



Necessary vs. sufficient

attack $\Rightarrow$ need $\sigma_1/\sigma \gtrsim \sqrt{Q_x^\infty}/c$

our reduction needs $s_{1,B}/s_B \geq \eta \sqrt{Q_x^\infty}$

σ_1, σ = standard deviations of the mask $\mathbf{e}'_j$ and the MLWE error $\mathbf{e}_j$ ($\propto$ the widths s_1, s); $c > 1$ a small constant

From the assumption to the protocol (pseudorandomness, POPRF)

The simulator must answer a **malicious client** without using $\mathbf{k}$ (we remove it in three hybrids):

what the client sees (changes **red**, then **settle**)

$$pk = \text{Com}(\mathbf{k})$$

$$\text{client: } \mathbf{c}_r = \text{Com}(\mathbf{R}), \mathbf{d}_x = \text{Com}(x), \mathbf{C}_x, \pi_c$$

$$\pi_s \leftarrow \text{NIZK.P}(\text{stmt}; \mathbf{k}, \mathbf{e}_s, \mathbf{e}'_s, \rho_k)$$

$$\mathbf{v}_k = \mathbf{A}_r \mathbf{k} + \mathbf{e}_s$$

$$\mathbf{u}_x = \mathbf{C}_x \mathbf{k} + \mathbf{e}'_s$$

$$y = \text{RO}_z(t, x, [\mathbf{u}_x - \mathbf{R}\mathbf{v}_k]_p, pk)$$

From the assumption to the protocol (pseudorandomness, POPRF)

The simulator must answer a **malicious client** without using $\mathbf{k}$ (we remove it in three hybrids):

what the client sees (changes **red**, then **settle**)

$$pk = \text{Com}(\mathbf{k})$$

$$\text{client: } \mathbf{c}_r = \text{Com}(\mathbf{R}), \mathbf{d}_x = \text{Com}(x), \mathbf{C}_x, \pi_c$$

$$\pi_s \leftarrow \text{NIZK}.\mathcal{S}(\text{stmt})$$

$$\mathbf{v}_k = \mathbf{A}_r \mathbf{k} + \mathbf{e}_s$$

$$\mathbf{u}_x = \mathbf{C}_x \mathbf{k} + \mathbf{e}'_s$$

$$y = \text{RO}_z(t, x, [\mathbf{u}_x - \mathbf{R}\mathbf{v}_k]_p, pk)$$

- Hybrid 1: simulate π_s
zero-knowledge of NIZK

From the assumption to the protocol (pseudorandomness, POPRF)

The simulator must answer a **malicious client** without using $\mathbf{k}$ (we remove it in three hybrids):

what the client sees (changes **red**, then **settle**)

$$pk = \text{Com}(\mathbf{0})$$

$$\text{client: } \mathbf{c}_r = \text{Com}(\mathbf{R}), \mathbf{d}_x = \text{Com}(x), \mathbf{C}_x, \pi_c$$

$$\pi_s \leftarrow \text{NIZK}.\mathcal{S}(\text{stmt})$$

$$\mathbf{v}_k = \mathbf{A}_r \mathbf{k} + \mathbf{e}_s$$

$$\mathbf{u}_x = \mathbf{C}_x \mathbf{k} + \mathbf{e}'_s$$

$$y = \text{RO}_z(t, x, \lfloor \mathbf{u}_x - \mathbf{R} \mathbf{v}_k \rfloor_p, pk)$$

- Hybrid 1: simulate π_s
zero-knowledge of NIZK
- Hybrid 2: commit to $\mathbf{0}$ in pk
hiding of the commitment

From the assumption to the protocol (pseudorandomness, POPRF)

