Breaking Poseidon with **Graeffe**: Root-Finding for Fun (and No Profit)

Antonio Sanso (asanso 𝕏)

Researcher - Ethereum Foundation

October 31, 2025

Joint work with Z. Zhao, G. Vitto, J. Ding

https://eprint.iacr.org/2025/1916

Graeffe transform

Poseidon

Attacking Poseidon via Graeffe-Based Root-Finding over NTT-Friendly Fields*

Antonio Sanso[1] and Giuseppe Vitto[2]

[1] Ethereum Foundation name.surname@ethereum.org
[2] Zircuit name@zircuit.com

+

Breaking Poseidon Challenges with Graeffe Transforms and Complexity Analysis by FFT Lower Bounds

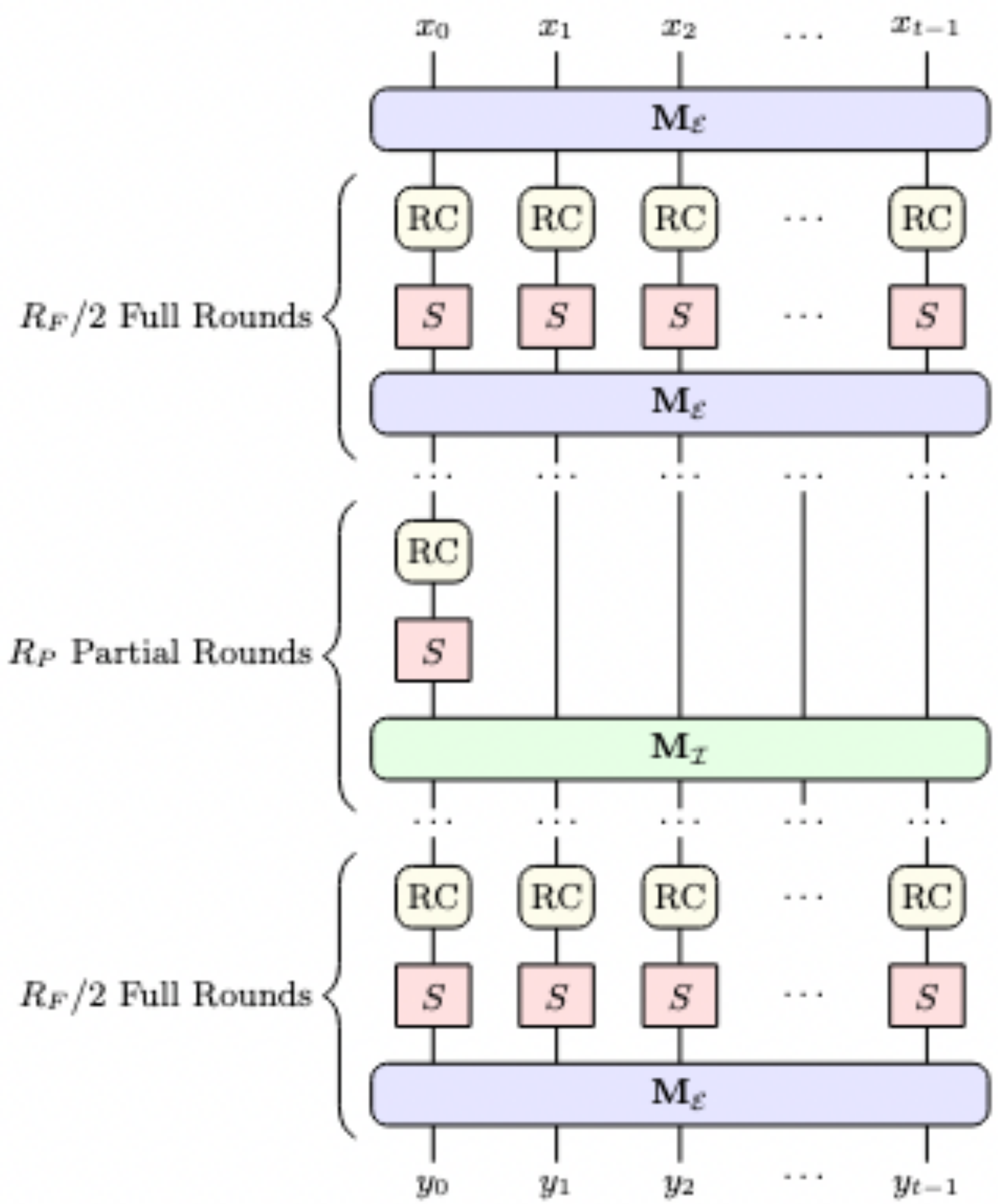
Ziyu Zhao[1] and Jintai Ding[2] (✉)

