Isogeny Unpredictability Assumptions, and Applying Generic Proof Systems to Isogenies

Shai Levin, University of Auckland

Supervisor: Steven Galbraith

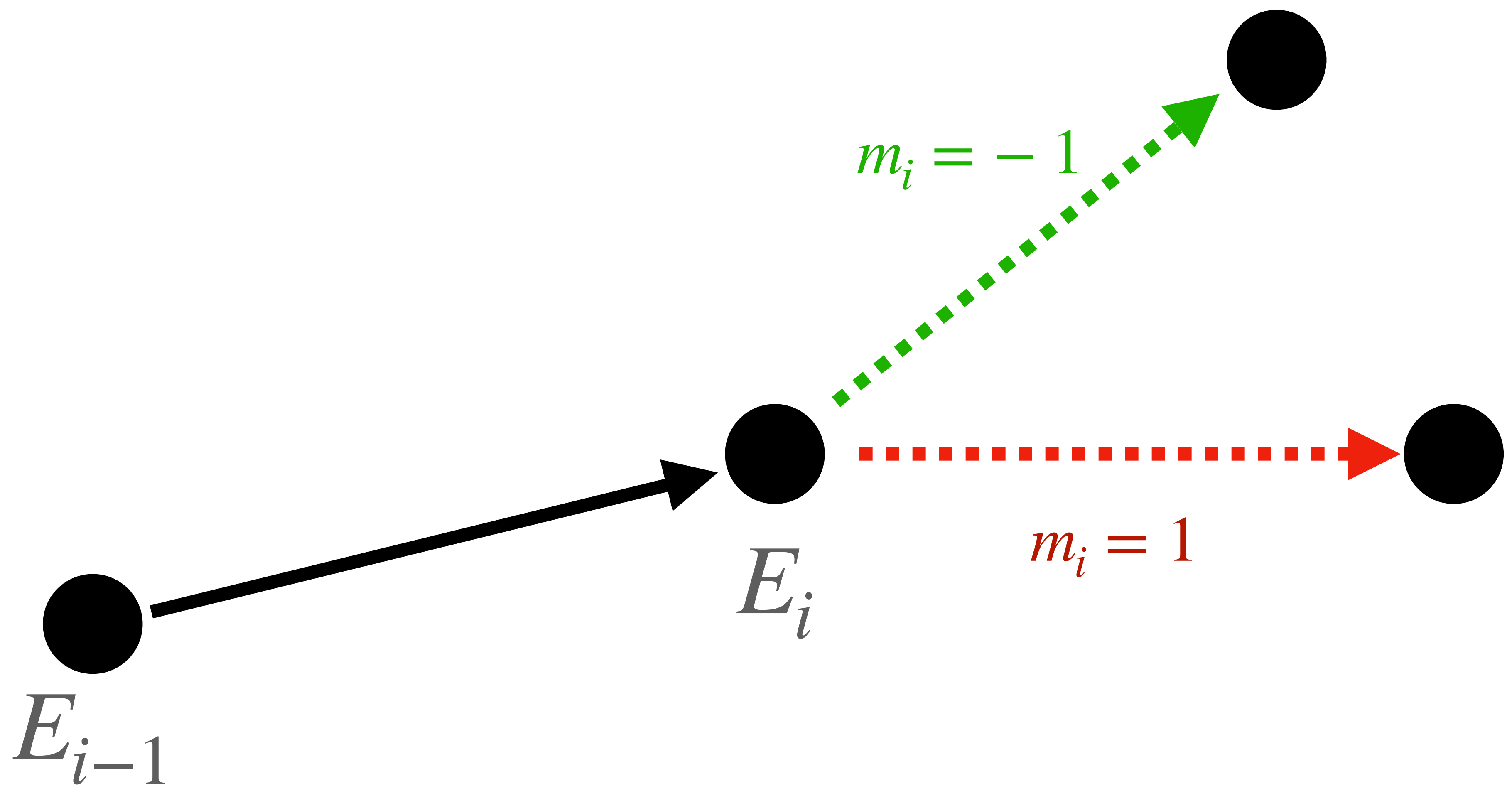
Prepared with content from a collaboration with Robi Pedersen (eprint:2024/1626) and my PhD thesis.

UNIVERSITY OF AUCKLAND
Waipapa Taumata Rau
NEW ZEALAND

Part I: A novel unpredictability assumption from isogenies

CGL Hash Function

Over the Full 2-Supersingular Isogeny Graph



At Step i: go “left” $m_i = -1$ and “right” if $m_i = 1$

For Part 1, have this in your mind:

NOT Generalising: genus or ℓ (yet)

Generalising in the following sense:

- Representation for E_i
- Step function
- Direction (“left”/“right”)

Assume E_0 has unknown endomorphism ring, and we walk n steps

Radical CGL Hash Function (I)

Radical Isogenies [1]:

Given E and $P \in E[2]$, produces a point $P' \in E'[2]$ such that the composition

$$E \to E' \to E/\langle P' \rangle$$

is a cyclic (non-backtracking) 4-isogeny.

- Using the formulas, over $p \equiv 3 \mod 4$:

  $\mathbb{F}_{p^2} \cong \mathbb{F}_p[j]$ for $j^2 = -1$

- For $x = a + bj \in \mathbb{F}_p[j]$, we denote

  $\mathrm{Re}(x) = a,$ and $\mathrm{Im}(x) = b$

Representation for E_i:

$$E_i : y^2 = x^3 + A_i x^2 + C_i x$$

Step Function:

$$E_{i+1} : y^2 = x^3 + A_{i+1} x^2 + C_{i+1} x$$

$$A_{i+1} = 6m_i\alpha_i + A_i, \quad C_{i+1} = 4m_i\alpha_i A_i + 8C_i$$

Direction:

$$\alpha_i = \sqrt{C_i}$$

If $\mathrm{Re}(\alpha_i) \neq 0$, s.t. $\mathrm{Re}(\alpha_i)$ is a QR in $\mathbb{F}_p$

*(If $\mathrm{Re}(\alpha_i) = 0$, s.t. $\mathrm{Im}(\alpha_i)$ is a QR in $\mathbb{F}_p$)

[1]: Castryck, W., Decru, T., Vercauteren, F. (2020). Radical Isogenies. In: Moriai, S., Wang, H. (eds) Advances in Cryptology – ASIACRYPT 2020. ASIACRYPT 2020. Lecture Notes in Computer Science(), vol 12492. Springer, Cham. https://doi.org/10.1007/978-3-030-64834-3_17

Radical CGL Hash Function (II)

For the remainder of this talk:

- CGL(m) refers to evaluating **this** CGL(E_0, m)
- CGL$(m \,||\, k)$ refers to **this** CGL(CGL$(E_0, m), k)$

Algorithm 1 $\mathsf{CGL}(E_0, m)$: Novel variant of CGL using radical isogeny formulas

