Design and analysis of dual attacks in code- and lattice-based cryptography

PhD Defense, Inria Paris, September 30, 2025

Charles Meyer-Hilfiger, Irisa & Univ. Rennes

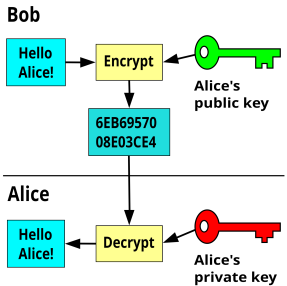
Under the supervision of Nicolas Sendrier and Jean-Pierre Tillich

Section 1

Introduction

Public-Key cryptography

Used for safe communication over insecure channel without pre-shared secret.



RSA, DH

↓

Hard computational problem

↑

Easily solved by quantum computer

Post-Quantum (Public-Key) cryptography

Lattice, Code, Multivariate, Isogenies

	Code-based	Lattice-based
Encryption	**HQC (NIST)**, McEliece, Bike, ...	**Kyber (NIST)**,...
Signature	SDiTH,...	**Dilithium (NIST)**,...
Security	**Decoding problem**	**Learning with Errors**

$\rightarrow$ **Hard** problem even for quantum computer

Complexity of best algorithms used to parametrize schemes.

Binary Decoding Problem

$$\text{Binary Linear code} \quad \rightarrow \quad \mathscr{C} = \{ \mathbf{mG} \ : \ \mathbf{m} \in \mathbb{F}_2^n \}$$

Decoding at a **small** distance t

- **Input:** $(\mathbf{G}, \mathbf{y} = \mathbf{c} + \mathbf{e}) \in \mathbb{F}_2^{k \times n} \times \mathbb{F}_2^n$ where $\mathbf{c} \in \mathscr{C}$ and $|\mathbf{e}| = t$
- **Output:** $\mathbf{e}$ such that $|\mathbf{e}| = t$ and $\mathbf{y} - \mathbf{e} \in \mathscr{C}$

Constraint $\mathbf{e}$ **small** Hamming weight $\rightarrow$ Make **problem hard**

Binary Decoding (Code)	Learning with Errors (Lattice)
$\mathbb{F}_2$	$\mathbb{F}_q$
Hamming weight	Euclidean norm

Hardness of decoding problem

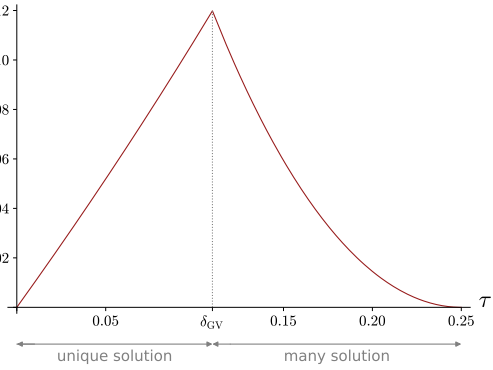


Figure: Complexity exponent of Prange algorithm $\alpha(\tau)$: Complexity $= 2^{\alpha\, n}$, $\tau \stackrel{\triangle}{=} \frac{t}{n}$ at rate $R = \frac{k}{n} = \frac{1}{2}$

δ_{GV} maximum distance where typically unique solution

Setting for Dual Attacks

Dual code

$$\mathscr{C}^{\perp} = \{\mathbf{h} \in \mathbb{F}_2^n \,:\, \langle \mathbf{h}, \mathbf{c} \rangle = 0 \quad \forall \mathbf{c} \in \mathscr{C}\} \qquad \text{with} \qquad \langle \mathbf{x}, \mathbf{y} \rangle = \sum x_i\, y_i \pmod{q}$$

Compute dual vector $\mathbf{h} \in \mathscr{C}^{\perp}$

Observation:

Given $\mathbf{y} = \mathbf{c} + \mathbf{e}$ $\rightarrow \langle \mathbf{y}, \mathbf{h} \rangle = \langle \mathbf{c} + \mathbf{e}, \mathbf{h} \rangle = \langle \mathbf{e}, \mathbf{h} \rangle$

Exploit:

More biased toward 0 as $|\mathbf{e}|$ and $|\mathbf{h}|$ smaller.

