Raccoon: A Masking-Friendly Signature Proven in the Probing Model

Rafael del Pino
PQShield

Shuichi Katsumata
PQShield
AIST

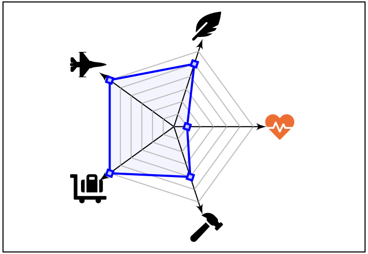
Thomas Prest
PQShield

Mélissa Rossi
ANSSI

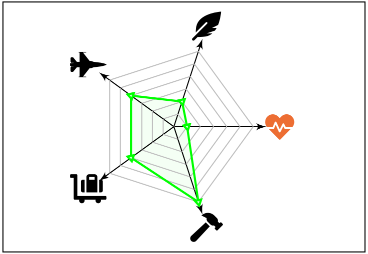
CRYPTO 2024

Motivation

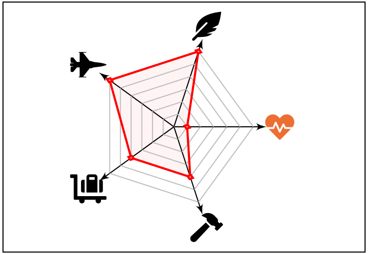
Dilithium (2017)



SPHINCS+ (2017)



Falcon (2017)



NIST PQC standards, selected in 2022, strike a balance between several criteria.

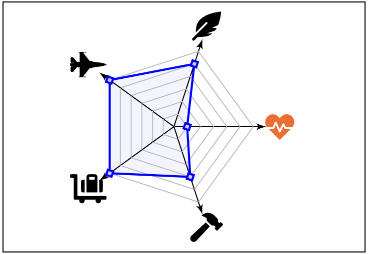
But what about :

- **Side-channel protection?**

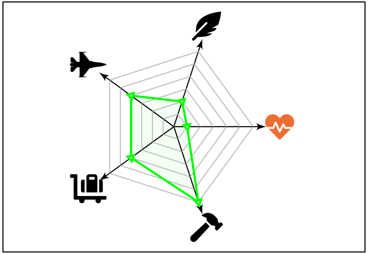
Size Speed Portability Assumptions SCA protection

Motivation

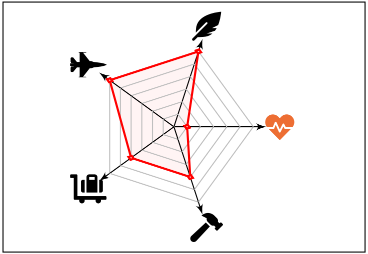
Dilithium (2017)



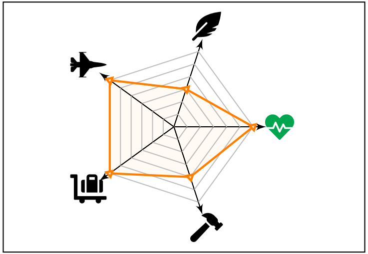
SPHINCS+ (2017)



Falcon (2017)



Raccoon (2023)

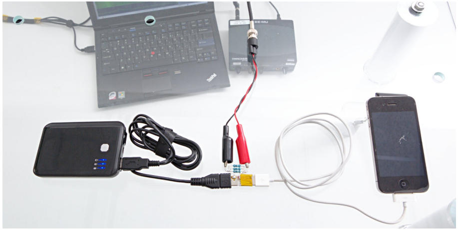


Size Speed Portability Assumptions SCA protection

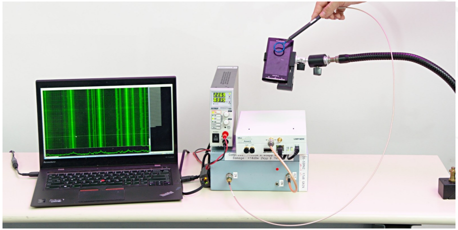
Side-Channel Attacks

Side-channel attacks in cryptography

Power consumption [KJJ99]



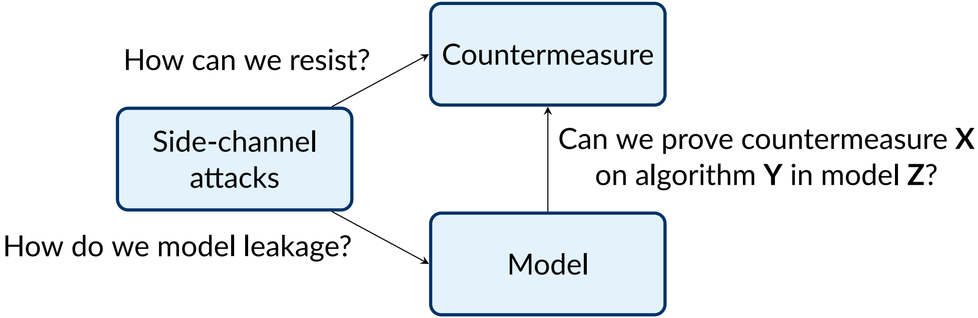
Electromagnetic emissions [Eck85]