[1] Department of Mathematical Science, Tsinghua University, Beijing, China
ziyuzhao0008@outlook.com
[2] Xi'an Jiaotong-Liverpool University, Suzhou, China
Basque Center For Applied Mathematics, Bilbao, Spain
jintai.ding@gmail.com

merging of concurrent and independent works

Why Poseidon
(and other **arithmetization-oriented** primitives)?

- Hashing built for **zero-knowledge** circuits
- **Native-field friendly**: uses additions, multiplications, and a simple power S-box over prime fields



Poseidon

d : S-box degree

$R = R_{full} + R_{partial}$:

number of total rounds

Poseidon Initiative 2024-2026

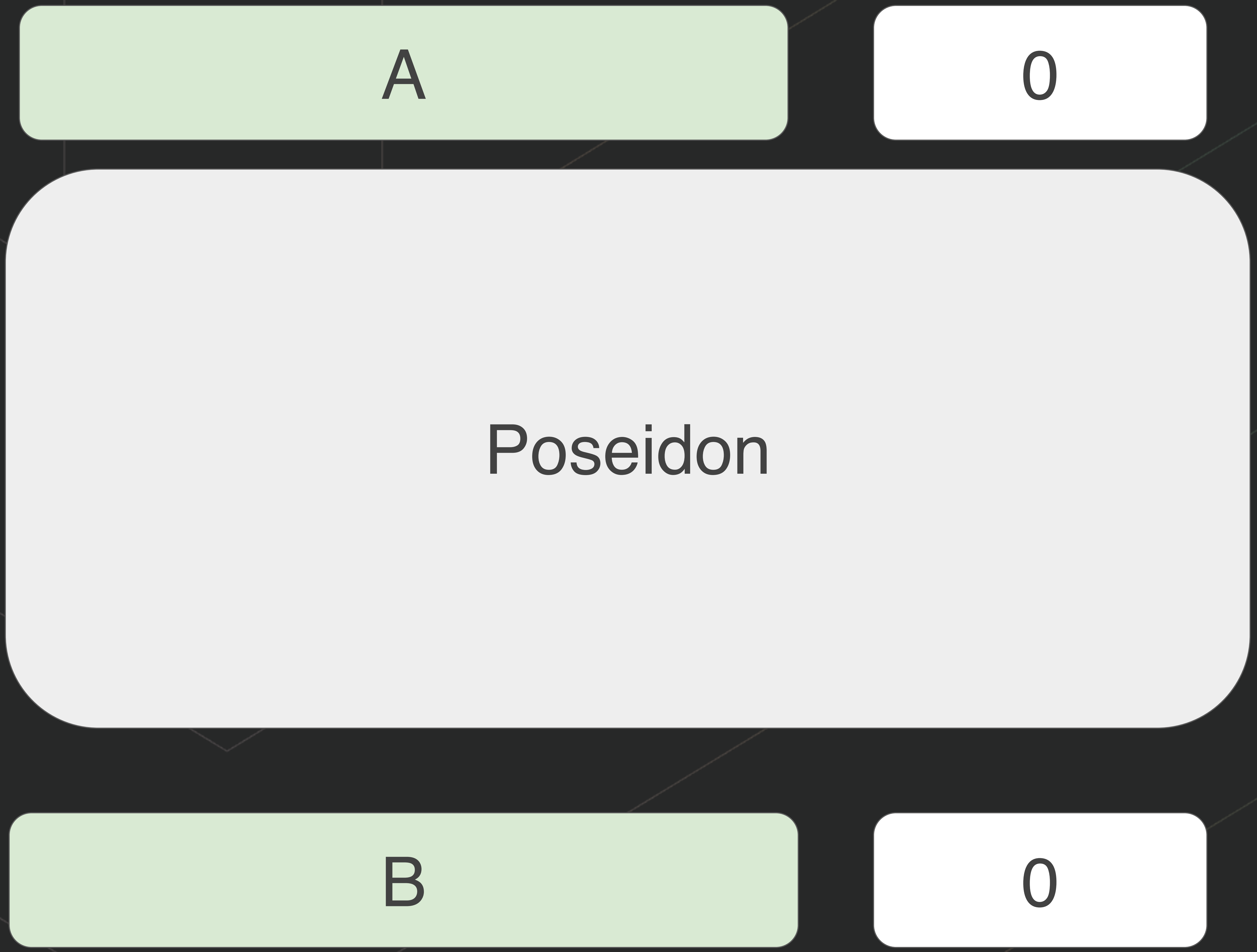
- Poseidon instances: 31-, 64-, 256-bit fields.
- Poseidon Group at EF: G. Kadianakis, D. Khovratovich, A. Sanso
- Advisory board: JP Aumasson, E. Ben-Sasson, DE Hopwood, D. Lubarov, R. Rothblum