Statistical decoding (Al-Jabri 2001)

Compute $\mathbf{h} \in \mathscr{C}^{\perp}$ of low weight $|\mathbf{h}| = w$ such that $\mathbf{h}_1 = 1$:

$$\langle \mathbf{y}, \mathbf{h} \rangle = \langle \mathbf{e}, \mathbf{h} \rangle = \sum \mathbf{e}_i \mathbf{h}_i = \mathbf{e}_1 + \sum \mathbf{e}_i \mathbf{h}_i \sim \begin{cases} \text{Bernouilli}\left(\frac{1-\varepsilon}{2}\right) & \text{if } \mathbf{e}_1 = 0 \\ \text{Bernouilli}\left(\frac{1+\varepsilon}{2}\right) & \text{if } \mathbf{e}_1 = 1 \end{cases}$$

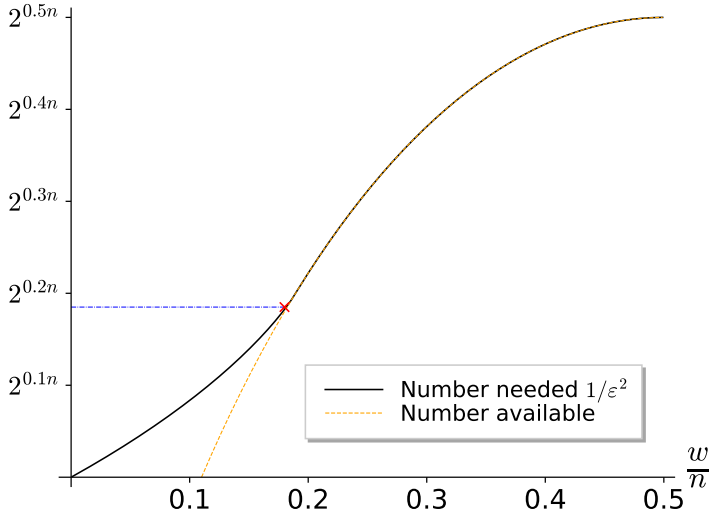
Compute N such dual vectors $\rightarrow$ Decide with majority voting

How big must N be to make good decision?

Assumptions	**Estimating** ε	**Independence**
	$\mathbf{h} \hookleftarrow \{ \mathbf{h} \in \mathbb{F}_2^n \ : \ |\mathbf{h}| = w \}$ $\downarrow$ Bias $\varepsilon \approx \delta_w^n(t)$	$\langle \mathbf{y}, \mathbf{h} \rangle's$ are **Independent** $\downarrow$ $N > \frac{1}{\varepsilon^2}$

$$N > \frac{1}{\delta_w^n(t)^2}$$

Lower bound on the complexity of the algorithm



State of the art: performance of some decoders

- Information Set Decoders (ISD) [P62,D89,MMT11,BJMM12, BM17, BM18]
- Dual attacks : Statistical decoding (Al-Jabri 2001)

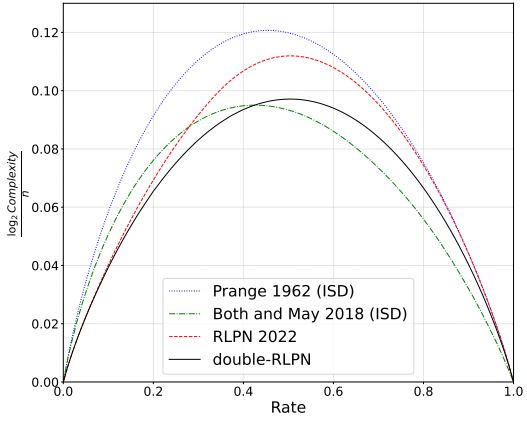
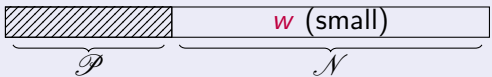


Figure: Rate $= k/n$. Decoding distance t at Gilbert-Varshamov. Complexity is in $2^{\alpha n}$.