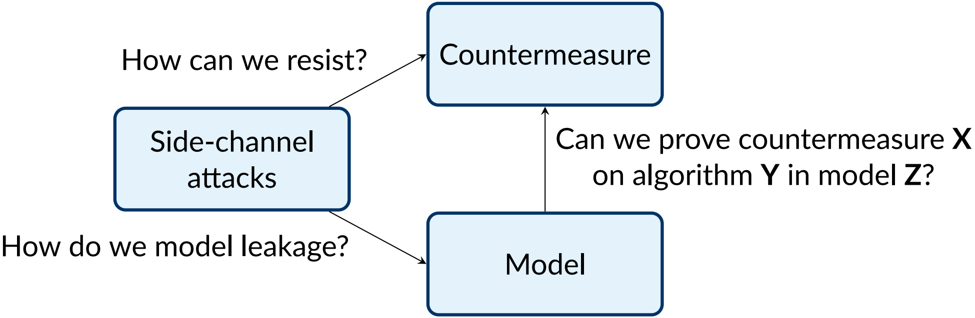
Timing measurement [Koc96]

Acoustic emissions [AA04]

Formalizing SCA



Formalizing SCA



Countermeasure: masking.

- Split sensitive value x in d shares: $\begin{cases} [\![x]\!] & = (x_0, x_1, \ldots, x_{d-1}) \\ x & = x_0 + x_1 + \cdots + x_{d-1} \end{cases}$
- Computations performed via MPC-style techniques

Model: threshold probing model. Adversary can probe any t circuit values.

- Less realistic but more convenient than other models
- Ideally, any set of t probes leaks nothing (think: masking with $d > t$ shares)

Masking Dilithium: what to expect

Dilithium-Sign

1. Sample $\mathbf{r} \leftarrow \text{Uniform}(S)$
2. $\mathbf{w} := \mathbf{A}\,\mathbf{r}$
3. $\mathbf{w}_\top := \lfloor \mathbf{w} \rceil_k$
4. $c := H(\mathbf{w}_\top, \text{msg})$
5. $\mathbf{z} := \mathbf{s}\,c + \mathbf{r}$
6. If $\mathbf{z}$ not in S', goto 1
7. $\mathbf{h} := \mathbf{w}_\top - \lfloor \mathbf{A}\,\mathbf{z} - \mathbf{t}\,c \rceil_k$
8. Output sig $= (c, \mathbf{z}, \mathbf{h})$

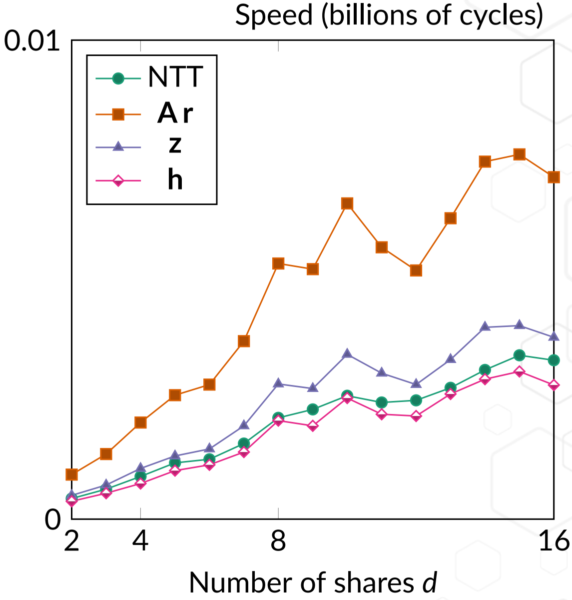
Observations:

- → All operations except 4 and 8 need to be masked
- → Three operations require mask conversions (overhead: $O(d^2 \log q)$) :
  - 1 Sampling
  - 3 Rounding
  - 6 Rejection sampling