The simulator must answer a **malicious client** without using $\mathbf{k}$ (we remove it in three hybrids):

what the client sees (changes **red**, then **settle**)

$$pk = \text{Com}(\mathbf{0})$$

$$\text{client: } \mathbf{c}_r = \text{Com}(\mathbf{R}), \mathbf{d}_x = \text{Com}(x), \mathbf{C}_x, \pi_c$$

$$\pi_s \leftarrow \text{NIZK}.\mathcal{S}(\text{stmt})$$

$$\mathbf{v}_k = \mathbf{c} \leftarrow \text{SampMLWEcom}(\mathbf{R})$$

$$\mathbf{u}_x = \mathbf{z} + \mathbf{R}\mathbf{c}, \mathbf{z} \leftarrow \text{SampPRF-RU}(\mathbf{A}_r, x)$$

$$y = \text{RO}_z(t, x, \lfloor \mathbf{u}_x - \mathbf{R}\mathbf{v}_k \rfloor_p, pk)$$

- Hybrid 1: simulate π_s
zero-knowledge of NIZK
- Hybrid 2: commit to $\mathbf{0}$ in pk
hiding of the commitment
- Hybrid 3: **extract** $(\mathbf{R}, x)$ from $(\mathbf{c}_r, \mathbf{d}_x)$,
answer via the MLWE-PRF-RU oracles
extractable *commitments* (Regev encryption [Reg05])

From the assumption to the protocol (pseudorandomness, POPRF)

The simulator must answer a **malicious client** without using $\mathbf{k}$ (we remove it in three hybrids):

what the client sees (changes **red**, then **settle**)

$$pk = \text{Com}(\mathbf{0})$$

$$\text{client: } \mathbf{c}_r = \text{Com}(\mathbf{R}), \mathbf{d}_x = \text{Com}(x), \mathbf{C}_x, \pi_c$$

$$\pi_s \leftarrow \text{NIZK}.\mathcal{S}(\text{stmt})$$

$$\mathbf{v}_k = \mathbf{c} \leftarrow \text{SampMLWEcom}(\mathbf{R})$$

$$\mathbf{u}_x = \mathbf{z} + \mathbf{R}\mathbf{c}, \mathbf{z} \leftarrow \text{SampPRF-RU}(\mathbf{A}_r, x)$$

$$y = \text{RO}_z(t, x, \lfloor \mathbf{u}_x - \mathbf{R}\mathbf{v}_k \rfloor_p, pk)$$

- Hybrid 1: simulate π_s
zero-knowledge of NIZK
- Hybrid 2: commit to $\mathbf{0}$ in pk
hiding of the commitment
- Hybrid 3: **extract** $(\mathbf{R}, x)$ from $(\mathbf{c}_r, \mathbf{d}_x)$,
answer via the MLWE-PRF-RU oracles
extractable *commitments* (Regev encryption [Reg05])
- $b = 0$: $(\mathbf{v}_k, \mathbf{u}_x)$ uniform $\Rightarrow$ view
independent of $\mathbf{k}$ ✓

From the assumption to the protocol (pseudorandomness, POPRF)

The simulator must answer a **malicious client** without using $\mathbf{k}$ (we remove it in three hybrids):

what the client sees (changes **red**, then **settle**)

$$pk = \text{Com}(\mathbf{0})$$

$$\text{client: } \mathbf{c}_r = \text{Com}(\mathbf{R}), \mathbf{d}_x = \text{Com}(x), \mathbf{C}_x, \pi_c$$

$$\pi_s \leftarrow \text{NIZK.S}(\text{stmt})$$

$$\mathbf{v}_k = \mathbf{c} \leftarrow \text{SampMLWEcom}(\mathbf{R})$$

$$\mathbf{u}_x = \mathbf{z} + \mathbf{R}\mathbf{c}, \mathbf{z} \leftarrow \text{SampPRF-RU}(\mathbf{A}_r, x)$$

$$y = \text{RO}_z(t, x, \lfloor \mathbf{u}_x - \mathbf{R}\mathbf{v}_k \rfloor_p, pk)$$

- Hybrid 1: simulate π_s
zero-knowledge of NIZK
- Hybrid 2: commit to $\mathbf{0}$ in pk
hiding of the commitment
- Hybrid 3: **extract** $(\mathbf{R}, x)$ from $(\mathbf{c}_r, \mathbf{d}_x)$,
answer via the MLWE-PRF-RU oracles
extractable *commitments* (Regev encryption [Reg05])
- $b = 0$: $(\mathbf{v}_k, \mathbf{u}_x)$ uniform $\Rightarrow$ view
independent of $\mathbf{k}$ ✓

Malicious server (request privacy, POPRIV2) follows from **Knapsack MLWE**.