Main ingredient : Splitting strategy (Reducing to LPN)

→ Suggested by [DT17]. All modern lattice-based dual attacks [Alb17,EJK20,GJ21,Matzov22]

- Split support in complementary part $\mathscr{P}$ and $\mathscr{N}$ → Recover $\mathbf{e}_{\mathscr{P}}$?
- Compute dual vector $\mathbf{h} =$ 

$$\rightarrow \langle \mathbf{y}, \mathbf{h} \rangle = \langle \mathbf{e}, \mathbf{h} \rangle = \langle \underbrace{\mathbf{e}_{\mathscr{P}}}_{\text{secret}}, \mathbf{h}_{\mathscr{P}} \rangle + \underbrace{\langle \mathbf{e}_{\mathscr{N}}, \mathbf{h}_{\mathscr{N}} \rangle}_{\text{noise: biased to 0}}$$

LPN Problem

- **Input:** Many samples $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e)$
  - $\mathbf{s} \in \mathbb{F}_2^s$ fixed secret
  - $\mathbf{a}$ taken at random in $\mathbb{F}_2^s$
  - $e \sim \mathrm{Bern}(p)$
- **Output:** $\mathbf{s}$

N dual vectors → N LPN samples

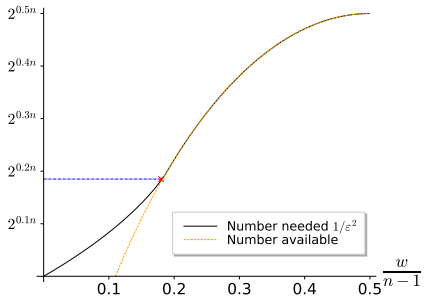
$$(\mathbf{a}, \langle \mathbf{s}, \mathbf{a} \rangle + e) \text{ w.t } \begin{cases} \mathbf{a} = \mathbf{h}_{\mathscr{P}} & \in \mathbb{F}_2^{|\mathscr{P}|} \\ \mathbf{s} = \mathbf{e}_{\mathscr{P}} \\ e = \langle \mathbf{e}_{\mathscr{N}}, \mathbf{h}_{\mathscr{N}} \rangle \end{cases}$$

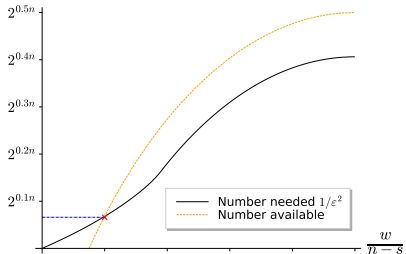
Hardness of this LPN problem

Supposing **Independence assumption**

$$N \geq \frac{1}{\mathrm{bias}\left(\langle \mathbf{e}_{\mathscr{N}}, \mathbf{h}_{\mathscr{N}} \rangle\right)^2} \quad \rightarrow \quad \text{Can recover secret } \mathbf{e}_{\mathscr{P}}$$

$$+ \text{ approximate bias by } \delta_w\left(|\mathbf{e}_{\mathscr{N}}|\right)$$





Code-Based Contribution (1)

Significant improvement of statistical decoding

Decoding Problem

Complexity exponent α

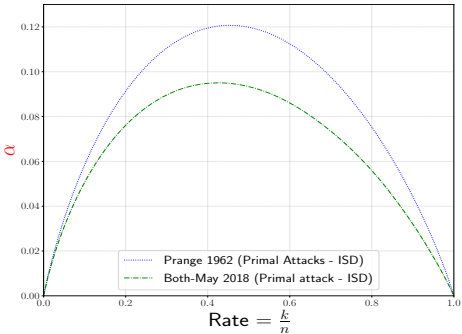


Figure: Complexity $2^{\alpha\, n}$

Code-Based Contribution (1)

Significant improvement of statistical decoding

Decoding Problem
↓
Reduced to LPN [CDMT22]

Complexity exponent α

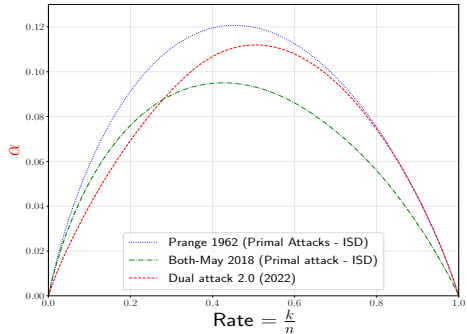


Figure: Complexity $2^{\alpha n}$

$\rightarrow$ Big gain for rather small rates

Code-Based Contribution (1)