Masked Dilithium [CGTZ23] – only the fast operations

Dilithium-Sign

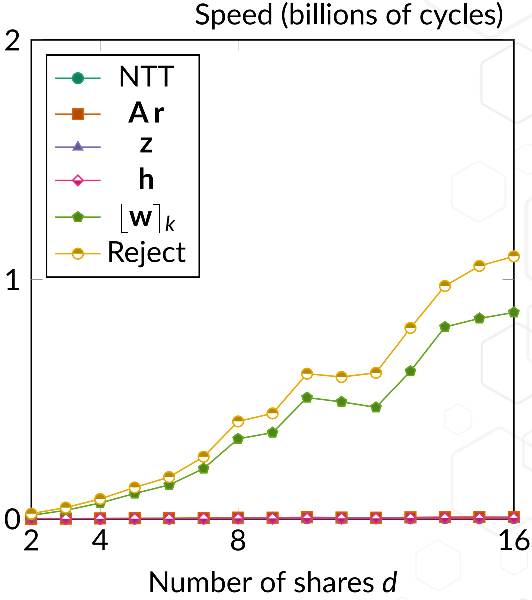
1. Sample $\mathbf{r} \leftarrow S$
2. $\mathbf{w} := \mathbf{A}\,\mathbf{r}$ $\triangleright$ $\tilde{O}(d)$
3. $\mathbf{w}_\top := \lfloor \mathbf{w} \rceil_k$
4. $c := H(\mathbf{w}_\top, \text{msg})$ $\triangleright$ No mask
5. $\mathbf{z} := \mathbf{s}\,c + \mathbf{r}$ $\triangleright$ $\tilde{O}(d)$
6. If $\mathbf{z} \notin S'$, goto 1
7. $\mathbf{h} := \mathbf{w}_\top - \lfloor \mathbf{A}\,\mathbf{z} - \mathbf{t}\,c \rceil_k$ $\triangleright$ $\tilde{O}(d)$
8. Output $\text{sig} = (c, \mathbf{z}, \mathbf{h})$



Masked Dilithium [CGTZ23] – add rounding & rejection

Dilithium-Sign

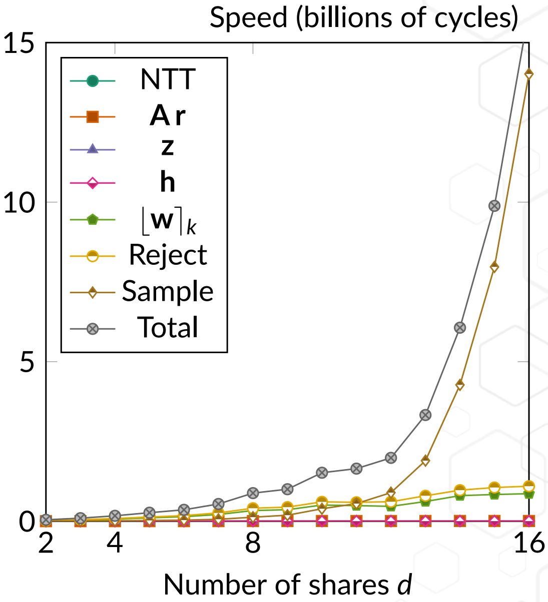
1. Sample $\mathbf{r} \leftarrow S$
2. $\mathbf{w} := \mathbf{A}\,\mathbf{r}$ $\quad \triangleright\ \tilde{O}(d)$
3. $\mathbf{w}_\top := \lfloor \mathbf{w} \rceil_k$ $\quad \triangleright\ O(d^2 \log q)$
4. $c := H(\mathbf{w}_\top, \mathrm{msg})$ $\quad \triangleright$ No mask
5. $\mathbf{z} := \mathbf{s}\,c + \mathbf{r}$ $\quad \triangleright\ \tilde{O}(d)$
6. If $\mathbf{z} \notin S'$, goto 1 $\quad \triangleright\ O(d^2 \log q)$
7. $\mathbf{h} := \mathbf{w}_\top - \lfloor \mathbf{A}\,\mathbf{z} - \mathbf{t}\,c \rceil_k$ $\quad \triangleright\ \tilde{O}(d)$
8. Output sig $= (c, \mathbf{z}, \mathbf{h})$



Masked Dilithium [CGTZ23] - add sampling

Dilithium-Sign

1. Sample $\mathbf{r} \leftarrow S$ $\triangleright$ $O(d^2 \log q)$
2. $\mathbf{w} := \mathbf{A}\,\mathbf{r}$ $\triangleright$ $\tilde{O}(d)$
3. $\mathbf{w}_\top := \lfloor \mathbf{w} \rceil_k$ $\triangleright$ $O(d^2 \log q)$
4. $c := H(\mathbf{w}_\top, \mathrm{msg})$ $\triangleright$ No mask
5. $\mathbf{z} := \mathbf{s}\,c + \mathbf{r}$ $\triangleright$ $\tilde{O}(d)$
6. If $\mathbf{z} \notin S'$, goto 1 $\triangleright$ $O(d^2 \log q)$
7. $\mathbf{h} := \mathbf{w}_\top - \lfloor \mathbf{A}\,\mathbf{z} - \mathbf{t}\,c \rceil_k$ $\triangleright$ $\tilde{O}(d)$
8. Output $\mathrm{sig} = (c, \mathbf{z}, \mathbf{h})$



Raccoon

Contribution

Contribution and main idea

- **Masking-friendly** lattice signature from scratch
- **Security proof:** instead of

$$\{\text{Masked scheme}, t \text{ probes}\} \Leftrightarrow \{\text{Unmasked scheme}\}$$

we prove:

$\{\text{Masked Raccoon}, t \text{ probes}\} \geq \{\text{Unmasked Raccoon w/ } \textbf{different parameters}\}$

Timeline:

1. **SP 2023:** Raccoon SP [dPRS23]
   - Fully heuristic
2. **NIST PQC 2023:** Raccoon NIST
   - Much improved construction
   - Still no proof
3. **EC 2024:** Plover [EEN+24]
   - Applies our ideas to Hash-&-Sign
   - Introduce the SNIu property
4. **CRYPTO 2024:** This paper
   - Formal security proof for Raccoon
   - Smooth Rényi divergence

Unmasked and masked Raccoon

Sign(sk, vk = (A, t), msg) → sig

1. Generate a short ephemeral $\mathbf{r}$
2. Compute $\mathbf{w} = \begin{bmatrix}\mathbf{A} & \mathbf{I}\end{bmatrix} \cdot \mathbf{r}$
3. Compute the challenge $c = \mathsf{H}(\mathbf{w}, \mathsf{msg}, \mathsf{vk})$
4. Compute the response $\mathbf{z} = \mathsf{sk} \cdot c + \mathbf{r}$
5. Output $\mathsf{sig} = (c, \mathbf{z})$

Starting point is "Schnorr over lattices":

- ✓ No Rejection sampling
  - › We argue that $\mathsf{sk} \cdot c + \mathbf{r} \approx \mathbf{r}$
- ✓ Rounding is not needed for security
  - › No need to mask it
- ? What about Sampling (step 1)?

Unmasked and masked Raccoon

Sign(sk, vk = (A, t), msg) → sig

1. Generate a short ephemeral $\mathbf{r}$
2. Compute $\mathbf{w} = \begin{bmatrix}\mathbf{A} & \mathbf{I}\end{bmatrix} \cdot \mathbf{r}$
3. Compute the challenge $c = \mathrm{H}(\mathbf{w}, \mathrm{msg}, \mathrm{vk})$
4. Compute the response $\mathbf{z} = \mathrm{sk} \cdot c + \mathbf{r}$
5. Output $\mathrm{sig} = (c, \mathbf{z})$

MaskSign(⟦sk⟧, vk, msg) → sig

1. $[\![\mathbf{r}]\!] = [\![\mathbf{0}]\!]$
2. For $i \in [\mathrm{rep}]$:
   1. $[\![\mathbf{r}_i]\!] = (\mathbf{r}_{i,1}, \ldots, \mathbf{r}_{i,d}) \leftarrow \chi_{\mathbf{r}}^{d}$
   2. $[\![\mathbf{r}]\!] = [\![\mathbf{r}]\!] + [\![\mathbf{r}_i]\!]$
   3. $\mathrm{Refresh}([\![\mathbf{r}]\!])$
3. $[\![\mathbf{w}]\!] = \begin{bmatrix}\mathbf{A} & \mathbf{I}\end{bmatrix} \cdot [\![\mathbf{r}]\!]$
4. $\mathrm{Refresh}([\![\mathbf{w}]\!])$
5. $\mathbf{w} = \mathrm{Decode}([\![\mathbf{w}]\!])$
6. $c = \mathrm{H}(\mathbf{w}, \mathrm{msg}, \mathrm{vk})$
7. $[\![\mathbf{z}]\!] = [\![\mathrm{sk}]\!] \cdot c + [\![\mathbf{r}]\!]$
8. $\mathrm{Refresh}([\![\mathbf{z}]\!], [\![\mathrm{sk}]\!])$
9. $\mathbf{z} = \mathrm{Decode}([\![\mathbf{z}]\!])$
10. Output $\mathrm{sig} = (c, \mathbf{z})$

Unmasked and masked Raccoon

Sign(sk, vk = (A, t), msg) → sig

1. Generate a short ephemeral $\mathbf{r}$
2. Compute $\mathbf{w} = [\mathbf{A} \quad \mathbf{I}] \cdot \mathbf{r}$
3. Compute the challenge $c = \mathrm{H}(\mathbf{w}, \mathrm{msg}, \mathrm{vk})$
4. Compute the response $\mathbf{z} = \mathrm{sk} \cdot c + \mathbf{r}$
5. Output $\mathrm{sig} = (c, \mathbf{z})$

→ AddRepNoise in lime green

- A t-probing adversary learns at most t of the $(d \cdot \mathrm{rep})$ values $\mathbf{r}_{i,j}$
- Formal analysis in [EEN+24]

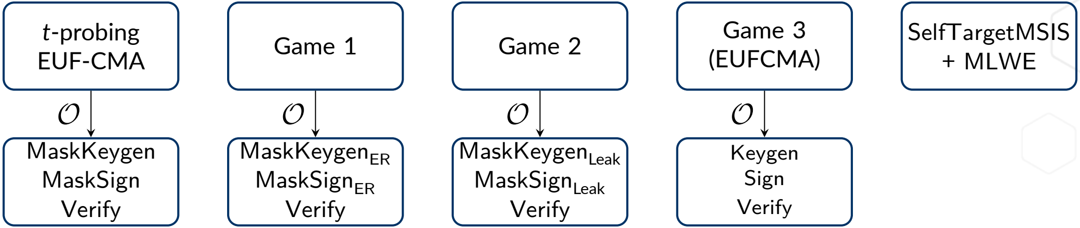
→ Refresh is useful for:

- Concrete security
- Composing gadgets (SNI)
- Moving probes around (SNI)

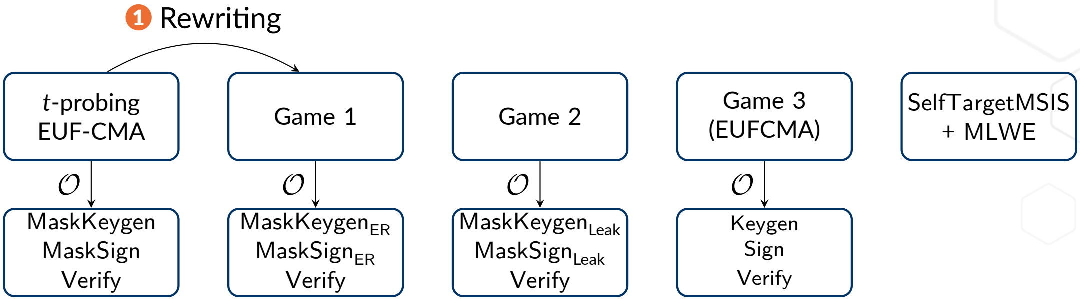
MaskSign([[sk]], vk, msg) → sig

1. $[\![\mathbf{r}]\!] = [\![\mathbf{0}]\!]$
2. For $i \in [\mathrm{rep}]$:
   1. $[\![\mathbf{r}_i]\!] = (\mathbf{r}_{i,1}, \ldots, \mathbf{r}_{i,d}) \leftarrow \chi_{\mathbf{r}}^d$
   2. $[\![\mathbf{r}]\!] = [\![\mathbf{r}]\!] + [\![\mathbf{r}_i]\!]$
   3. $\mathrm{Refresh}([\![\mathbf{r}]\!])$
3. $[\![\mathbf{w}]\!] = [\mathbf{A} \quad \mathbf{I}] \cdot [\![\mathbf{r}]\!]$
4. $\mathrm{Refresh}([\![\mathbf{w}]\!])$
5. $\mathbf{w} = \mathrm{Decode}([\![\mathbf{w}]\!])$
6. $c = \mathrm{H}(\mathbf{w}, \mathrm{msg}, \mathrm{vk})$
7. $[\![\mathbf{z}]\!] = [\![\mathrm{sk}]\!] \cdot c + [\![\mathbf{r}]\!]$
8. $\mathrm{Refresh}([\![\mathbf{z}]\!], [\![\mathrm{sk}]\!])$
9. $\mathbf{z} = \mathrm{Decode}([\![\mathbf{z}]\!])$
10. Output $\mathrm{sig} = (c, \mathbf{z})$

Proof outline (simplified)

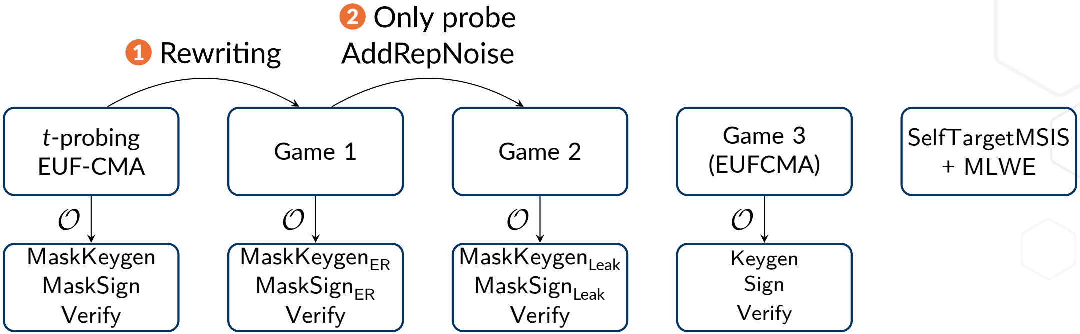


Proof outline (simplified)



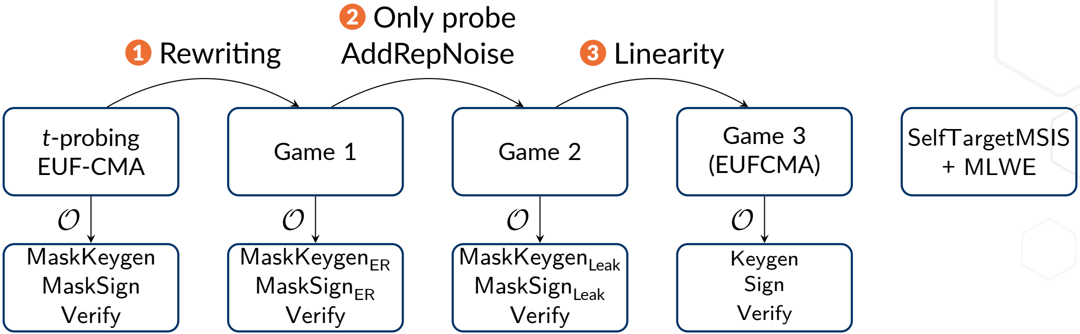
1 **Rewriting:** make randomness explicit as input