To end by January 2027

CICO Problem

(Constrained Input, Constrained Output)

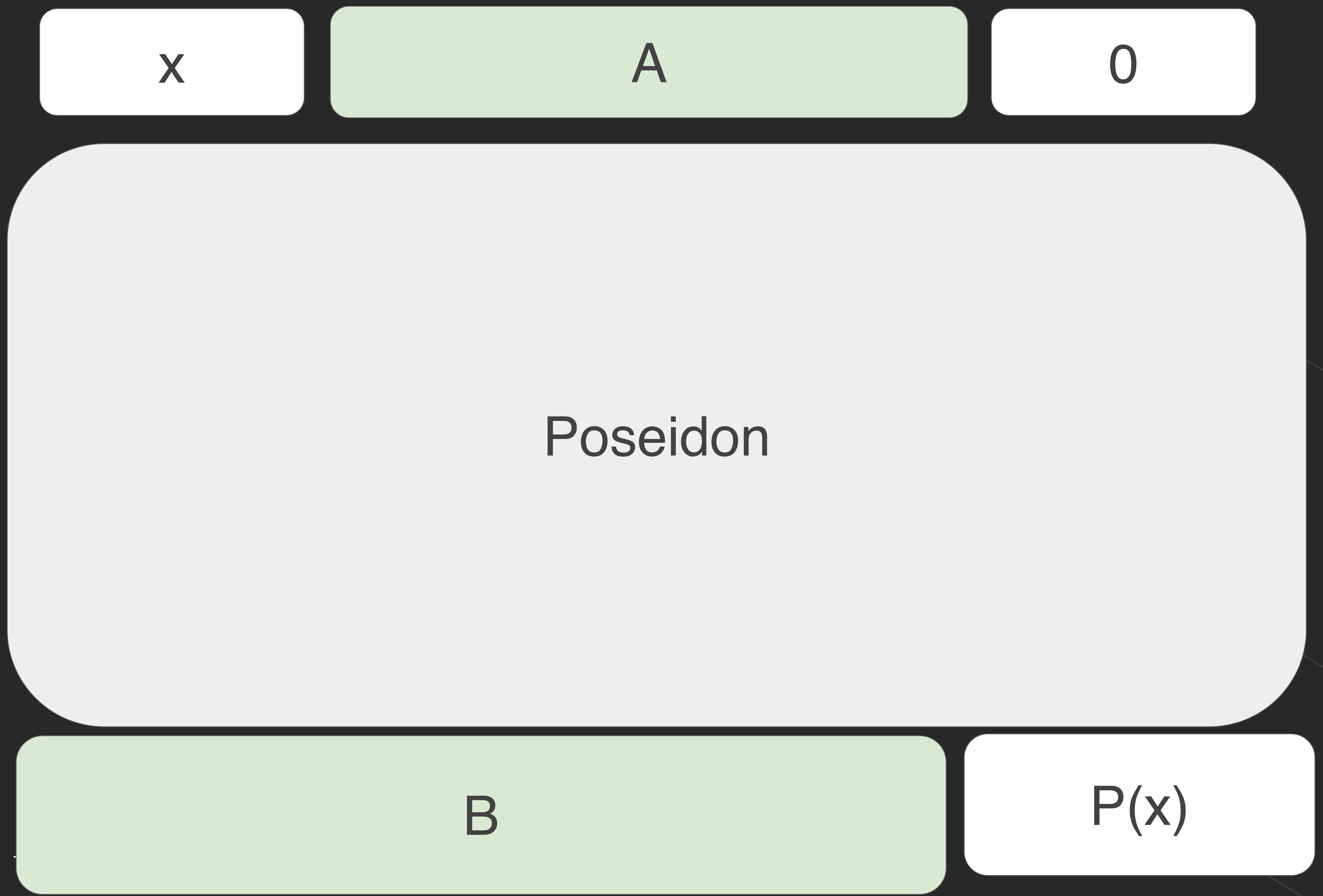
Find A, B such that



As part of the Poseidon Cryptanalysis Initiative, a bug-bounty program presents multiple CICO problem instances for participants to break.

Solving CICO problems

CICO-1 → Root Finding for <u>**Univariate**</u> Polynomials



$P(x)$: univariate polynomial of degree d^R

Solve $P(x) = 0$

Univariate system solving

2022 - *Algebraic Attacks against Some Arithmetization-Oriented Primitives* (Bariant, Bouvier, Leurent, Perrin)

Univariate system solving

Find the **roots** of a polynomial $f \in F_p[x]$ with degree $D = d^R$

(Idea behind the Rabin/Cantor-Zassenhaus algorithms)