Significant improvement of statistical decoding

Decoding Problem
↓
Reduced to **sparse** LPN [CDMT22]
↓
Reduced to plain LPN of smaller dim. [CDMT24]

Complexity exponent α

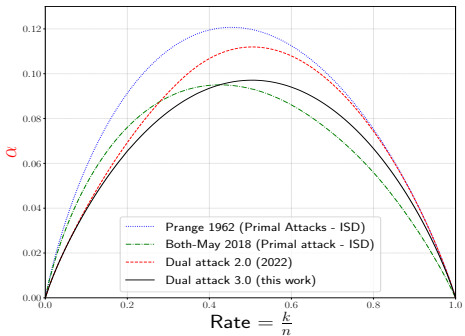


Figure: Complexity $2^{\alpha\, n}$

$\rightarrow$Beats state of art for $R < 0.42$

Code-Based Contribution (2)

Contribution

New tools and tweaks to analyze dual attacks

- In [CDMT22] : Independence assumption not always experimentally accurate.
- Ducas & Pulles 2023 : Disprove independence assumptions for lattice-based dual attacks.

[MT23]

~~**Independence Assumptions**~~

↓ Replaced by

Poisson model

The weight enumerator of a random linear code is a Poisson variable.

To appear on eprint 2025

Fully prove, without any model, dual attacks

Outline of the presentation

Our dual attacks and their analysis

- Reducing decoding to solving an LPN problem
- LPN solvers
  - RLPN : FFT (Leveil & Fouque 2007)
  - doubleRLPN : Reduction sparse to plain LPN (Guo & Johansson 2014)
- Analysis
  - How the analysis is carried currently. Usage of Poisson model.
  - Fully provable variant.
- Quick comparison with lattice-based attacks

Section 2

Our dual attacks

Main ingredient : Splitting strategy (Reducing to LPN)

$$\mathbf{y} = \mathbf{c} + \mathbf{e}$$

- Split support in complementary part $\mathscr{P}$ and $\mathscr{N}$ $\rightarrow$ Recover $\mathbf{e}_{\mathscr{P}}$?
- Compute dual vector $\mathbf{h} =$ 

$$\rightarrow \langle \mathbf{y}, \mathbf{h} \rangle = \langle \mathbf{e}, \mathbf{h} \rangle = \langle \underbrace{\mathbf{e}_{\mathscr{P}}}_{\text{secret}}, \mathbf{h}_{\mathscr{P}} \rangle + \underbrace{\langle \mathbf{e}_{\mathscr{N}}, \mathbf{h}_{\mathscr{N}} \rangle}_{\text{noise: biased to 0}}$$

LPN Problem

- **Input:** Many samples $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e)$
  - $\mathbf{s} \in \mathbb{F}_2^{\mathbf{s}}$ fixed secret
  - $\mathbf{a}$ taken at random in $\mathbb{F}_2^{\mathbf{s}}$
  - $e \sim \mathrm{Bern}(p)$
- **Output:** $\mathbf{s}$

N dual vectors $\rightarrow$ N LPN samples

$$(\mathbf{a}, \langle \mathbf{s}, \mathbf{a} \rangle + e) \text{ w.t } \begin{cases} \mathbf{a} = \mathbf{h}_{\mathscr{P}} & \in \mathbb{F}_2^{|\mathscr{P}|} \\ \mathbf{s} = \mathbf{e}_{\mathscr{P}} \\ e = \langle \mathbf{e}_{\mathscr{N}}, \mathbf{h}_{\mathscr{N}} \rangle \end{cases}$$

Key quantity : score function

$$\text{LPN sample } \langle \mathbf{y}, \mathbf{h} \rangle = \langle \mathbf{e}_{\mathscr{P}}, \mathbf{h}_{\mathscr{P}} \rangle + \langle \mathbf{h}_{\mathscr{N}}, \mathbf{e}_{\mathscr{N}} \rangle$$

$$\langle \mathbf{y}, \mathbf{h} \rangle - \langle \mathbf{e}_{\mathscr{P}}, \mathbf{h}_{\mathscr{P}} \rangle = \langle \mathbf{h}_{\mathscr{N}}, \mathbf{e}_{\mathscr{N}} \rangle \rightarrow \text{Biased toward } 0$$