Proof outline (simplified)



1. **Rewriting:** make randomness explicit as input
2. **SNI(u) property:** move all probes to AddRepNoise randomness

Proof outline (simplified)

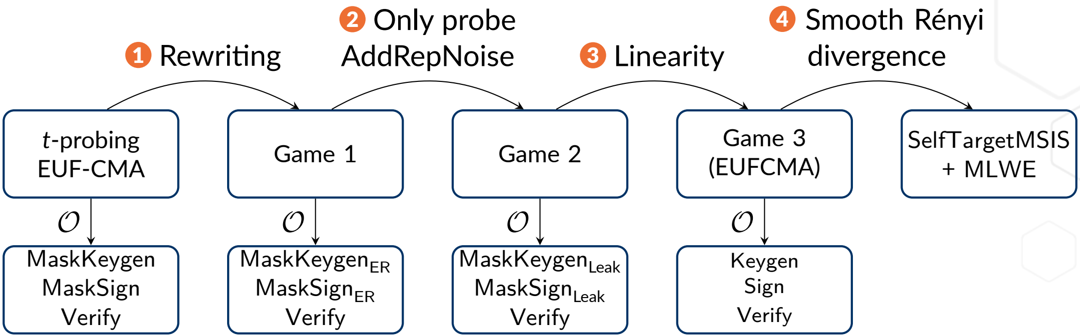


1. **Rewriting:** make randomness explicit as input
2. **SNI(u) property:** move all probes to AddRepNoise randomness
3. **Linearity:** we argue that we can can simulate Game 2 from Game 3

   Game 2: $\mathbf{w} = [\mathbf{A} \quad \mathbf{I}] \cdot \mathbf{r}$ where $\mathbf{r} = \sum_{i \in [d \cdot rep]} \mathbf{r}_i$ and we leak $(\mathbf{r}_i)_{i \in S}$ for $|S| = t$

   Game 3: $\mathbf{w} = [\mathbf{A} \quad \mathbf{I}] \cdot \mathbf{r}'$ where $\mathbf{r}' = \sum_{i \in [d \cdot rep - t]} \mathbf{r}_i$

Proof outline (simplified)



1. **Rewriting:** make randomness explicit as input
2. **SNI(u) property:** move all probes to AddRepNoise randomness
3. **Linearity:** we argue that we can can simulate Game 2 from Game 3

   Game 2: $\mathbf{w} = \begin{bmatrix}\mathbf{A} & \mathbf{I}\end{bmatrix} \cdot \mathbf{r}$ where $\mathbf{r} = \sum_{i \in [d \cdot rep]} \mathbf{r}_i$ and we leak $(\mathbf{r}_i)_{i \in S}$ for $|S| = t$

   Game 3: $\mathbf{w} = \begin{bmatrix}\mathbf{A} & \mathbf{I}\end{bmatrix} \cdot \mathbf{r}'$ where $\mathbf{r}' = \sum_{i \in [d \cdot rep - t]} \mathbf{r}_i$
4. **Final hop:** {EUFCMA of Raccoon} $\geq$ {SelfTargetMSIS + MLWE }
   - Making this formal requires introducing the *smooth Rényi divergence*

Sums of Uniforms & Smooth Rényi Divergence

Sums of Uniforms

The final reduction argues that:

$$c \cdot \mathbf{s} + \mathbf{r} \overset{s}{\approx} \mathbf{r}, \quad \text{where} \quad \mathbf{r} \leftarrow \underbrace{\chi_{\mathbf{r}} + \cdots + \chi_{\mathbf{r}}}_{T=(d \cdot \mathrm{rep}-t) \text{ times}} \tag{1}$$

How do we choose $\chi_{\mathbf{r}}$?

- **Choice 1: $\chi_{\mathbf{r}}$ is the discrete Gaussian $D_{\sigma_{\mathbf{r}}}$.**
  - + Security analysis:

$$\text{Statistical distance } SD \quad \Rightarrow \quad \sigma_{\mathbf{r}}\sqrt{T} \geq \sigma(\mathrm{sk}) \cdot \|c\|_1 \cdot 2^{\lambda} \tag{2}$$

$$\text{Rényi divergence } R_{\alpha} \quad \Rightarrow \quad \sigma_{\mathbf{r}}\sqrt{T} \geq \sigma(\mathrm{sk}) \cdot \|c\| \cdot \sqrt{\text{Queries} \cdot \dim(\mathrm{sk}) \cdot \lambda} \tag{3}$$

  - − Gaussians are difficult to sample securely against SCA

- **Choice 2: $\chi_{\mathbf{r}}$ is uniform over $\{-2^b, \ldots, 2^b - 1\}$.**
  - + Way simpler to sample securely against SCA
  - − The Rényi divergence proof strategy goes through the window