Require: Coordinates $(A_0, C_0) \in \mathbb{F}_{p^2}$ defining a supersingular elliptic curve $E_0 : y^2 = x^3 + A_0x^2 + C_0x$, message $m = m_1m_2 \dots m_n \in \{-1, 1\}^n$

Ensure: Coordinates $(A_n, C_n) \in \mathbb{F}_{p^2}$ defining a supersingular elliptic curve $E_n : y^2 = x^3 + A_nx^2 + C_nx$

for $i = 0$ to $n - 1$ **do**
  $\alpha_i \leftarrow \sqrt{C_i}$ ▷ Start with arbitrary root
  if $\mathsf{Re}(\alpha_i) \neq 0$ and $\mathsf{Re}(\alpha_i)$ is not a square **then**
    $\alpha_i \leftarrow -\alpha_i$
  else if $\mathsf{Re}(\alpha_i) = 0$ and $\mathsf{Im}(\alpha_i)$ is not a square **then**
    $\alpha_i \leftarrow -\alpha_i$
  end if
  $A_{i+1} \leftarrow 6m_i\alpha_i + A_i$
  $C_{i+1} \leftarrow 4m_i\alpha_iA_i + 8C_i$
end for
return (A_n, C_n)

Unpredictable Functions

- Let $f : \mathscr{E} \times \mathscr{K} \to \mathscr{E}$ be a deterministic function
- We are considering evaluations of the composition:

$$g(m, k) = f(f(E_0, m), k)$$

(for $m, k \in \mathscr{K}$ & fixed public parameter E_0)

- f is *unpredictable* if a PPT adversary $\mathscr{A}$ wins the unpredictability game with negligible probability.

1. $k \leftarrow \mathscr{K}$
2. $E_k \leftarrow f(E_0, k)$
3. $(m^*, E^*) \leftarrow \mathscr{A}^{g(\cdot,k)}(E_k)$

Unpredictability Game

$\mathscr{A}$ wins if

$$E^* = g(m^*, k) = f(f(E_0, m^*), k)$$

and $\mathscr{A}$ did not query m^*

Unpredictable Functions

Unpredictability implies:

- $f(E_0, \cdot\,)$ is preimage resistant
- $f(E_0, \cdot\,)$ is collision resistant
- f is non-commutative in the following sense:

$$f(f(E_0, m), k) \neq f(f(E_0, k), m)$$

Are isogenies a good fit?

1. $k \leftarrow \mathscr{K}$
2. $E_k \leftarrow f(E_0, k)$
3. $(m^*, E^*) \leftarrow \mathscr{A}^{g(\cdot,k)}(E_k)$

Unpredictability Game

$\mathscr{A}$ wins if

$$E^* = g(m^*, k) = f(f(E_0, m^*), k)$$

and $\mathscr{A}$ did not query m^*

High Level - Isogeny Unpredictability Assumption

Conjecture:

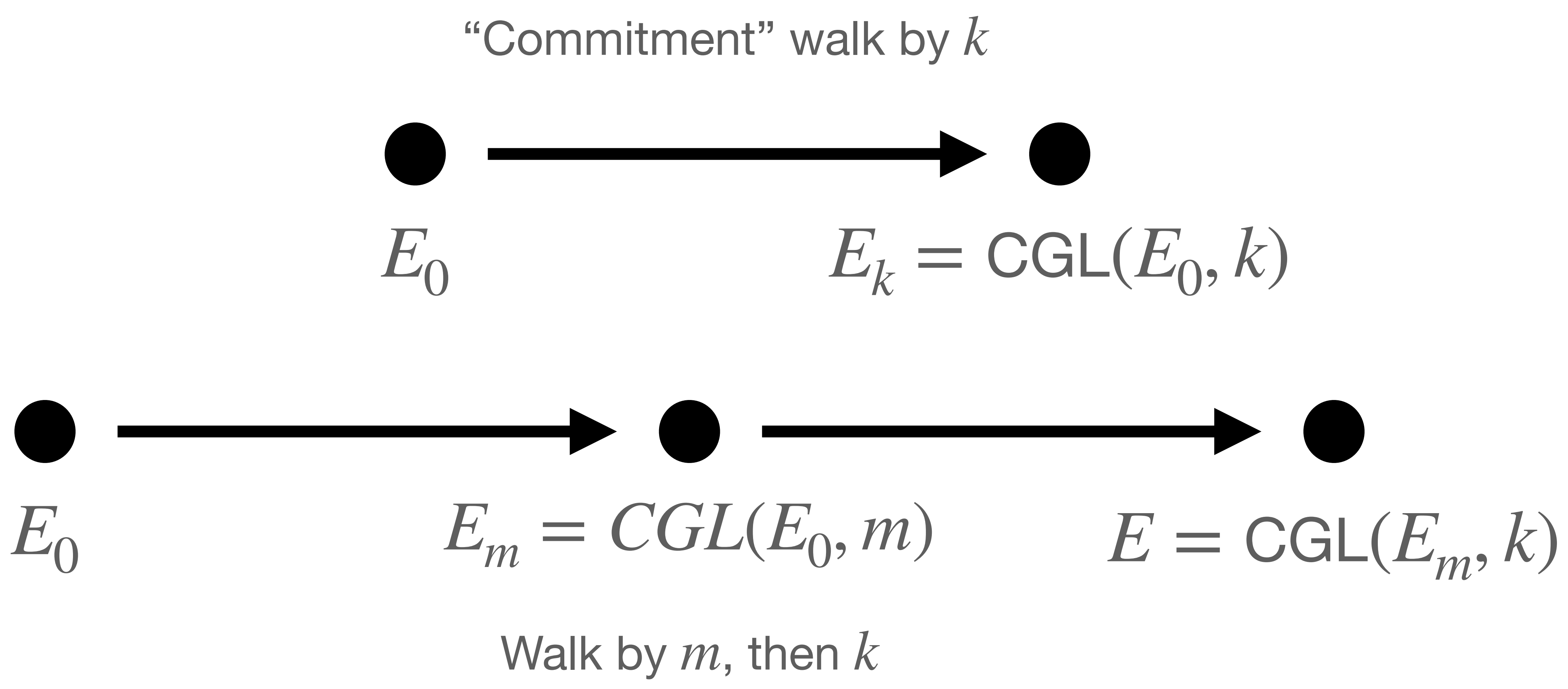
"Good" instantiations of CGL are unpredictable

Let:

- $\mathscr{E}$ be the set of supersingular curves
- $\mathscr{K} = \{-1,1\}^n$ (or $\{0,1\}^n$)
- E_0 be a curve of unknown endomorphism ring*
- $f(E_0, m) = \text{CGL}(E_0, m)$

On Security:

- We require $n \gtrsim 2\lambda$ and $\log p \gtrsim 2\lambda$ for preimage resistance
- Challenging to analyse this hardness assumption:
  - "Directions" are important (i.e. defined by the residuosity of square roots)
  - Appears "algebraically unrelated" to the isogeny structure
  - Expander mixing lemmas do not apply since walks are not random.



