The supersingular Endomorphism Ring and One Endomorphism problems are equivalent

Aurel Page
Inria and Université de Bordeaux

Benjamin Wesolowski
CNRS and ENS de Lyon
An **elliptic curve** over \( \mathbb{F}_q \) is:

a curve of the form

\[ y^2 = x^3 + ax + b \]
Elliptic curves

An **elliptic curve** over $\mathbb{F}_q$ is:

\[ y^2 = x^3 + ax + b \]

\[ y^2 = x^3 - 4x \]
Isogenies

An isogeny is:

A map between two curves

\[ y^2 = x^3 + x \]

\[ y^2 = x^3 - 4x \]
Isogenies

An isogeny is:

a map between two curves

\[ \varphi : E_1 \to E_2 \]

\[ (x, y) \rightarrow \left( \frac{x^2 + 1}{x}, \frac{y(x^2 + 1)}{x^2} \right) \]

\[ y^2 = x^3 - 4x \]
The isogeny problem

**Isogeny problem:** Given two elliptic curves $E_1$ and $E_2$, find an isogeny $\varphi : E_1 \to E_2$

\[
y^2 = x^3 + x
\]

\[
y^2 = x^3 - 4x
\]
The isogeny problem

**Isogeny problem:** Given two elliptic curves $E_1$ and $E_2$, find an isogeny $\varphi : E_1 \rightarrow E_2$

A useful specialization:

**$\ell$-Isogeny Path:** Given two elliptic curves $E_1$ and $E_2$, find an $\ell$-isogeny paths $E_1 \rightarrow E_2$
The isogeny problem

**Isogeny problem:** Given two elliptic curves $E_1$ and $E_2$, find an isogeny $\varphi : E_1 \to E_2$

A useful specialization:

**$\ell$-IsogenyPath:** Given two elliptic curves $E_1$ and $E_2$, find an $\ell$-isogeny paths $E_1 \to E_2$.
Isogeny-based cryptography

**Expectations:** cryptosystems as secure as $\ell$-IsogenyPath is hard

$\ell$-IsogenyPath $=$ Security of cryptosystems

Hard even for quantum algorithms

Post-quantum cryptography
Isogeny-based cryptography

**Reality**: upper and lower bounds

?? \leq \text{Security of cryptosystems} \leq \ell\text{-IsogenyPath}
Isogeny-based cryptography

**Reality**: upper and lower bounds

$\ell$-IsogenyPath $\leq$ Security of cryptosystems $\leq$ $\ell$-IsogenyPath

$\ell$-IsogenyPath $=\text{CGL hash function (preimage)}$
Isogeny-based cryptography

**Reality:** upper and lower bounds

\[
\ell \text{-IsogenyPath} \leq \text{Security of cryptosystems} \leq \ell \text{-IsogenyPath}
\]

\[
\ell \text{-IsogenyPath} = \text{CGL hash function (preimage)}
\]

\[
\text{OneEnd} \leq \text{CGL hash function (collision)}
\]
Isogeny-based cryptography

**Reality:** upper and lower bounds

\[ \ell \text{-IsogenyPath} \leq \text{Security of cryptosystems} \leq \ell \text{-IsogenyPath} \]

- \( \ell \text{-IsogenyPath} \) = CGL hash function (preimage)
- OneEnd \( \leq \) CGL hash function (collision)
- OneEnd \( \leq \) SQLsign (soundness)
Endomorphisms

OneEnd and EndRing
An **endomorphism** of $E$ is an isogeny $\varphi : E \to E$. 