Score function

For $\mathbf{x} \in \mathbb{F}_2^{|\mathscr{P}|}$ score function

$$\mathbf{F}(\mathbf{x}) \stackrel{\triangle}{=} \sum_{\mathbf{h}} (-1)^{\langle \mathbf{y}, \mathbf{h} \rangle - \langle \mathbf{x}, \mathbf{h}_{\mathscr{P}} \rangle}$$

General dual attack framework

Given $\mathscr{C}$ and $\mathbf{y} = \mathbf{c} + \mathbf{e} \rightarrow$ Goal recover $\mathbf{e}$:

- Choose subset $\mathscr{P}$ and $\mathscr{N}$ at random
- Compute N dual vector $\mathbf{h} \in \mathscr{C}^{\perp}$ s.t $|\mathbf{h}_{\mathscr{N}}| = w$
  - With technique taken from ISD's.
  - Get LPN samples $\langle \mathbf{y}, \mathbf{h} \rangle = \langle \mathbf{e}_{\mathscr{P}}, \mathbf{h}_{\mathscr{P}} \rangle + \langle \mathbf{h}_{\mathscr{N}}, \mathbf{e}_{\mathscr{N}} \rangle$
- Solve LPN problem : return set of candidate $\mathbf{x}$ for $\mathbf{e}_{\mathscr{P}}$
  - Compute score function $\mathbf{F}$
  - Keep candidates $\mathbf{x}$ such that $\mathbf{F}(\mathbf{x}) > T$ big enough
- Test each candidate $\mathbf{x}$ for $\mathbf{e}_{\mathscr{P}}$:
  - Solve smaller decoding problem at length $n - |\mathscr{P}|$, dimension $k - |\mathscr{P}|$.
  - Exponential cost
  - When $\mathbf{x} = \mathbf{e}_{\mathscr{P}}$ returns the rest of the error $\mathbf{e}_{\mathscr{N}}$

Next slides

Algorithm $\rightarrow$ How to solve the LPN problem: Computing the set of candidates efficiently?

Analysis $\rightarrow$ How to estimate the number of false candidates $\mathbf{x} \neq \mathbf{e}_{\mathscr{P}}$?

Subsection 1

LPN solvers

LPN solvers (1) : FFT $\rightarrow$ RLPN

We have computed N dual vectors $\mathbf{h}$. Compute for each $\mathbf{x} \in \mathbb{F}_2^{|\mathscr{P}|}$

$$\mathbf{F}(\mathbf{x}) \stackrel{\triangle}{=} \sum_{\mathbf{h}} (-1)^{\langle \mathbf{y}, \mathbf{h} \rangle - \langle \mathbf{x}, \mathbf{h}_{\mathscr{P}} \rangle}$$

Naive search

$$2^{|\mathscr{P}|} \times N$$

Leveil & Fouque 2006

Use a Fast Fourier Transform

$$|\mathscr{P}|\, 2^{|\mathscr{P}|} + N$$

$\rightarrow$ Exponential speed-up

This FFT LPN solver inside our framework $\rightarrow$ RLPN [CDMT22]

LPN solvers (2) : Remark

$\mathbf{e}_{\mathscr{P}}$ is sparse and yet FFT computes $F(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{F}_2^{|\mathscr{P}|}$

General approach : dimension reduction

LPN sample : $\left(\underset{\mathbb{F}_2^{|\mathscr{P}|}}{\mathbf{a}} \ , \ \langle \overset{\text{sparse}}{\mathbf{s}} , \mathbf{a} \rangle + e \right)$ $\xrightarrow[\text{Increase Noise}]{\text{Lower Dimension}}$ $\left(\underset{\mathbb{F}_2^{\leq |\mathscr{P}|}}{\mathbf{a}'} \ , \ \langle \overset{\text{plain}}{\mathbf{s}'} , \mathbf{a}' \rangle + e' \right)$