Sums of Uniforms and their caveat

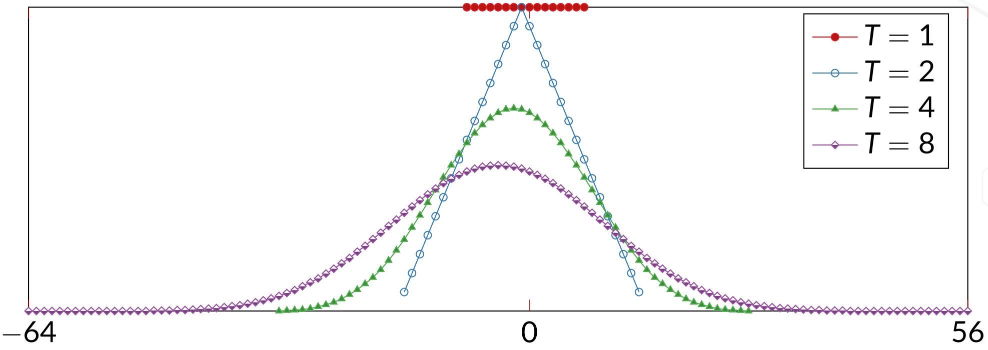


Figure 1: Sums of T uniforms in $\{-2^3, \ldots, 2^3 - 1\}$, for $T \in \{1, 2, 4, 8\}$

- + The sum of T uniforms quickly become “Gaussian-like”
- − The support is finite, so the Rényi divergence is infinite (therefore useless)

Solution: Smooth Rényi Divergence

Definition

The smooth Rényi divergence of parameters (α, ε) between P and Q is:

$$R_\alpha^\varepsilon(P;Q) = \min_{\substack{SD(P';P)\leq\varepsilon \\ SD(Q';Q)\leq\varepsilon}} R_\alpha(P';Q'),$$

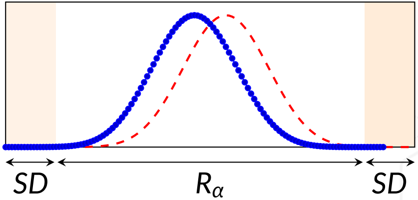
where SD is the statistical distance and R_α is the usual Rényi divergence.

Since R_α^ε is a simple composition of two f-divergences, the usual "nice" properties are immediate:

- ✓ Data processing
- ✓ Probability preservation
- ✓ Tensorization

We can leverage the complementary strengths of SD and R_α on different parts of the support:

- → The *tightness* of R_α on the heads
- → The *robustness* of SD on the tails



Wrap-up and open questions

What we have:

- → A masking-friendly lattice signature in the t-probing model
- → Simple design, but required new analytic tools (SNIu, smooth Rényi)

Open questions:

- → Security proof/arguments in more realistic models?
- → Concrete SCA resistance?

Questions?

https://raccoonfamily.org/
https://ia.cr/2024/1291

Dmitri Asonov and Rakesh Agrawal.
Keyboard acoustic emanations.
In *2004 IEEE Symposium on Security and Privacy*, pages 3–11. IEEE Computer Society Press, May 2004.

Jean-Sébastien Coron, François Gérard, Matthias Trannoy, and Rina Zeitoun.
Improved gadgets for the high-order masking of dilithium.
IACR Transactions on Cryptographic Hardware and Embedded Systems, 2023(4):110–145, Aug. 2023.

Rafaël del Pino, Thomas Prest, Mélissa Rossi, and Markku-Juhani O. Saarinen.
High-order masking of lattice signatures in quasilinear time.
In *2023 IEEE Symposium on Security and Privacy*, pages 1168–1185. IEEE Computer Society Press, May 2023.

Wim Van Eck.
Electromagnetic radiation from video display units: An eavesdropping risk?
Computers & Security, 4:269–286, 1985.

Muhammed F. Esgin, Thomas Espitau, Guilhem Niot, Thomas Prest, Amin Sakzad, and Ron Steinfeld.
sfPlover: Masking-friendly hash-and-sign lattice signatures.

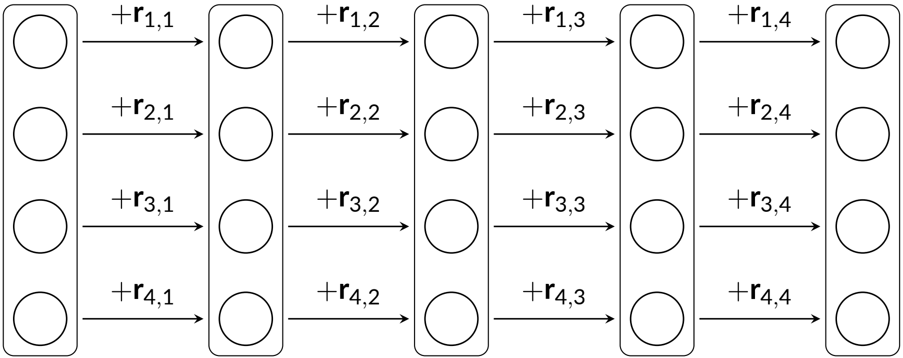
In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024*, *Part VII*, volume 14657 of *LNCS*, pages 316–345. Springer, Cham, May 2024.