**Endomorphism ring**
Endomorphism ring

An endomorphism of $E$ is an isogeny $\varphi : E \to E$

The endomorphism ring of $E$ is $\text{End}(E) = \{\varphi : E \to E\}$
Endomorphism ring

An endomorphism of $E$ is an isogeny $\phi : E \to E$

The endomorphism ring of $E$ is $\text{End}(E) = \{ \phi : E \to E \}$

• It contains $\mathbb{Z} \subset \text{End}(E)$  \hspace{1cm} (1 = identity, 2 = point doubling, -1 = negation...)
Endomorphism ring

An **endomorphism** of $E$ is an isogeny $\varphi : E \to E$

The **endomorphism ring** of $E$ is $\text{End}(E) = \{\varphi : E \to E\}$

- It contains $\mathbb{Z} \subset \text{End}(E)$ (1 = identity, 2 = point doubling, -1 = negation...)
- $(\text{End}(E), +)$ is a **lattice** of dimension 2 or 4
Endomorphism ring

An **endomorphism** of $E$ is an isogeny $\varphi : E \to E$

The **endomorphism ring** of $E$ is $\text{End}(E) = \{\varphi : E \to E\}$

- It contains $\mathbb{Z} \subset \text{End}(E)$  
  
  \[ (1 = \text{identity}, \ 2 = \text{point doubling}, \ -1 = \text{negation}...) \]

- $(\text{End}(E), +)$ is a **lattice** of dimension 2 or 4
Endomorphism ring

An **endomorphism** of \( E \) is an isogeny \( \varphi : E \to E \)

The **endomorphism ring** of \( E \) is \( \text{End}(E) = \{ \varphi : E \to E \} \)

- It contains \( \mathbb{Z} \subset \text{End}(E) \) (1 = identity, 2 = point doubling, -1 = negation...)
- \( (\text{End}(E), +) \) is a **lattice** of dimension 2 or 4
Endomorphism ring

An **endomorphism** of $E$ is an isogeny $\varphi : E \to E$

The **endomorphism ring** of $E$ is $\text{End}(E) = \{ \varphi : E \to E \}$

- It contains $\mathbb{Z} \subset \text{End}(E)$ (1 = identity, 2 = point doubling, -1 = negation...)
- $(\text{End}(E), +)$ is a **lattice** of dimension 2 or 4
The endomorphism ring problem

**EndRing:** Given a supersingular $E$, find four endomorphisms generating $\text{End}(E)$
The endomorphism ring problem

**EndRing**: Given a supersingular $E$, find four endomorphisms generating $\text{End}(E)$

**Theorem [W. – FOCS 2021]**: $\text{EndRing}$ is equivalent to $\ell$-$\text{IsogenyPath}$ (assuming the Generalised Riemann Hypothesis)

Earlier **heuristic** reductions in:


The [one] endomorphism [ring] problem

**EndRing:** Given a supersingular $E$, find four endomorphisms generating $\text{End}(E)$

**OneEnd:** Given a supersingular $E$, find a single endomorphism $\alpha \in \text{End}(E) \setminus \mathbb{Z}$
The [one] endomorphism [ring] problem

\textbf{EndRing:} Given a supersingular \( E \), find four endomorphisms generating \( \text{End}(E) \)

\textbf{OneEnd:} Given a supersingular \( E \), find a single endomorphism \( \alpha \in \text{End}(E) \setminus \mathbb{Z} \)

Clearly, \textbf{OneEnd} \leq \textbf{EndRing}...
The \text{[one]} \text{ endomorphism} \text{[ring]} \text{problem}

\textbf{EndRing:} Given a supersingular $E$, find four endomorphisms generating $\text{End}(E)$

\textbf{OneEnd:} Given a supersingular $E$, find a single endomorphism $\alpha \in \text{End}(E) \setminus \mathbb{Z}$

Clearly, \textbf{OneEnd} \leq \textbf{EndRing}...