Concrete parameters and cost

Online communication, excluding NIZKs ($\approx$ 95-bit security)

	$\log q$	server	client
LeOPaRd (partial), $\kappa = 16$	43	0.34 KB	7.05 KB
LeOPaRd (partial), $\kappa = 32$	60	0.47 KB	13.19 KB
[AG24], $Q = 2^{16}$	143	1.80 KB	71.50 KB
[AG24], $Q = 2^{32}$	151	1.80 KB	75.50 KB

$\kappa =$ correctness/uniqueness parameter

Concrete parameters and cost

Online communication, excluding NIZKs ($\approx$ 95-bit security)

	$\log q$	server	client
LeOPaRd (partial), $\kappa = 16$	43	0.34 KB	7.05 KB
LeOPaRd (partial), $\kappa = 32$	60	0.47 KB	13.19 KB
[AG24], $Q = 2^{16}$	143	1.80 KB	71.50 KB
[AG24], $Q = 2^{32}$	151	1.80 KB	75.50 KB

$\kappa =$ correctness/uniqueness parameter

- No flooding $\Rightarrow \log q \approx 43$ – 93 instead of 143 – 169 — and communication is quadratic in $\log q$.

Concrete parameters and cost

Online communication, excluding NIZKs ($\approx$ 95-bit security)

	$\log q$	server	client
LeOPaRd (partial), $\kappa = 16$	43	0.34 KB	7.05 KB
LeOPaRd (partial), $\kappa = 32$	60	0.47 KB	13.19 KB
[AG24], $Q = 2^{16}$	143	1.80 KB	71.50 KB
[AG24], $Q = 2^{32}$	151	1.80 KB	75.50 KB

$\kappa =$ correctness/uniqueness parameter

- No flooding $\Rightarrow \log q \approx 43$ – 93 instead of 143 – 169 — and communication is quadratic in $\log q$.
- $\sim 90\%$ of the online cost is now the NIZKs (≈ 45 KB each, LaBRADOR-based; a batch of 64 queries needs one proof of 79 KB, $< 1.8\times$ a single one).

Conclusion

A post-quantum VPOPRF with ≈ 138 KB communication —
no noise flooding, no FHE, no straight-line extractability.

- A new interactive lattice assumption, MLWE-PRF-RU, with a reduction to MLWE.
- The $\sqrt{Q_x^\infty}$ loss is *necessary* (averaging attack) — our parameters are optimal up to a small constant.

Conclusion

A post-quantum VPOPRF with ≈ 138 KB communication —
no noise flooding, no FHE, no straight-line extractability.

- A new interactive lattice assumption, MLWE-PRF-RU, with a reduction to MLWE.
- The $\sqrt{Q_x^\infty}$ loss is *necessary* (averaging attack) — our parameters are optimal up to a small constant.

Open questions:

- An efficient threshold VPOPRF construction.
- Full QROM analysis.
- Universal composability (UC) security.

Conclusion

A post-quantum VPOPRF with ≈ 138 KB communication —
no noise flooding, no FHE, no straight-line extractability.

- A new interactive lattice assumption, MLWE-PRF-RU, with a [reduction to MLWE](#).
- The $\sqrt{Q_x^\infty}$ loss is *necessary* (averaging attack) — our parameters are optimal up to a small constant.

Open questions:

- An efficient threshold VPOPRF construction.
- Full QROM analysis.
- Universal composability (UC) security.