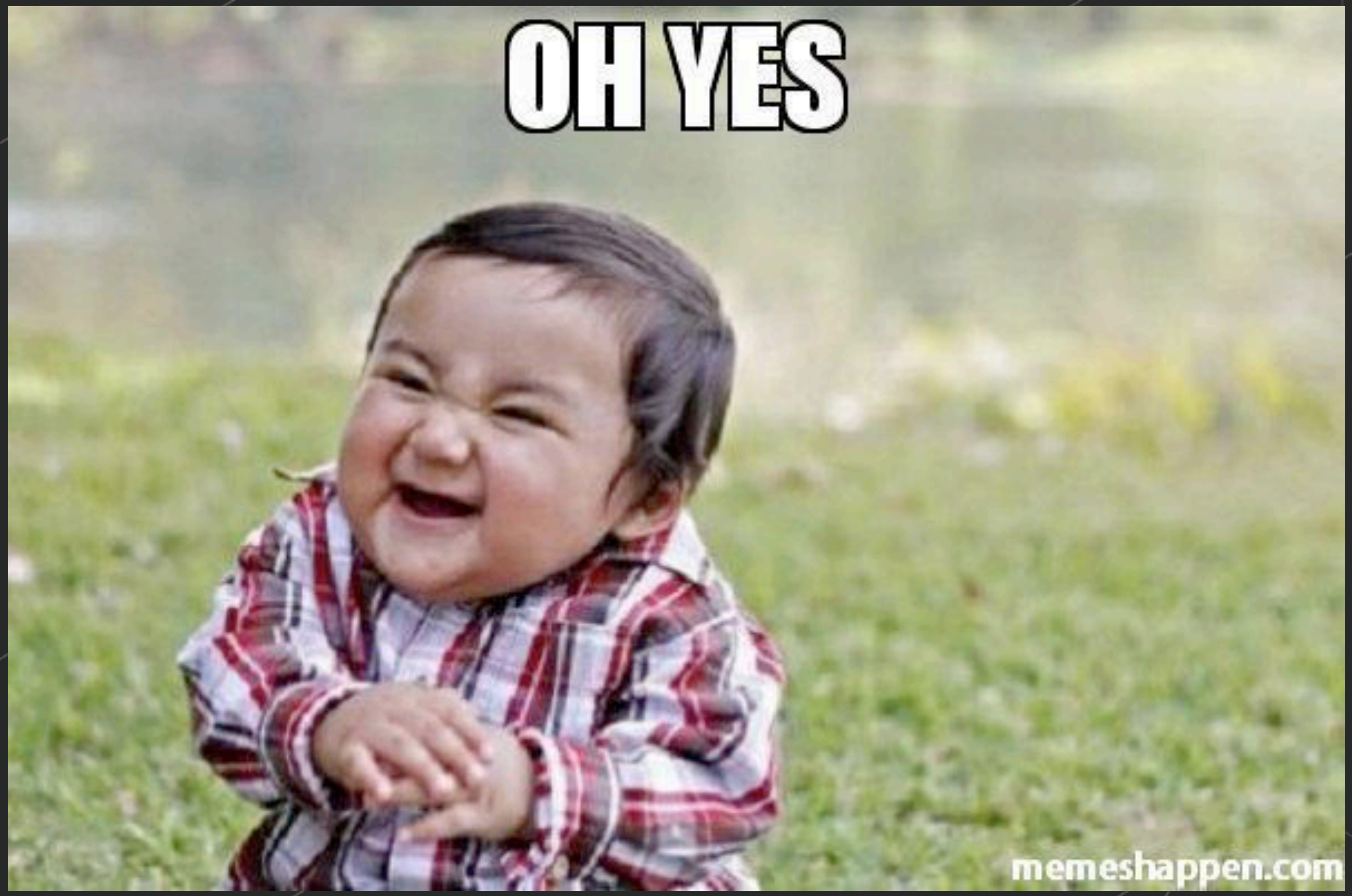
1. Compute $Q = x^p - x \pmod{P}$ — $O(M(D) log(p))$
2. Compute $R = gcd(P, Q)$ — $O(M(D))$
3. Factor — Negligible

Total cost: $O(M(D) log(p))$

$M(D)$ is the cost to multiply two polynomials of degree

$M(D) \in O(D\ log(D) log(log(D)))$ using Bluestein

Can we do any better when working with polynomials over *"special primes"* ?



Root finding over Finite FFT-fields

2015 - *Randomized root finding over finite fields using tangent Graeffe transforms* (*Grenet, van der Hoeven, Lecerf*)

works for primes $p = \sigma 2^k + 1$

Poseidon Cryptanalysis Initiative 2024-2026

Suitable bounty instances

- Poseidon-64

$$\rightarrow p - 1 = 2^{32} \cdot 3 \cdot 5 \cdot 17 \cdot 257 \cdot 65537$$

Poseidon-256

$$\rightarrow p - 1 = 2^{32} \cdot 3 \cdot 11 \cdot 19 \cdot 10177 \cdot 125527 \cdot 859267 \cdot 906349^2 \cdot 2508409 \cdot 2529403 \cdot 52437899 \cdot 254760293^2$$