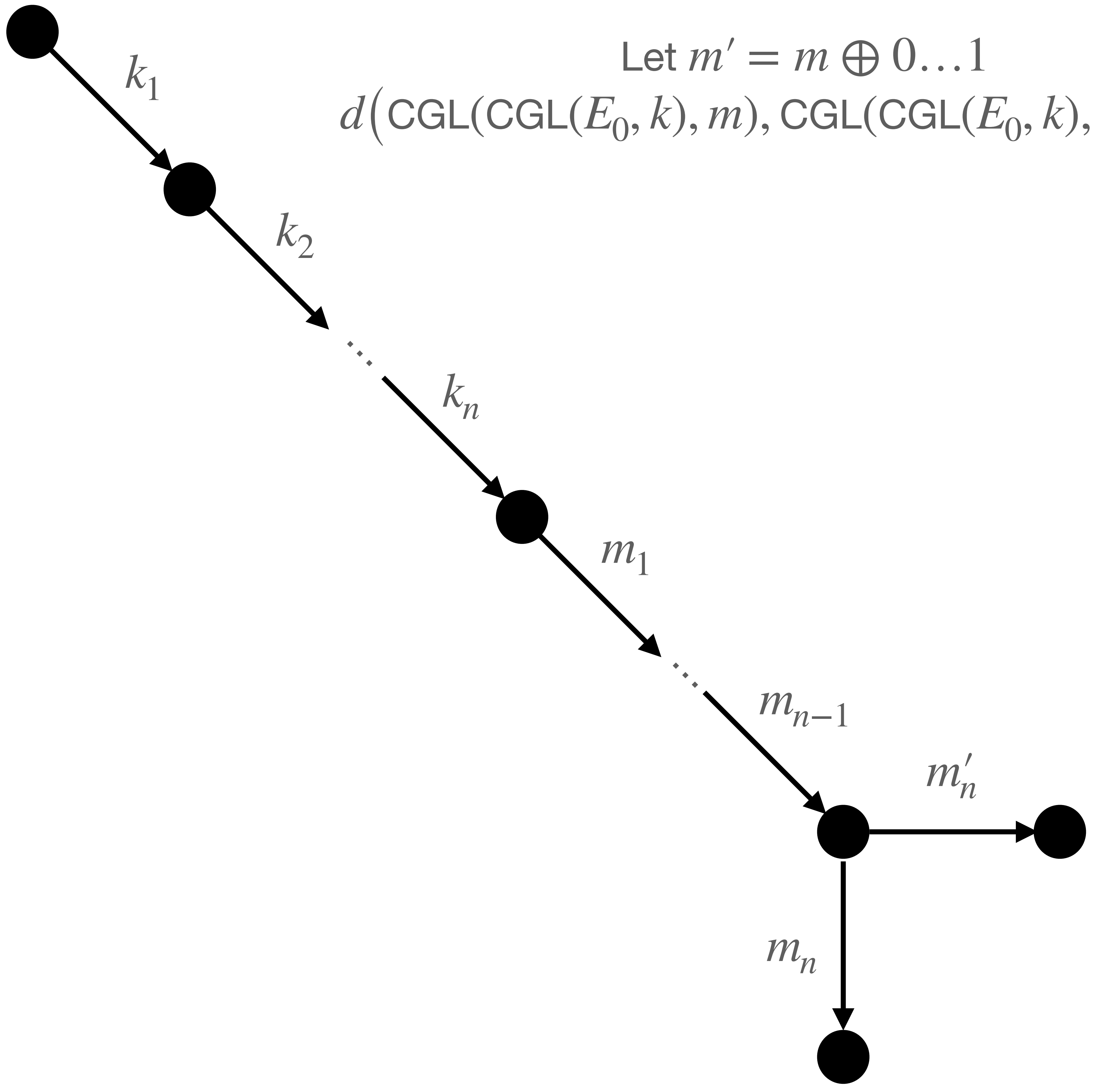
Intuition for motivating security:

Distinct, correlated messages m, m' should have uncorrelated outputs

$\text{CGL}(\text{CGL}(E_0, m), k)$, *and* $\text{CGL}(\text{CGL}(E_0, m'), k)$

What doesn't work:

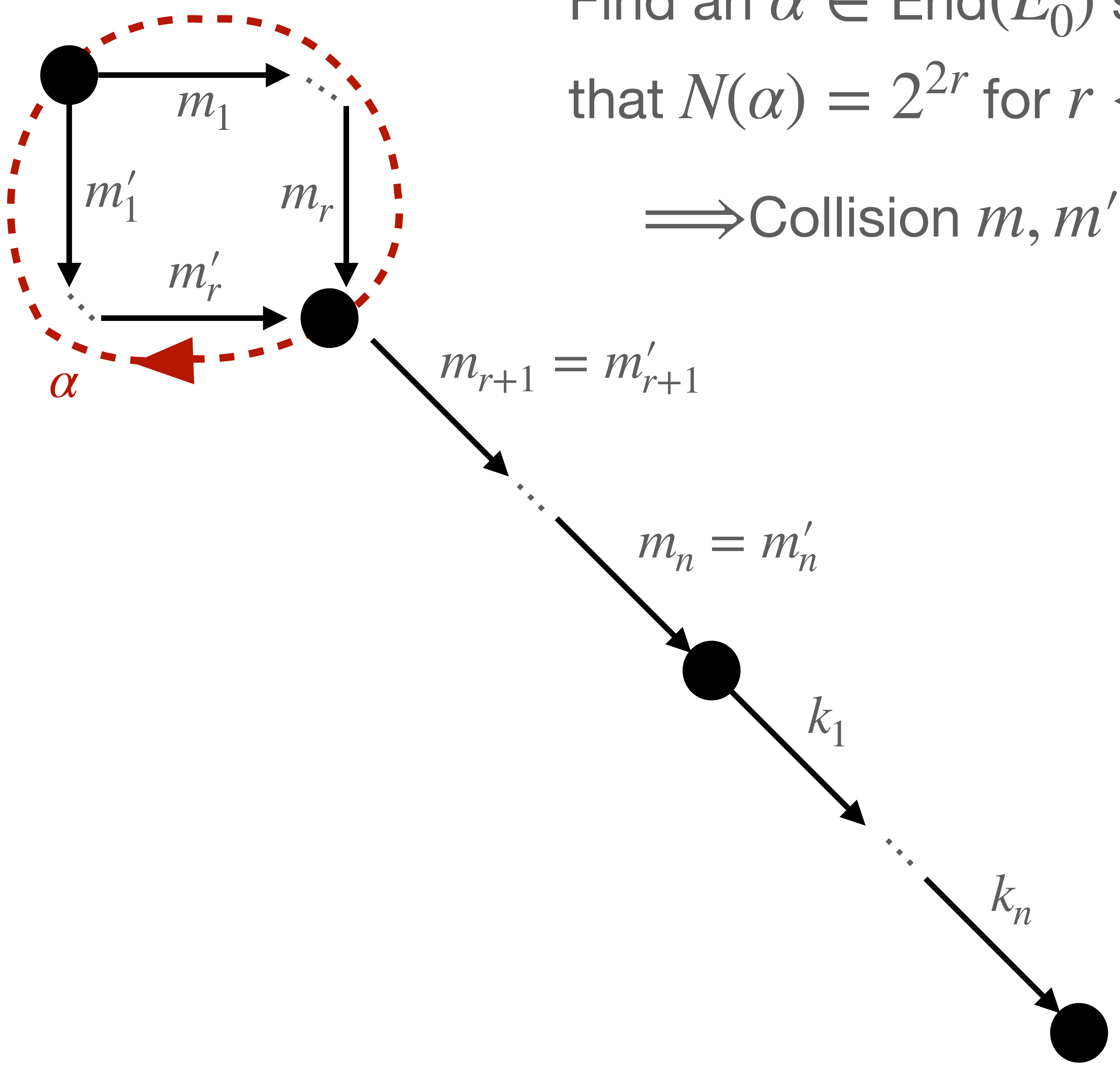
Example 1: Walk with the key first



Let $m' = m \oplus 0\ldots1$

$d\big(\text{CGL}(\text{CGL}(E_0, k), m), \text{CGL}(\text{CGL}(E_0, k), m')\big) = 2$

Example 2: Endomorphism Ring Known



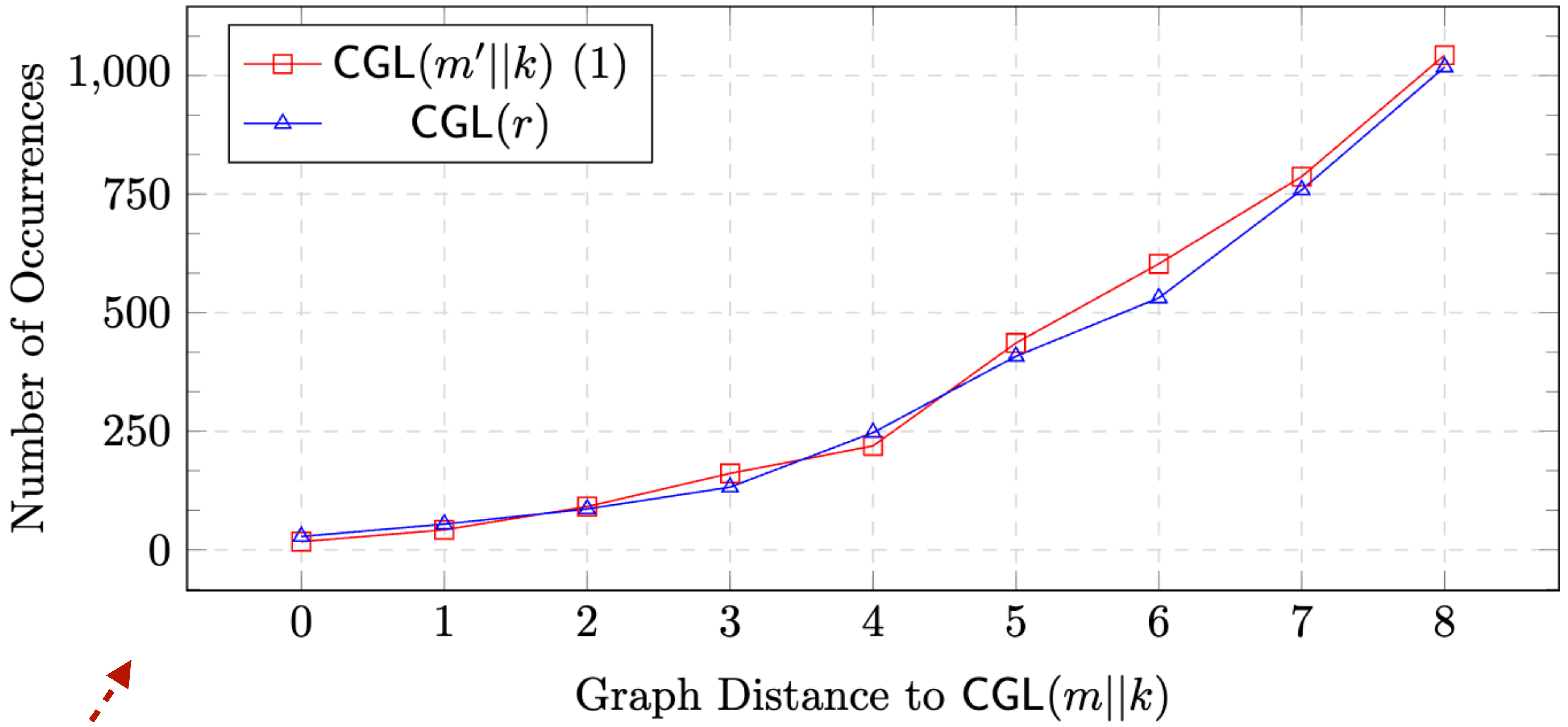
Find an $\alpha \in \text{End}(E_0)$ such that $N(\alpha) = 2^{2r}$ for $r < n$