Thank you!



ia.cr/2024/1615

References I

- [AG24] M. R. Albrecht and K. D. Gür. “Verifiable Oblivious Pseudorandom Functions from Lattices: Practical-Ish and Thresholdisable”. In: *ASIACRYPT 2024, Part IV*. Ed. by K.-M. Chung and Y. Sasaki. Vol. 15487. LNCS. Springer, Singapore, Dec. 2024, pp. 205–237. DOI: 10.1007/978-981-96-0894-2_7.
- [Alb+21] M. R. Albrecht et al. “Round-Optimal Verifiable Oblivious Pseudorandom Functions from Ideal Lattices”. In: *PKC 2021, Part II*. Ed. by J. Garay. Vol. 12711. LNCS. Springer, Cham, May 2021, pp. 261–289. DOI: 10.1007/978-3-030-75248-4_10.
- [Alb+24] M. R. Albrecht et al. “Crypto Dark Matter on the Torus - Oblivious PRFs from Shallow PRFs and TFHE”. In: *EUROCRYPT 2024, Part VI*. Ed. by M. Joye and G. Leander. Vol. 14656. LNCS. Springer, Cham, May 2024, pp. 447–476. DOI: 10.1007/978-3-031-58751-1_16.
- [Beu+25] W. Beullens et al. “The 2Hash OPRF Framework and Efficient Post-quantum Instantiations”. In: *EUROCRYPT 2025, Part VIII*. Ed. by S. Fehr and P.-A. Fouque. Vol. 15608. LNCS. Springer, Cham, May 2025, pp. 332–362. DOI: 10.1007/978-3-031-91101-9_12.
- [Bol03] A. Boldyreva. “Threshold Signatures, Multisignatures and Blind Signatures Based on the Gap-Diffie-Hellman-Group Signature Scheme”. In: *PKC 2003*. Ed. by Y. Desmedt. Vol. 2567. LNCS. Springer, Berlin, Heidelberg, Jan. 2003, pp. 31–46. DOI: 10.1007/3-540-36288-6_3.
- [BP14] A. Banerjee and C. Peikert. “New and Improved Key-Homomorphic Pseudorandom Functions”. In: *CRYPTO 2014, Part I*. Ed. by J. A. Garay and R. Gennaro. Vol. 8616. LNCS. Springer, Berlin, Heidelberg, Aug. 2014, pp. 353–370. DOI: 10.1007/978-3-662-44371-2_20.

References II

- [BS23] W. Beullens and G. Seiler. “LaBRADOR: Compact Proofs for R1CS from Module-SIS”. In: *CRYPTO 2023, Part V*. Ed. by H. Handschuh and A. Lysyanskaya. Vol. 14085. LNCS. Springer, Cham, Aug. 2023, pp. 518–548. DOI: 10.1007/978-3-031-38554-4_17.
- [CHL22] S. Casacuberta, J. Hesse, and A. Lehmann. “SoK: Oblivious Pseudorandom Functions”. In: *2022 IEEE European Symposium on Security and Privacy*. IEEE Computer Society Press, June 2022, pp. 625–646. DOI: 10.1109/EuroSP53844.2022.00045.
- [CJ25] J. H. Cheon and D. Jang. “Cryptanalysis on Lightweight Verifiable Homomorphic Encryption”. In: *ASIACRYPT 2025, Part VII*. Ed. by G. Hanaoka and B.-Y. Yang. Vol. 16251. LNCS. Springer, Singapore, Dec. 2025, pp. 366–397. DOI: 10.1007/978-981-95-5122-4_12.
- [Dav+18] A. Davidson et al. “Privacy Pass: Bypassing Internet Challenges Anonymously”. In: *PoPETs 2018.3* (July 2018), pp. 164–180. DOI: 10.1515/popets-2018-0026.
- [ENP24] T. Espitau, G. Niot, and T. Prest. “Flood and Submerge: Distributed Key Generation and Robust Threshold Signature from Lattices”. In: *CRYPTO 2024, Part VII*. Ed. by L. Reyzin and D. Stebila. Vol. 14926. LNCS. Springer, Cham, Aug. 2024, pp. 425–458. DOI: 10.1007/978-3-031-68394-7_14.
- [FLÖ25] K. Friedrichs, A. Lehmann, and C. Özbay. “Game Changer: A Modular Framework for OPRF Security”. In: *ASIACRYPT 2025, Part VIII*. Ed. by G. Hanaoka and B.-Y. Yang. Vol. 16252. LNCS. Springer, Singapore, Dec. 2025, pp. 582–613. DOI: 10.1007/978-981-95-5125-5_19.