Paul C. Kocher, Joshua Jaffe, and Benjamin Jun.
Differential power analysis.
In Michael J. Wiener, editor, *CRYPTO'99*, volume 1666 of *LNCS*, pages 388–397. Springer, Berlin, Heidelberg, August 1999.

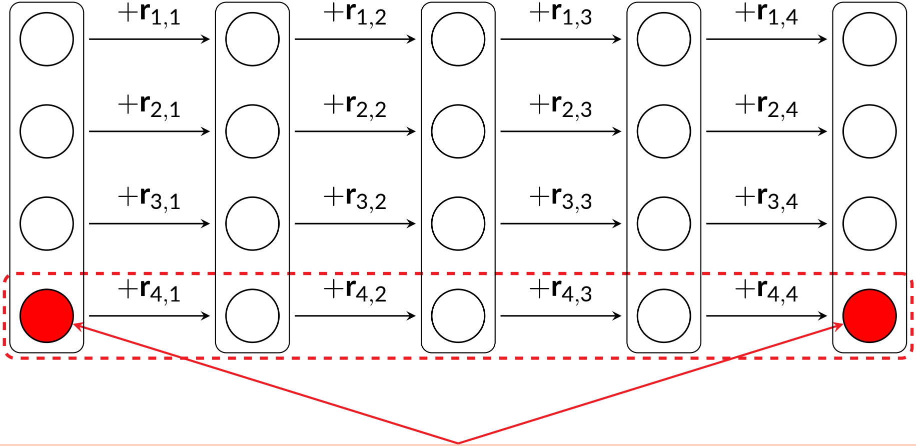
Paul C. Kocher.
Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems.
In Neal Koblitz, editor, *CRYPTO'96*, volume 1109 of *LNCS*, pages 104–113. Springer, Berlin, Heidelberg, August 1996.

Adeline Langlois, Damien Stehlé, and Ron Steinfeld.
GGHLite: More efficient multilinear maps from ideal lattices.
In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 239–256. Springer, Berlin, Heidelberg, May 2014.

What happens inside AddRepNoise?

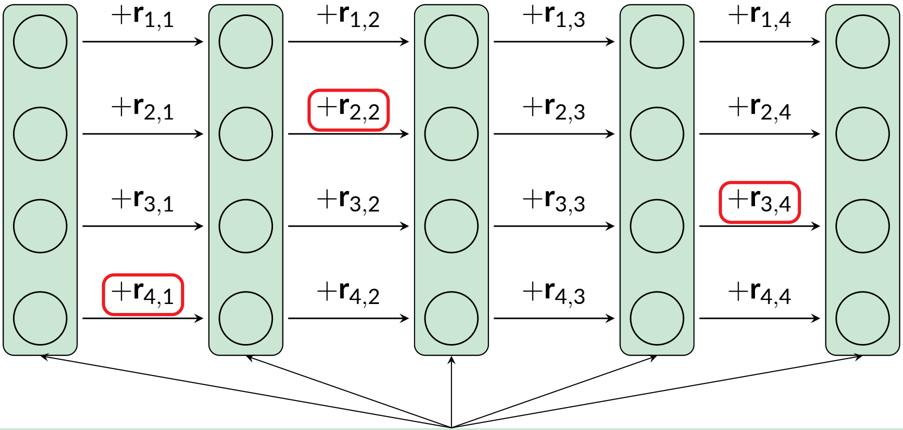


What happens inside AddRepNoise?



Problem: a probing adversary can learn the sum of T random in 2 probes.

What happens inside AddRepNoise?



Solution: add refresh gadgets to separate the algorithm in independent layers
Now a probing adversary learns at most (the sum of) t short noises.

Parameter selection and the modulus q.

Signature sizes are quadratic in $(\log q)$, so we want to minimize q (see below).

Method	Modulus q (logarithmic scale)
Smooth Rényi [Proven]	$\sigma(\text{sig})$ · MSIS (forgery) $\sigma(\text{sk})$ | $\|c\|\sqrt{\text{Queries} \cdot \text{dim}(\text{sk}) \cdot \lambda \cdot d^3}$ | $\sqrt{2}$ | $\Omega(1)$ MLWE (key rec.) · Smooth Rényi · Probing
Smooth Rényi [Conjecture]	$\sigma(\text{sig})$ · MSIS $\sigma(\text{sk})$ | $\|c\|\sqrt{\text{Queries} \cdot \text{dim}(\text{sk}) \cdot \lambda}$ | $\sqrt{2}$ | $\Omega(1)$ MLWE · Smooth Rényi (heuristic) · Probing
Hint-MLWE [Heuristic]	$\sigma(\text{sig})$ · MSIS $\sigma(\text{sk})$ | $\|c\|\sqrt{\text{Queries}}$ | $\sqrt{2}$ | $\Omega(1)$ MLWE · Hint-MLWE reduction (heur.) · Probing