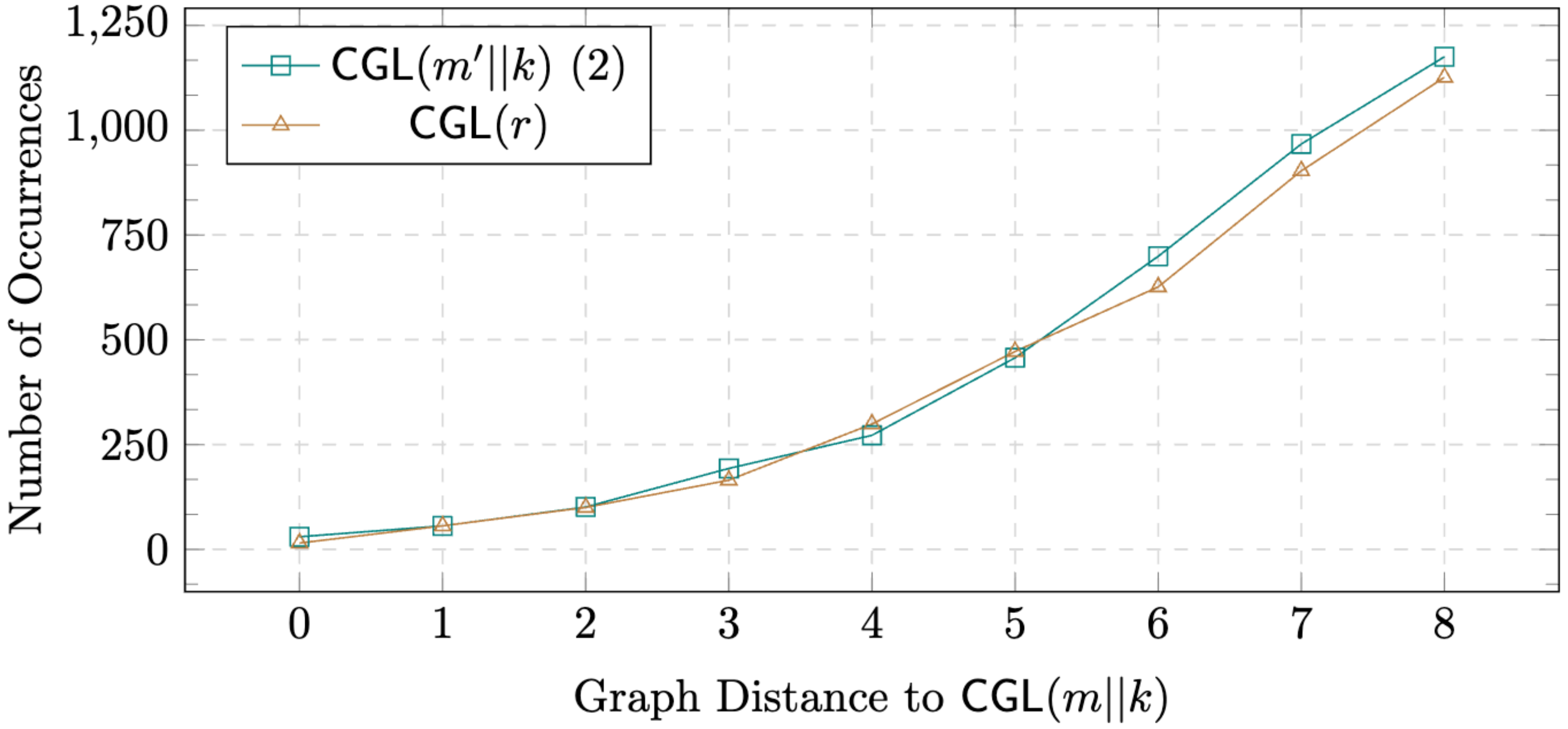
$\Longrightarrow$ Collision m, m'

Experiment Design

$n = 13$
$p = 8191$

- For small (feasible) parameters
- Sample a uniform key $k \leftarrow \{-1,1\}^n$
- Sample 1000 messages $m \leftarrow \{-1,1\}^n$
- Graph distance between evaluations of CGL$(m||k)$ and:
  - CGL$(m'||k)$ for "correlated" $m' \in \{-1,1\}^n$
  - CGL(r) for same number of $r \leftarrow \{-1,1\}^{2n}$
- Choice of "correlation":
  1. All m' such that $d(m,m') = 1$
  2. All m' which differ at the 4 least significant bits.
- Bounded Dijkstra's search (<9) to compute distances and increment occurrences of close evaluations.





Verifiable Random Functions

$\Pi_{\text{VRF}} = (\text{SetUp}, \text{KeyGen}, \text{Eval}, \text{Verify})$

- SetUp $\rightarrow$ pp
- Keygen $\rightarrow$ (sk, pk)
- $\text{Eval}_{sk}(m) \rightarrow (h, \pi)$

  π is a proof that $f_{\text{sk}}(m) = h$

- $\text{Verify}_{\text{pk}}(m, h, \pi) \rightarrow 0/1$

  accept/reject proof of evaluation

A pseudorandom function with verifiability

Motivation: Given a leader and some state, choose the next leader (or state) fairly.

Applications:

- **Blockchain Proof of Stake**
- **Randomness Beacons**
- **DNSSec**
- **Transparent Online Casinos, etc.**

Algorand

- Efficient, Robust Post Quantum VRFs are largely still an open problem
- Two other new isogeny based protocols:
  - Capybara and Tsubaki (Lai, CiC ’24) - slow proofs - GA-DDH
  - (Leroux, Eurocrypt ’25) - based on SQIsign variants - OMIP

Verifiable Random Functions - Security Properties

$\Pi_{\mathsf{VRF}} = (\mathsf{SetUp}, \mathsf{KeyGen}, \mathsf{Eval}, \mathsf{Verify})$

- SetUp $\rightarrow$ pp
- Keygen $\rightarrow$ (sk, pk)
- $\mathsf{Eval}_{sk}(m) \rightarrow (h, \pi)$

  π is a proof that $f_{\mathsf{sk}}(m) = h$
- $\mathsf{Verify}_{\mathsf{pk}}(m, h, \pi) \rightarrow 0/1$

  accept/reject proof of evaluation

Provability: *for all messages,*

$$\Pr\left[\mathsf{Verify}_{\mathsf{pk}}(m, h, \pi) = 1 \;\middle|\; \begin{array}{c} \mathsf{pp} \leftarrow \mathsf{SetUp}(1^\lambda) \\ (\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}(\mathsf{pp}) \\ (h, \pi) \leftarrow \mathsf{Eval}_{\mathsf{sk}}(m) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda) \,.$$

(Weak/Full) Unique Provability: *for (PPT/Unbounded) $\mathscr{A}$*

$$\Pr\left[\begin{array}{c} \mathsf{Verify}_{\mathsf{pk}}(m, h_1, \pi_1) = 1 \,\wedge \\ \mathsf{Verify}_{\mathsf{pk}}(m, h_2, \pi_2) = 1 \;\wedge\; h_1 \neq h_2 \end{array} \;\middle|\; \begin{array}{c} \mathsf{pp} \leftarrow \mathsf{SetUp}(1^\lambda) \\ (\mathsf{pk}, m, h_1, \pi_1, h_2, \pi_2) \leftarrow \mathcal{A}(1^\lambda, \mathsf{pp}) \end{array}\right]$$

is negligible.

Verifiable Random Functions - Security Properties

$\Pi_{\mathsf{VRF}} = (\mathsf{SetUp}, \mathsf{KeyGen}, \mathsf{Eval}, \mathsf{Verify})$

- SetUp $\rightarrow$ pp
- Keygen $\rightarrow$ (sk, pk)
- $\mathsf{Eval}_{sk}(m) \rightarrow (h, \pi)$

  π is a proof that $f_{\mathsf{sk}}(m) = h$
- $\mathsf{Verify}_{\mathsf{pk}}(m, h, \pi) \rightarrow 0/1$

  accept/reject proof of evaluation

Residual Pseudorandomness:

An adversary cannot distinguish the output of a message of their choice, even given access to an evaluation oracle.