The Graeffe Transform

Let $P(z) \in F_p[z]$ of degree d. The Graeffe transform of P is:

$$G(P) = P(z)P(-z)\,|_{z=\sqrt{z}} \in \mathbb{F}_p[z]$$

Lemma 1: if $P(z) = \prod_{i=1}^{d} (z - \alpha_i)$ then

$$G(P) = \prod_{i=1}^{d} (z - \alpha_i^2)$$

★

… more useful facts about Graeffe Transform

- $P_2(z) = f_0(z)^2 - z^2 f_1(z)^2$ → 2 NTTs + invNTT
- The Graeffe transform can be composed
- We can compute the Graeffe transform of arbitrary order (not just 2):

$$P_h(z) = \begin{cases} f(z), & \text{if } h = 1, \\ P_{h/2}(x)\, P_{h/2}\left(z\,\omega_\ell^{h/2}\right), & \text{if } h \text{ is even}, \\ f(z)\, P_{(h-1)/2}(z\,\omega_\ell)\, P_{(h-1)/2}\left(z\,\omega_\ell^{(h+1)/2}\right), & \text{otherwise}. \end{cases}$$

Idea

Let $p = \sigma 2^k + 1$, pick $r = 2^N$ such that $s = (p-1)/r \in [2d.4d)$:

Compute $\tilde{P} = G^{(N)}$. Then $\tilde{P} = \prod_{i=1}^{d} (z - \alpha_i^r)$

Let $\beta_i = \alpha_i^r$.