References III

- [Jar+16] S. Jarecki et al. “Highly-Efficient and Composable Password-Protected Secret Sharing (Or: How to Protect Your Bitcoin Wallet Online)”. In: *2016 IEEE European Symposium on Security and Privacy*. IEEE Computer Society Press, Mar. 2016, pp. 276–291. DOI: 10.1109/EuroSP.2016.30.
- [JKK14] S. Jarecki, A. Kiayias, and H. Krawczyk. “Round-Optimal Password-Protected Secret Sharing and T-PAKE in the Password-Only Model”. In: *ASIACRYPT 2014, Part II*. Ed. by P. Sarkar and T. Iwata. Vol. 8874. LNCS. Springer, Berlin, Heidelberg, Dec. 2014, pp. 233–253. DOI: 10.1007/978-3-662-45608-8_13.
- [JKX18] S. Jarecki, H. Krawczyk, and J. Xu. “OPAQUE: An Asymmetric PAKE Protocol Secure Against Pre-computation Attacks”. In: *EUROCRYPT 2018, Part III*. Ed. by J. B. Nielsen and V. Rijmen. Vol. 10822. LNCS. Springer, Cham, 2018, pp. 456–486. DOI: 10.1007/978-3-319-78372-7_15.
- [Kim+23] D. Kim et al. “Toward Practical Lattice-Based Proof of Knowledge from Hint-MLWE”. In: *CRYPTO 2023, Part V*. Ed. by H. Handschuh and A. Lysyanskaya. Vol. 14085. LNCS. Springer, Cham, Aug. 2023, pp. 549–580. DOI: 10.1007/978-3-031-38554-4_18.
- [Li+19] L. Li et al. “Protocols for Checking Compromised Credentials”. In: *ACM CCS 2019*. Ed. by L. Cavallaro et al. ACM Press, Nov. 2019, pp. 1387–1403. DOI: 10.1145/3319535.3354229.
- [LNP22] V. Lyubashevsky, N. K. Nguyen, and M. Plançon. “Lattice-Based Zero-Knowledge Proofs and Applications: Shorter, Simpler, and More General”. In: *CRYPTO 2022, Part II*. Ed. by Y. Dodis and T. Shrimpton. Vol. 13508. LNCS. Springer, Cham, Aug. 2022, pp. 71–101. DOI: 10.1007/978-3-031-15979-4_3.

References IV

- [Reg05] O. Regev. “On lattices, learning with errors, random linear codes, and cryptography”. In: *37th ACM STOC*. Ed. by H. N. Gabow and R. Fagin. ACM Press, May 2005, pp. 84–93. DOI: 10.1145/1060590.1060603.
- [Tya+22] N. Tyagi et al. “A Fast and Simple Partially Oblivious PRF, with Applications”. In: *EUROCRYPT 2022, Part II*. Ed. by O. Dunkelman and S. Dziembowski. Vol. 13276. LNCS. Springer, Cham, 2022, pp. 674–705. DOI: 10.1007/978-3-031-07085-3_23.
- [Yan+25] Y. Yang et al. “Gold OPRF: Post-Quantum Oblivious Power-Residue PRF”. In: *2025 IEEE Symposium on Security and Privacy*. Ed. by M. Blanton, W. Enck, and C. Nita-Rotaru. IEEE Computer Society Press, May 2025, pp. 259–278. DOI: 10.1109/SP61157.2025.00116.