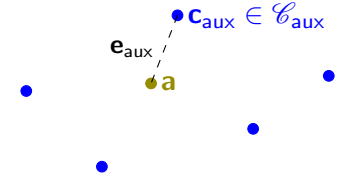
New reduced plain LPN problem $\rightarrow$ solve with FFT $\rightarrow$ doubleRLPN

LPN solvers (3) : Reduction from sparse to plain LPN → doubleRLPN

→ Technique by Guo & Johansson (2014)

Linear code $\mathscr{C}_{\text{aux}} \subset \mathbb{F}_2^{|\mathscr{P}|}$

$\mathscr{C}_{\text{aux}} = \{\mathbf{m}_{\text{aux}}\mathbf{G}_{\text{aux}} : \mathbf{m}_{\text{aux}} \in \mathbb{F}_2^{\dim(\mathscr{C}_{\text{aux}})}\}$



$$\mathbf{a} = \mathbf{c}_{\text{aux}} + \underbrace{\mathbf{e}_{\text{aux}}}_{\text{short}}$$

$$\langle \mathbf{s}, \mathbf{a} \rangle + e = \langle \mathbf{s}, \mathbf{c}_{\text{aux}} \rangle + \underbrace{\langle \mathbf{s}, \mathbf{e}_{\text{aux}} \rangle + e}_{e' \text{ new noise}}$$

$$\langle \mathbf{s}, \mathbf{c}_{\text{aux}} \rangle = \langle \mathbf{s}, \mathbf{m}_{\text{aux}}\mathbf{G}_{\text{aux}} \rangle = \langle \mathbf{s}\mathbf{G}_{\text{aux}}^{\top}, \mathbf{m}_{\text{aux}} \rangle$$

Sample space $\mathbb{F}_2^{|\mathscr{P}|} \to \mathbb{F}_2^{\dim(\mathscr{C}_{\text{aux}})}$ is smaller!

Subsection 2

Analysis : estimating the number of false candidates

Key question for the analysis

Set of candidates: $\mathbf{x}$ such that $\mathbf{F}(\mathbf{x}) > T$

Under the condition that

$$N > \frac{n}{\delta^2}$$

Easy to prove $\quad \mathbb{P}\left(\mathbf{F}(\mathbf{x}) > \mathbb{E}\left(\mathbf{F}\left(\mathbf{e}_{\mathscr{D}}\right)\right)\right) \leq \frac{1}{\mathrm{poly}(n)}$

Intractable $\quad \mathbb{P}\left(\mathbf{F}(\mathbf{x}) > \mathbb{E}\left(\mathbf{F}\left(\mathbf{e}_{\mathscr{D}}\right)\right)\right) \leq 2^{-\Theta(n)}$

Complexity of the algorithm

$\downarrow$

Estimating number of false candidates $\mathbf{x}$ s.t $\mathbf{F}(\mathbf{x}) > T$

Key question

What is the **exponential tail** distribution of $\mathbf{F}(\mathbf{x})$?

Distribution of the score function

$$\mathbb{P}\left(\mathbf{F}\left(\mathbf{x}\right) > T\right)$$

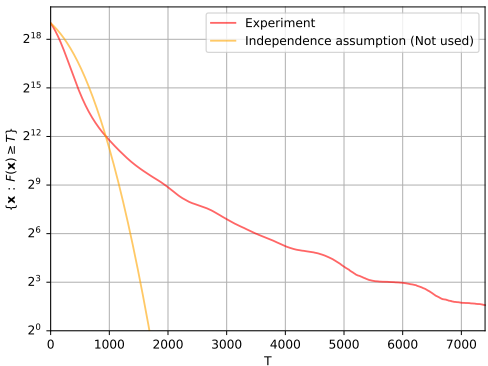


Figure: Distribution score function

Independence Assumptions

The terms in $\mathbf{F}\left(\mathbf{x}\right) = \sum_{\mathbf{h}} (-1)^{\langle \mathbf{y},\mathbf{h} \rangle - \langle \mathbf{x},\mathbf{h}_{\mathscr{P}} \rangle}$ are independent variables.