Observe $s = (p-1)/r \rightarrow p - 1 = rs \rightarrow \beta_i^s = 1$ (Fermat Little Theorem)

This means that the roots of the transform polynomial $\tilde{P}$ are the s roots of unity.

Let's compute them (brute force). Pick ω with order s in F_p and compute

$$\{\omega^i : \tilde{P}(\omega^i) = 0 \leq i \leq s\} = \{\beta_i\}$$

But we want the roots of P not the ones of $\tilde{P}$.

How do we obtain α_i from $\beta_i = \alpha_i^r$?

Main algorithm

Algorithm 3: Root Finding over the Goldilocks Field

Input: A polynomial $f(x) \in \mathbb{F}_{p64}[x]$ of degree d.
Output: A root of $f(x)$ in $\mathbb{F}_{p64}$, if one exists.

$\beta \leftarrow p_{64} - 1$; $\mu \leftarrow 1$; $g \leftarrow f$;
while β *is even* **do**
  $\beta \leftarrow \beta/2$;
  $g \leftarrow \mathsf{GT}_2(g) \bmod (x^\beta - \mu)$;
$\beta \leftarrow \beta/3$; $g_3 \leftarrow \mathsf{GT}_3(g) \bmod (x^\beta - \mu)$;
$\beta \leftarrow \beta/5$; $g_5 \leftarrow \mathsf{GT}_5(g_3) \bmod (x^\beta - \mu)$;
$\beta \leftarrow \beta/17$; $g_{17} \leftarrow \mathsf{GT}_{17}(g_5) \bmod (x^\beta - \mu)$;
$\beta \leftarrow \beta/257$; $g_{257} \leftarrow \mathsf{GT}_{257}(g_{17}) \bmod (x^\beta - \mu)$;
if g_{257} *has no roots in* $\mathbb{F}_{p64}$ **then return** $\perp$;
$\mu \leftarrow$ a common root of g_{257} and $x^{65537} - \mu$;
$\mu \leftarrow$ a common root of g_{17} and $x^{257} - \mu$;
$\mu \leftarrow$ a common root of g_5 and $x^{17} - \mu$;
$\mu \leftarrow$ a common root of g_3 and $x^5 - \mu$;
$\mu \leftarrow$ a common root of g and $x^3 - \mu$;
$\beta \leftarrow 2^{32}$; $h \leftarrow f \bmod (x^\beta - \mu)$;
return a common root of h and $x^{2^{32}} - \mu$;