For a PPT adversary $\mathscr{A}$, the following game is hard:

1. $\mathsf{pp} \leftarrow \mathsf{SetUp}(1^\lambda)$
2. $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}(\mathsf{pp})$
3. $(m^*) \leftarrow \mathcal{A}^{\mathsf{Eval}_{\mathsf{sk}}(\cdot)}(\mathsf{pk})$
4. $(h_0, \pi) \leftarrow \mathsf{Eval}_{\mathsf{sk}}(m^*)$
5. $h_1 \leftarrow \{0,1\}^{2\lambda}$
6. $b \leftarrow \{0,1\}$
7. $b' \leftarrow \mathcal{A}^{\mathsf{Eval}_{\mathsf{sk}}(\cdot)}(h_b)$

Residual PR Game

$\mathscr{A}$ wins if $b = b'$ and m^* was not queried

Unpredictable Functions + Proof of Evaluation $\Longrightarrow$ VRF

In the ROM, let:

- f be an unpredictable function
- NIZK be a non-interactive proof for the relation:

$$\mathscr{R} = \left\{ (E_1, E_2), (F_1, F_2), k \ : \ E_2 = f(E_1, k) \wedge F_2 = f(F_1, k) \right\}$$

(Pedersen, L., eprint:2024/1626)

Concurrent work (Giunta, Stewart, Eurocrypt '24):

- Prove security of 'equivalent' construction
- Additionally show it satisfies *unbiasability*
- Fairness in PoS application was not guaranteed from existing VRF properties

$\mathsf{SetUp}(1^\lambda)$

1 : $(f, \mathcal{E}, \mathcal{K}, E_0) \leftarrow \mathsf{SetUpUF}(1^\lambda)$.
2 : **return** $\mathsf{pp} := (f, \mathcal{E}, \mathcal{K}, E_0)$

$\mathsf{KeyGen}(\mathsf{pp})$

1 : Parse pp as $(f, \mathcal{E}, \mathcal{K}, E_0)$.
2 : $k \leftarrow\!\!\$\ \mathcal{K}$
3 : $E_k := f(E_0, k)$
4 : **return** $(\mathsf{sk}, \mathsf{pk}) := (k, E_k)$

$\mathsf{Eval}_k(m; \mathsf{pp})$

1 : Parse pp as $(f, \mathcal{E}, \mathcal{K}, E_0)$.
2 : **assert** $m \in \mathcal{K}$
3 : $E_m := f(E_0, m)$
4 : $E := f(E_m, k)$
5 : $\pi_1 \leftarrow \mathsf{NIZK}.P(k, (E_0, E_k, E_m, E); \mathsf{pp})$
6 : **return** $h := H(E_k, m, E), \pi := (\pi_1, E)$

$\mathsf{Verify}_{\mathsf{pk}}(h, (\pi_1, E), m; \mathsf{pp})$

1 : Parse pp as $(f, \mathcal{E}, \mathcal{K}, E_0)$.
2 : $E_m := f(E_0, m)$
3 : $b_1 \leftarrow h \stackrel{?}{=} H(E_k, m, E)$
4 : $b_2 \leftarrow \mathsf{NIZK}.V(\pi_1, (E_0, E_k, E_m, E); \mathsf{pp})$
5 : **return** $b_1 \wedge b_2$

Part 2: Tutorial On Applying Generic Proofs to Isogenies

Context - Isogeny Proofs

Proofs of knowledge of a cyclic isogeny:

$$\mathscr{R}_{\ell^k\text{–cyclic}} = \{((E_0, E_1), \phi) \ : \ \phi : E_0 \to E_1 \text{ is a cyclic isogeny}, \ \deg \phi = \ell^k\}$$

Prior approaches [1-4] via Σ-protocols:

- Small challenge spaces $\implies$ many repetitions
- Need rational coprime torsion $E[N] \implies$ field extensions or larger prime
- Requires an additional assumption - DSSP
- Extractor recovers a $N^2\ell^k$-isogeny

Can we do better?

$$\begin{tikzcd} E \arrow[r, "\phi"] \arrow[d, "\psi"'] & F \arrow[d, "\psi'"] \\ E' \arrow[r, "\phi'"'] & F' \end{tikzcd}$$

Is it really an isogeny talk
without an SIDH square?

[1] Jao, D., De Feo, L. (2011) *Towards Quantum-Resistant Cryptosystems from Supersingular Elliptic Curve Isogenies* https://doi.org/10.1007/978-3-642-25405-5_2
[2] Galbraith, S.D., Petit, C. & Silva, J. (2017) *Identification Protocols and Signature Schemes Based on Supersingular Isogeny Problems* https://doi.org/10.1007/s00145-019-09316-0
[3] De Feo, L., Dobson, S., Galbraith, S.D., Zobernig, L. (2022) *SIDH Proof of Knowledge* https://doi.org/10.1007/978-3-031-22966-4_11
[4] Basso, A. *et al*. (2023). Supersingular Curves You Can Trust. In: Hazay, C., Stam, M. (eds) Advances in Cryptology https://doi.org/10.1007/978-3-031-30617-4_14

Introducing Generic Proof Systems

In the last decade, we have efficient generic proof systems (zk-SNARKs):

- Succinct: $|\pi| = \text{poly}(\log(|w|))$
- Post-quantum (information theoretic security in the ROM)
- Prove relations (R1CS, AIR, Polynomial systems) expressing arbitrary computations over a finite field.

- Prior approaches (i.e. [1]) required FFT-friendly fields:
  - $2^r \mid \mathbb{F}^\times$ for $r \approx \lceil \max(\log n, \log m) \rceil$
- New approaches (i.e. [2]) using expander codes are *field agnostic:*
  - Used in the context of ECDSA signature verification.

[1] Ben-Sasson, E., Chiesa, A., Riabzev, M., Spooner, N., Virza, M., Ward, N.P. (2019). *Aurora: Transparent Succinct Arguments for R1CS* EUROCRYPT 2019 https://doi.org/10.1007/978-3-030-17653-2_4
[2] Block, A.R., Fang, Z., Katz, J., Thaler, J., Waldner, H., Zhang, Y. (2024). *Field-Agnostic SNARKs from Expand-Accumulate Codes*. CRYPTO 2024. https://doi.org/10.1007/978-3-031-68403-6_9

Rank 1 Constraint Systems (R1CS)

R1CS is parameterised by: $\mathbb{F}_q$, Number of constraints m, Number of variables n

$$\mathscr{R}_{\text{R1CS}} = \{(A, B, C, \mathbf{v}, q), (\mathbf{w}) \mid A\mathbf{z} \circ B\mathbf{z} = C\mathbf{z}, \mathbf{z} = (1 \ \ \mathbf{v} \ \ \mathbf{w})\}$$

Where:

- $A, B, C \in \mathbb{F}_q^{m \times (n+1)}$
- $z := (1 \ \ \mathbf{v} \ \ \mathbf{w}) \in \mathbb{F}_q^{n+1}$

and

$$\begin{pmatrix} a_1 \\ \cdots \\ a_n \end{pmatrix} \circ \begin{pmatrix} b_1 \\ \cdots \\ b_n \end{pmatrix} = \begin{pmatrix} a_1 b_1 \\ \cdots \\ a_n b_n \end{pmatrix}$$

Hadamard (coordinate-wise) product

Conceptually:

- Rows of A, B, C encode quadratic equations

$(\textit{linear expression}) \times (\textit{linear expression}) = (\textit{linear expression})$