Under independence assumptions if

$$N > \frac{n}{\delta^2}$$

then

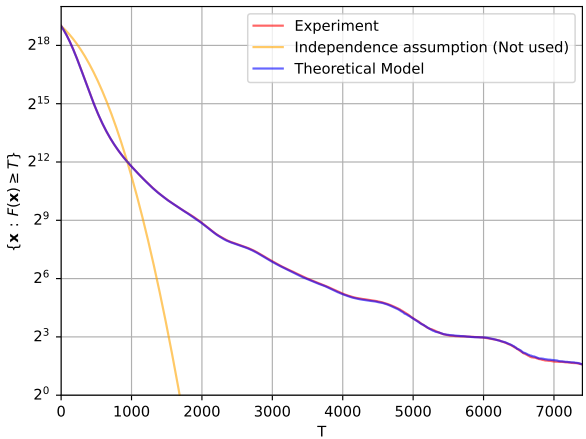
$$\mathbf{F}\left(\mathbf{x}\right) < \mathbf{F}\left(\mathbf{e}_{\mathscr{P}}\right) \quad \forall \mathbf{x} \neq \mathbf{e}_{\mathscr{P}}$$

$\rightarrow$ Can distinguish $\mathbf{e}_{\mathscr{P}}$, no false candidate.

~~**Independence Assumptions**~~

Prediction of score function

$\mathbb{P}(\mathbf{F}(\mathbf{x}) > T)$



Theorem : Dual formula

$$F(\mathbf{x}) = \sum_{i \in \mathbb{N}} N_i \left(\mathscr{C}^{\mathscr{N}} + g(\mathbf{x}) \right) K_w(i)$$

- $\mathscr{C}^{\mathscr{N}} \stackrel{\triangle}{=} \{ \mathbf{c}_{\mathscr{N}} \ : \ \mathbf{c} \in \mathscr{C} \text{ s.t } \mathbf{c}_{\mathscr{P}} = 0 \}$
- $N_i(\mathscr{D})$ number word of weight i of $\mathscr{D}$
- $K_w(i)$ is Krawtchouk polynomial
- $g(\mathbf{x})$ a known affine function

Proof: Poisson formula + $\widehat{1_w} = K_w$

Model: $N_i(\mathscr{D}) \sim$ Poisson variable of good expected value

Number of false candidates

Theorem

Under the Poisson Model the number of false candidates when

$$N > \frac{n^8}{\delta^2}$$

- RLPN : $\mathrm{poly}\,(n)$
- doubleRLPN : $2^{\alpha n}$ from some $\alpha > 0$ that we can compute

$\rightarrow$ Checking a false candidate has exponential cost.
$\rightarrow$ Complicates algorithms.
$\rightarrow$ **Overall cost of dealing with false candidates is negligible.**

Section 3

Fully provable dual attack

Goal

Theorem

There exists an algorithm that has the **same performance**, *up to polynomial factors, as (double)***RLPN** *and that we can* **fully prove**.

$$\downarrow$$

Easy to prove	$\mathbb{P}\left(\mathbf{F}\left(\mathbf{x}\right) > \mathbb{E}\left(\mathbf{F}\left(\mathbf{e}_{\mathscr{D}}\right)\right)\right) \leq \dfrac{1}{\mathrm{poly}\left(n\right)}$
Intractable	$\mathbb{P}\left(\mathbf{F}\left(\mathbf{x}\right) > \mathbb{E}\left(\mathbf{F}\left(\mathbf{e}_{\mathscr{D}}\right)\right)\right) \leq 2^{-\Theta(n)}$

Make a variant whose proof rely only on the easy bound.

Main observation

$$\mathbf{y}^{(i)} \triangleq \begin{cases} \mathbf{y}^{(i)}_{\mathscr{P}} & = \mathbf{y}_{\mathscr{P}} \\ \mathbf{y}^{(i)}_{\mathscr{N}} & = \mathbf{y}_{\mathscr{N}} + \delta_i = \mathbf{c}_{\mathscr{N}} + \underbrace{(\mathbf{e}_{\mathscr{N}} + \delta_i)}_{\text{New Error}} \end{cases}$$