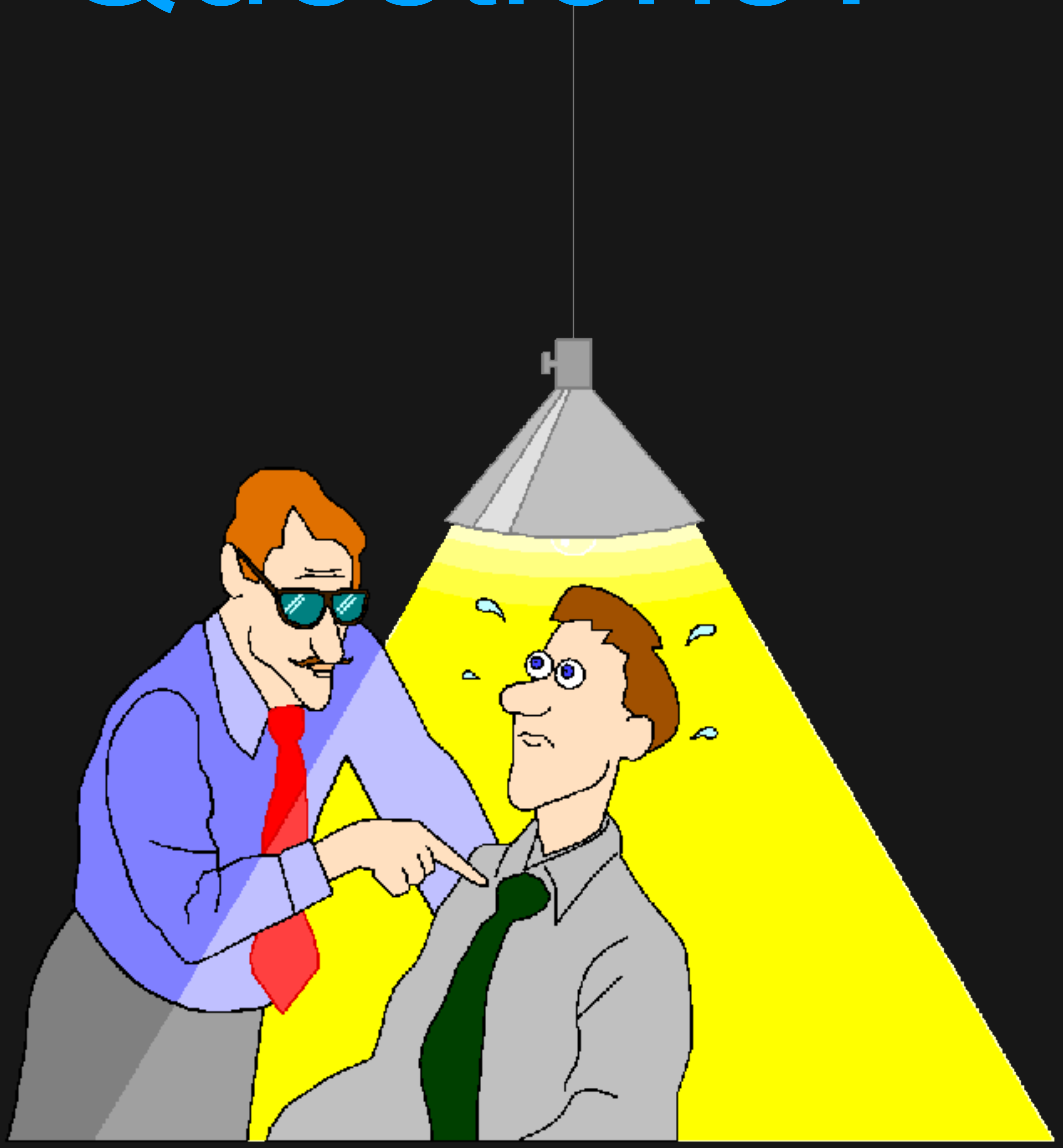
Poseidon Cryptanalysis Initiative 2024-2026

Ethereum Foundation

- Poseidon-64:
- ~~24-bit estimated security: RF=6, RP=7~~ **$4000 claimed 23 Apr 2025**
- ~~28-bit estimated security: RF=6, RP=8.~~ **$6000 claimed 27 Apr 2025**
- ~~32-bit estimated security: RF=6, RP=10.~~ **$10000 claimed 24 May 2025**
- 40-bit estimated security: RF=6, RP=13. $15000

Instance	Field	κ	R_P	R_F	Degree	Time	Memory	Time [7]	Memory [7]
P2_6_7		24	6	7	7^{12}	$2^{8.56}$s	0.32TB	$2^{21.81}$s	6.1TB
P2_6_8	Goldilocks	28	6	8	7^{13}	$2^{11.38}$s	1.8TB	$2^{24.83}$s	41TB
P2_6_10		32	6	10	7^{15}	$2^{18.35}$s†	90TB	$2^{30.88}$s	1.9PB

Questions?



Seeking:

STARK Research Intern

(MSc/PhD)