- $\mathbf{v}$ - public input variables
- $\mathbf{w}$ - secret input, and intermediate variables

Constructing R1CS for $\mathscr{R}_{2^k\text{-cyclic}}$ via Modular Polynomials

(Cong, Lai, L. ACNS 2023). Represent an ℓ^k-isogeny from $E_0 \to E_1$ by $j_1, j_2, \ldots, j_{k-1}$ such that:

$$\Phi_\ell(j(E_0), j_1) = 0$$
$$\Phi_\ell(j_i, j_{i+1}) = 0 \quad \text{for all } i \in \{1,\ldots,k-2\}$$
$$\Phi_\ell(j_{k-1}, j(E_1)) = 0$$

We can represent the equation $\Phi_2(X, Y) = 0$ by the equation:

$$-XY(c_4X + c_4Y - XY) = c_0 + c_1(X+Y) + c_2(X^2+Y^2) + X^3 + Y^3 + c_3XY$$

If $k = 1$, we obtain the R1CS instance →

$$z = \begin{pmatrix} 1 & j_0 & j_1 & j_0^2 & j_1^2 & j_0^3 & j_1^3 & j_0 j_1 \end{pmatrix}^T$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & c_4 & c_4 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ c_0 & c_1 & c_1 & c_2 & c_2 & 1 & 1 & c_3 \end{bmatrix}$$

For paths of length $k \Longrightarrow (n,\ m) = (4k+3,\ 4k-2)$

Constructing R1CS for $\mathscr{R}_{2^k\text{–cyclic}}$ via Radical Isogeny Formulae

(Pedersen, L., eprint:2024/1626) Represent an 2^k-isogeny from $E_0 \to E_k$ by a sequence of A_i, C_i for $i = 0, \ldots, k$:

$$E_i : y^2 = x^3 + A_i x^2 + C_i x$$

$$A_{i+1} = 6\sqrt{C_i} + A_i\,, \quad C_{i+1} = 4\sqrt{C_i}A_i + 8C_i\,.$$

Representing via the quadratic equations:

$$6C_{i+1} - 48C_i = 4A_i(A_{i+1} - A_i), \qquad 36 \cdot C_i = (A_{i+1} - A_i)^2$$

If $k = 1$, we obtain the R1CS instance

$$z = \begin{pmatrix} 1 & A_0 & C_0 & A_1 & C_1 \end{pmatrix}^T$$

$$A = \begin{bmatrix} 0 & 4 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 0 & -48 & 0 & 6 \\ 0 & 0 & 36 & 0 & 0 \end{bmatrix}$$

For paths of length $k \implies (n, m) = (2k + 2, 2k)$

Proof of Radical CGL Evaluation

$$\mathscr{R}_{\mathsf{CGL}} = \{(E_0, E_n), m \mid E_n = \mathsf{CGL}(E_0, m)\}$$

By rearranging and substituting, the relation is satisfied if,

for all $i = 0, \ldots n-1,\ \exists \beta_i \in \mathbb{F}_p$

$$\beta_i^2 = \mathrm{Re}(\alpha_i)$$
$$\alpha_i^2 = C_i$$
$$A_{i+1} - A_i = 6m_i\alpha_i$$
$$2A_i(A_{i+1} - A_i) = 3C_{i+1} - 24C_i$$
$$0 = (m_i + 1)(m_i - 1)$$

How to realise Re($\cdot$) **as a polynomial?**

- $\mathrm{Re}(x) = 2^{-1}(x + x^p)$ - too costly
- Encode the arithmetic in $\mathbb{F}_p$!

Algorithm 1 $\mathsf{CGL}(E_0, m)$: Novel variant of CGL using radical isogeny formulas

Require: Coordinates $(A_0, C_0) \in \mathbb{F}_{p^2}$ defining a supersingular elliptic curve $E_0 : y^2 = x^3 + A_0x^2 + C_0x$, message $m = m_1m_2 \ldots m_n \in \{-1, 1\}^n$

Ensure: Coordinates $(A_n, C_n) \in \mathbb{F}_{p^2}$ defining a supersingular elliptic curve $E_n : y^2 = x^3 + A_nx^2 + C_nx$

for $i = 0$ to $n-1$ **do**
  $\alpha_i \leftarrow \sqrt{C_i}$ ▷ Start with arbitrary root
  if $\mathrm{Re}(\alpha_i) \neq 0$ and $\mathrm{Re}(\alpha_i)$ is not a square **then**
    $\alpha_i \leftarrow -\alpha_i$
  else if $\mathrm{Re}(\alpha_i) = 0$ and $\mathrm{Im}(\alpha_i)$ is not a square **then**
    $\alpha_i \leftarrow -\alpha_i$
  end if
  $A_{i+1} \leftarrow 6m_i\alpha_i + A_i$
  $C_{i+1} \leftarrow 4m_i\alpha_iA_i + 8C_i$
end for
return (A_n, C_n)

Embedding $\mathbb{F}_{p^2}$ Arithmetic In $\mathbb{F}_p$

Recall $\mathbb{F}_{p^2} \cong \mathbb{F}_p[j]$ for $j^2 = -1$, and $x = \text{Re}(x) + \text{Im}(x)j \in \mathbb{F}_p[j]$

Addition:

$$a + b = c \iff \begin{array}{c} \text{Re}(a) + \text{Re}(b) = \text{Re}(c) \\ \text{Im}(a) + \text{Im}(b) = \text{Im}(c) \end{array}$$

Squaring:

$$a^2 = b \iff \begin{array}{l} \text{Im}(b) = 2\text{Re}(a)\text{Im}(a) \\ \text{Re}(b) = \big(\text{Re}(a) + \text{Im}(a)\big) \cdot \big(\text{Re}(a) - \text{Im}(a)\big) \end{array}$$

Arbitrary R1CS Constraints:

$$\left(\sum_{l=1}^{n} c_l \mathbf{z}_l\right) \cdot \left(\sum_{r=1}^{n} d_r \mathbf{z}_r\right) = \sum_{o=1}^{n} e_o \mathbf{z}_o$$

$$\Updownarrow$$

$$\left(\sum_{l=1}^{n} c_l \text{Re}(\mathbf{z}_l)\right) \cdot \left(\sum_{r=1}^{n} d_r \text{Re}(\mathbf{z}_r)\right) = \sum_{o=1}^{n} e_o \text{Re}(\mathbf{z}_o) + u \qquad \sum_{l=1}^{n} c_l \text{Im}(\mathbf{z}_l) \cdot \sum_{r=1}^{n} d_r \text{Im}(\mathbf{z}_r) = u$$

$$\left(\sum_{l=1}^{n} c_l \text{Re}(\mathbf{z}_l) + c_l \text{Im}(\mathbf{z}_l)\right) \cdot \left(\sum_{r=1}^{n} d_r \text{Re}(\mathbf{z}_r) + d_r \text{Im}(\mathbf{z}_r)\right) = \left(\sum_{o=1}^{n} e_o \text{Im}(\mathbf{z}_o) + e_o \text{Re}(\mathbf{z}_o)\right) + 2u$$

v variables, s squaring constraints and g general constraints over $\mathbb{F}_p[j]$