Noise of LPN sample $\left\langle \mathbf{y}^{(i)}, \mathbf{h} \right\rangle = \langle \mathbf{e}_{\mathscr{P}}, \mathbf{h}_{\mathscr{P}} \rangle + \langle \mathbf{e}_{\mathscr{N}} + \delta_i, \mathbf{h}_{\mathscr{N}} \rangle$ smaller if $\mathbf{e}_{\mathscr{N}} = 1$

$$\mathbf{F}_i(\mathbf{x}) = \sum_{\mathbf{h}} (-1)^{\left\langle \mathbf{y}^{(i)}, \mathbf{h} \right\rangle - \langle \mathbf{x}, \mathbf{h}_{\mathscr{P}} \rangle}$$

$\mathbf{F}_i$ is score when we flipped i'th bit of $\mathbf{y}_{\mathscr{N}}$

Main observation

If $\mathbf{e}_{\mathscr{N}} = 1$ expect $\mathbf{F}_i(\mathbf{e}_{\mathscr{P}}) > \mathbf{F}(\mathbf{e}_{\mathscr{P}})$

Fully provable variant of RLPN

Algorithm

- Compute score $\mathbf{F}, \mathbf{F}_1, \mathbf{F}_2, \cdots, \mathbf{F}_{|\mathscr{N}|}$
- For each $\mathbf{x}$:
  - Guess that the i'th bit of $\mathbf{e}_{\mathscr{N}}$ is 1 if $\mathbf{F}_i(\mathbf{x}) > \mathbf{F}(\mathbf{x})$
  - Construct a vector $\mathbf{g}_{\mathscr{P}} = \mathbf{x}$ and $\mathbf{g}_{\mathscr{N}}$ guessed bits.
  - Test $\mathbf{g}$ solution to decoding problem: $|\mathbf{g}| = t$ and $\mathbf{y} - \mathbf{g} \in \mathscr{C}$.

Complexity : Same up to polynomial factor as original attack

Analysis

Proposition

If $N > \dfrac{\mathbf{poly}(n)}{\delta^2}$ then when $\mathbf{x} = \mathbf{e}_{\mathscr{P}}$ our guess on $\mathbf{e}_{\mathscr{N}}$ is good

Section 4

Lattices

LWE problem

LWE problem

- **Input:** $(\mathbf{G}, \mathbf{y} = \mathbf{c} + \mathbf{e}) \in \mathbb{F}_q^{k \times n} \times \mathbb{F}_q^n$ where $\mathbf{c} \in \mathscr{C}$ and $\mathbf{e} \sim \chi^n$
- **Output:** $\mathbf{e}$

Dual attacks

- Compute small (Euclidean norm) dual vectors of $\mathbf{h} \in \mathscr{C}^{\perp}$
  $\rightarrow$ By sampling short vectors in Euclidean lattice $\Lambda = \mathscr{C}^{\perp} + q\mathbb{Z}^n$

- Exploit
$$\langle \mathbf{y}, \mathbf{h} \rangle = \langle \mathbf{c} + \mathbf{e}, \mathbf{h} \rangle = \langle \mathbf{e}, \mathbf{h} \rangle$$
  is more biased toward small values of $\mathbb{F}_q$ as $\mathbf{e}$ and $\mathbf{h}$ small

Score function

$$\mathbf{F} = \sum_{\mathbf{h}} \cos\left(\frac{2\pi}{q} \langle \mathbf{y}, \mathbf{h} \rangle\right)$$

Modern dual attacks

Same splitting strategy:

$$\langle \mathbf{y}, \mathbf{h} \rangle = \langle \mathbf{e}_{\mathscr{P}}, \mathbf{h}_{\mathscr{P}} \rangle + \langle \mathbf{e}_{\mathscr{N}}, \mathbf{h}_{\mathscr{N}} \rangle$$

Solver: FFT too expensive as is in $\mathbb{F}_q$.

- Sparsification + FFT (Guo & Johansson 2021)
- Modulus switching ($\mathbb{F}_q \to \mathbb{F}_p$) + FFT (Matzov)
- **Both claim to dent security of Kyber**

Analysis:

- Ducas & Pulles 2022 showed that could not use independence assumptions to model score function. **Does dual attack work?**
- CMST24 & DP23: Model the score function
- CMST25 :
  - Dual attack: Decoding technique + FFT
  - Analyze : Generalize model and use it to analyze the attack.
  - **Indeed dent security of Kyber**

Conclusion

Thank you!