$\implies 2v + g$ variables and $2s + 3g$ constraints over $\mathbb{F}_p$

Proof of Radical CGL Evaluation

$$\mathscr{R}_{\text{CGL}} = \{(E_0, E_1), m \mid E_1 = \text{CGL}(E_0, m)\}$$

By rearranging and substituting, the relation is satisfied if,

for all $i = 0, \ldots n-1$, $\exists \beta_i \in \mathbb{F}_p$:

$$\beta_i^2 = \text{Re}(\alpha_i)$$
$$\alpha_i^2 = C_i$$
$$A_{i+1} - A_i = 6 m_i \alpha_i$$
$$2A_i(A_{i+1} - A_i) = 3C_{i+1} - 24C_i$$
$$0 = (m_i + 1)(m_i - 1)$$

- By embedding in $\mathbb{F}_p$, we get

$9n + 4$ variables $9n$ constraints

Algorithm 1 CGL(E_0, m): Novel variant of CGL using radical isogeny formulas

Require: Coordinates $(A_0, C_0) \in \mathbb{F}_{p^2}$ defining a supersingular elliptic curve $E_0 : y^2 = x^3 + A_0x^2 + C_0x$, message $m = m_1 m_2 \ldots m_n \in \{-1, 1\}^n$

Ensure: Coordinates $(A_n, C_n) \in \mathbb{F}_{p^2}$ defining a supersingular elliptic curve $E_n : y^2 = x^3 + A_nx^2 + C_nx$

```
for i = 0 to n − 1 do
    α_i ← √C_i                                        ▷ Start with arbitrary root
    if Re(α_i) ≠ 0 and Re(α_i) is not a square then
        α_i ← −α_i
    else if Re(α_i) = 0 and Im(α_i) is not a square then
        α_i ← −α_i
    end if
    A_{i+1} ← 6 m_i α_i + A_i
    C_{i+1} ← 4 m_i α_i A_i + 8 C_i
end for
return (A_n, C_n)
```

Proof of Radical CGL Evaluation

$$\mathscr{R}_{\mathsf{CGL}} = \{(E_0, E_1), m \mid E_1 = \mathsf{CGL}(E_0, m)\}$$

By rearranging and substituting, the relation is satisfied if, for all $i = 0, \ldots n-1$, $\exists \beta_i \in \mathbb{F}_p$:

$$\beta_i^2 = \mathrm{Re}(\alpha_i)$$
$$\alpha_i^2 = C_i$$
$$A_{i+1} - A_i = 6m_i\alpha_i$$
$$2A_i(A_{i+1} - A_i) = 3C_{i+1} - 24C_i$$
$$0 = (m_i + 1)(m_i - 1)$$

$$b_i \mathrm{Re}(\alpha_i) = 0$$
$$b_i(b_i - 1) = 0$$
$$(b - \mathrm{Re}(\alpha_i)) \cdot \gamma_i = 1$$
$$b_i \mathrm{Im}(\alpha_i) + \mathrm{Re}(\alpha_i) = \beta_i^2$$

- By embedding in $\mathbb{F}_p$, we get

$9n + 4$ variables $9n$ constraints

$13n + 4$ variables $14n$ constraints

Algorithm 1 $\mathsf{CGL}(E_0, m)$: Novel variant of CGL using radical isogeny formulas

Require: Coordinates $(A_0, C_0) \in \mathbb{F}_{p^2}$ defining a supersingular elliptic curve $E_0 : y^2 = x^3 + A_0x^2 + C_0x$, message $m = m_1m_2 \ldots m_n \in \{-1, 1\}^n$

Ensure: Coordinates $(A_n, C_n) \in \mathbb{F}_{p^2}$ defining a supersingular elliptic curve $E_n : y^2 = x^3 + A_nx^2 + C_nx$

```
for i = 0 to n − 1 do
    α_i ← √C_i                                         ▷ Start with arbitrary root
    if [Re(α_i) ≠ 0 and] Re(α_i) is not a square then
        α_i ← −α_i
    else if Re(α_i) = 0 and Im(α_i) is not a square then
        α_i ← −α_i
    end if
    A_{i+1} ← 6 m_i α_i + A_i
    C_{i+1} ← 4 m_i α_i A_i + 8 C_i
end for
return (A_n, C_n)
```

	$\mathscr{R}_{\ell^k\text{–cyclic}}$	$\mathscr{R}_{\mathsf{CGL}}$	$\mathscr{R}_{\mathsf{UPF}}$
Instance Size	$< 2^{11}$	$< 2^{12}$	$< 2^{13}$
Prover Time (ms)	25	45	75
Verification Time (ms)	15	20	25
Proof size (kB)	230	320	430

Table 4.2: Rough performance estimates obtained from the proof system in [BFK+24, Fig. 2] on the Radical isogeny R1CS instances over $\mathbb{F}_p$ (third row of Table 4.1). We set $\lambda = 128$, and hence $k = 256$, and $\log_2 p = 256$.

Open Research Directions

- Building tailored SNARKs for $\mathcal{R}_{\ell^k}$–cyclic
- Constructing R1CS for proving Kani-type isogeny evaluations
- Robust Software Implementations

Thank you! Questions?

Feel free to email me for reference requests, thoughts or concerns!
My Auckland email will disappear soon, so contact me at
shailevanin@gmail.com

Vélu CGL Hash Function

First introduced in [2]:

Input: E_0, message $m \in \{0,1\}^n$, deterministic torsion basis algorithm $\mathscr{B}_{2^n}$

Output: $E_0/\langle P + mQ \rangle$ where $P, Q \leftarrow \mathscr{B}_{2^n}(E_0)$

Representation for E_i:

Montgomery/short weierstrass form

Step Function:

$$\phi_{i+1} : E_i \to E_{i+1} = E_i/\langle P_i + m_i Q_i \rangle$$

where

$$P_0, Q_0 \leftarrow [2^{n-1}]P, [2^{n-1}]Q$$

And for $i > 0$:

$$P_i = [2^{n-i-1}]\phi_i \circ \ldots \cdot \phi_1(P), Q_i = [2^{n-i-1}]\phi_{i-1} \circ \ldots \cdot \phi_1(Q)$$

Direction:

Predetermined by the basis P, Q

[2]: Doliskani, J., Pereira, G.C.C.F., Barreto, P.S.L.M. (2024). Faster Cryptographic Hash Function from Supersingular Isogeny Graphs. In: Smith, B., Wu, H. (eds) Selected Areas in Cryptography. SAC 2022. Lecture Notes in Computer Science, vol 13742. Springer, Cham. https://doi.org/10.1007/978-3-031-58411-4_18