\textbf{Theorem (main result of this work):} \textbf{OneEnd} is equivalent to \textbf{EndRing}, under probabilistic polynomial time reductions
Applications

of OneEnd = EndRing
New security reductions

**Reality:** upper and lower bounds

\[
\ell\text{-IsogenyPath} \leq \text{Security of cryptosystems} \leq \ell\text{-IsogenyPath}
\]

- \(\ell\text{-IsogenyPath} = \text{CGL hash function (preimage)}\)
- \(\text{OneEnd} \leq \text{CGL hash function (collision)}\)
- \(\text{OneEnd} \leq \text{SQIsign (soundness)}\)
New security reductions

**Reality**: upper and lower bounds

\[ ?? \leq \text{Security of cryptosystems} \leq \ell\text{-IsogenyPath} \]

\[ \ell\text{-IsogenyPath} = \text{CGL hash function (preimage)} \]
\[ \text{OneEnd} \leq \text{CGL hash function (collision)} \]
\[ \text{OneEnd} \leq \text{SQIsign (soundness)} \]

= EndRing
New security reductions

**Reality:** upper and lower bounds

\[ \ell \text{-IsogenyPath} \leq \text{Security of cryptosystems} \leq \ell \text{-IsogenyPath} \]

\[ \begin{align*}
\ell \text{-IsogenyPath} &= \text{CGL hash function (preimage)} \\
\text{OneEnd} &= \text{CGL hash function (collision)} \\
\text{OneEnd} &\leq \text{SQIsign (soundness)}
\end{align*} \]

\[ \text{EndRing} = \]
New security reductions

Reality: upper and lower bounds

\[ \ell\text{-IsogenyPath} \leq \text{Security of cryptosystems} \leq \ell\text{-IsogenyPath} \]

\[ \text{EndRing} = \ell\text{-IsogenyPath} \]

\[ \text{OneEnd} \leq \text{CGL hash function (preimage)} \]

\[ \text{OneEnd} \leq \text{CGL hash function (collision)} \]

\[ \text{OneEnd} \leq \text{SQIsign (soundness)} \]

Theorem (Application 1): CGL is collision-resistant if and only if EndRing is hard

Theorem (Application 2): SQIsign is sound if and only if EndRing is hard
EndRing is equivalent to Isogeny

Theorem (Application 3): EndRing is equivalent to the Isogeny problem
Theorem (Application 3): \textbf{EndRing} is equivalent to the \textbf{Isogeny} problem

Previous work:

- \textbf{Isogeny} ≤ \textbf{EndRing}: already known (assuming GRH [W. – FOCS 2021])
- \textbf{EndRing} ≤ \textbf{Isogeny}: only known for special case \(\ell\)-\textbf{IsogenyPath} (assuming GRH)
EndRing is equivalent to Isogeny

Theorem (Application 3): EndRing is equivalent to the Isogeny problem

Previous work:
• Isogeny ≤ EndRing: already known (assuming GRH [W. – FOCS 2021])
• EndRing ≤ Isogeny: only known for special case ℓ-IsogenyPath (assuming GRH)

Idea of the proof: Suffices to show that OneEnd ≤ Isogeny
  ▶ Given E (an instance of OneEnd), sample random isogeny φ : E → F, solve Isogeny to find ψ : F → E, and return ψ ◦ φ (a solution of OneEnd)
  ▶ No need to assume GRH!
Theorem (Application 4): There is an algorithm for EndRing in time $\tilde{O}(p^{1/2})$
Solving EndRing

**Theorem (Application 4):** There is an algorithm for \texttt{EndRing} in time $\tilde{O}(p^{1/2})$

**Previous work:**
- Only known under GRH (see [W. – FOCS 2021], or [Fuselier, Iezzi, Kozek, Morrison, Namoijam – preprint 2023])
- Unconditionally, best known was $\tilde{O}(p)$ [Kohel – PhD thesis 1996]
Solving EndRing

Theorem (Application 4): There is an algorithm for $\text{EndRing}$ in time $\tilde{O}(p^{1/2})$

Previous work:

- Only known under GRH (see [W. – FOCS 2021], or [Fuselier, Iezzi, Kozek, Morrison, Namoijam – preprint 2023])
- Unconditionally, best known was $\tilde{O}(p)$ [Kohel – PhD thesis 1996]

Idea of the proof:

- By the previous application, $\text{EndRing} \leq \text{Isogeny}$ (unconditionally!)
- Meet-in-the-middle solves $\text{Isogeny}$ with complexity $\tilde{O}(p^{1/2})$
Sketch of the proof

Main ideas and obstacles
Reducing EndRing to OneEnd

Suppose we have an oracle $\mathcal{O}$ solving $\text{OneEnd}$

Let $E$ be an instance of $\text{EndRing}$: we wish to find generators of $\text{End}(E)$
Reducing EndRing to OneEnd

Suppose we have an oracle $\mathcal{O}$ solving $\text{OneEnd}$

Let $E$ be an instance of $\text{EndRing}$: we wish to find generators of $\text{End}(E)$

**Idea 0:** Sample until you make it...

1. For $i = 1, 2, \ldots$ call $\mathcal{O}(E)$, which returns some $\alpha_i \in \text{End}(E) \setminus \mathbb{Z}$
2. As soon as $(\alpha_i)_i$ generates $\text{End}(E)$, extract a basis and return it
Reducing EndRing to OneEnd

Suppose we have an oracle $\mathcal{O}$ solving $\text{OneEnd}$

Let $E$ be an instance of $\text{EndRing}$: we wish to find generators of $\text{End}(E)$

**Idea 0:** Sample until you make it...

1. For $i = 1, 2,...$ call $\mathcal{O}(E)$, which returns some $\alpha_i \in \text{End}(E) \setminus \mathbb{Z}$

2. As soon as $(\alpha_i)_i$ generates $\text{End}(E)$, extract a basis and return it
Reducing \text{EndRing} to \text{OneEnd}

Suppose we have an oracle $\mathcal{O}$ solving \text{OneEnd}

Let $E$ be an instance of \text{EndRing}: we wish to find generators of $\text{End}(E)$

**Idea 0:** Sample until you make it...

1. For $i = 1, 2, \ldots$ call $\mathcal{O}(E)$, which returns some $\alpha_i \in \text{End}(E) \setminus \mathbb{Z}$

2. As soon as $(\alpha_i)_i$ generates $\text{End}(E)$, extract a basis and return it

What if $\mathcal{O}(E)$ always returns the same $\alpha$? 🤣

👍 Efficient linear algebra!
Reduction of EndRing to OneEnd

Suppose we have an oracle $\mathcal{O}$ solving OneEnd

Let $E$ be an instance of EndRing: we wish to find generators of $\text{End}(E)$

**Idea 0:** Sample until you make it...

1. For $i = 1, 2, \ldots$ call $\mathcal{O}(E)$, which returns some $\alpha_i \in \text{End}(E) \setminus \mathbb{Z}$
2. As soon as $(\alpha_i)_i$ generates $\text{End}(E)$, extract a basis and return it

**What if $\mathcal{O}(E)$ always returns the same $\alpha$?**

**Idea 1** [Eisenträger, Hallgren, Lauter, Morrison, Petit – Eurocrypt 2018]:

Randomize the oracle...

Efficient linear algebra!
**Enriching the oracle**

**Idea 1: Randomize the oracle**

We construct a new oracle $\text{Rich}_o$
Enriching the oracle

Idea 1: Randomize the oracle
We construct a new oracle $\text{Rich}^\sigma$

On input $E$: 
Idea 1: Randomize the oracle
We construct a new oracle $\text{Rich}^\sigma$

On input $E$:
1. Sample a random isogeny $\varphi : E \to F$
**Enriching the oracle**

**Idea 1: Randomize the oracle**

We construct a new oracle $\text{Rich}^\sigma$

On input $E$:

1. Sample a random isogeny $\varphi : E \to F$
2. Call $\mathcal{O}(F)$ which returns $\alpha \in \text{End}(F) \setminus \mathbb{Z}$
**Idea 1: Randomize the oracle**

We construct a new oracle $\text{Rich}^\sigma$

On input $E$:

1. Sample a random isogeny $\phi : E \to F$
2. Call $\mathcal{O}(F)$ which returns $\alpha \in \text{End}(F) \setminus \mathbb{Z}$
3. Return $\hat{\phi} \circ \alpha \circ \phi \in \text{End}(E) \setminus \mathbb{Z}$
Idea 1: **Randomize** the oracle

1. For \( i = 1, 2, \ldots \) call \( \text{Rich}^o(E) \), which returns some \( \alpha_i \in \text{End}(E) \setminus \mathbb{Z} \)

2. As soon as \( (\alpha_i)_i \) generates \( \text{End}(E) \), extract a basis and return it

Reducing EndRing to OneEnd
Reducing EndRing to OneEnd

**Idea 1: Randomize** the oracle

1. For $i = 1, 2, \ldots$ call $\text{Rich}^\alpha(E)$, which returns some $\alpha_i \in \text{End}(E) \setminus \mathbb{Z}$
2. As soon as $(\alpha_i)_i$ generates $\text{End}(E)$, extract a basis and return it

**Heuristic claim** [Eisenträger, Hallgren, Lauter, Morrison, Petit – Eurocrypt 2018]: $\text{Rich}^\alpha$ is "random enough": it rapidly produces a generating set
Reducing EndRing to OneEnd

Idea 1: Randomize the oracle

1. For \( i = 1, 2, \ldots \) call Rich\( ^o (E) \), which returns some \( \alpha_i \in \text{End}(E) \setminus \mathbb{Z} \)

2. As soon as \( (\alpha_i)_i \) generates \( \text{End}(E) \), extract a basis and return it

Heuristic claim [Eisenträger, Hallgren, Lauter, Morrison, Petit – Eurocrypt 2018]: Rich\( ^o \) is "random enough": it rapidly produces a generating set

Problem: It fails. There exist oracles \( \mathcal{O} \) for which the algorithm does not terminate
Stabilization

Idea 2: Prove that the ring generated by \((\alpha_i)_i\) eventually stabilizes
Stabilization

**Idea 2:** Prove that the ring generated by \((\alpha_i)_i\) eventually stabilizes

**Theorem 1:** The probability distribution of \(\text{Rich}^o(E)\) is stable under conjugation

**In essence:** any output \(\alpha\) is as likely as any conjugate \(\beta^{-1}\alpha\beta\)
**Stabilization**

**Idea 2:** Prove that the ring generated by \((\alpha_i)i\) eventually stabilizes

**Theorem 1:** The probability distribution of \(\text{Rich}^\circ(E)\) is stable under conjugation

*In essence: any output \(\alpha\) is as likely as any conjugate \(\beta^{-1}\alpha\beta\)*

**Theorem 2:** Subrings of \(\text{End}(E)\) stable under conjugation are \(\mathbb{Z} + M \cdot \text{End}(E)\) for \(M \in \mathbb{Z}\)
Stabilization

Idea 2: Prove that the ring generated by $(\alpha_i)_i$ eventually stabilizes

**Theorem 1:** The probability distribution of $\text{Rich}^\circ(E)$ is stable under conjugation

*In essence: any output $\alpha$ is as likely as any conjugate $\beta^{-1}\alpha\beta$*

**Theorem 2:** Subrings of $\text{End}(E)$ stable under conjugation are $\mathbb{Z} + M \cdot \text{End}(E)$ for $M \in \mathbb{Z}$

**Conclusion:** The algorithm *eventually* generates a ring of the form $\mathbb{Z} + M \cdot \text{End}(E)$
Stabilization

Idea 2: Prove that the ring generated by \((\alpha_i)i\) eventually stabilizes

**Theorem 1:** The probability distribution of \(\text{Rich}_\sigma(E)\) is stable under conjugation

*In essence: any output \(\alpha\) is as likely as any conjugate \(\beta^{-1}\alpha\beta\)*

**Theorem 2:** Subrings of \(\text{End}(E)\) stable under conjugation are \(\mathbb{Z} + M \cdot \text{End}(E)\) for \(M \in \mathbb{Z}\)

**Conclusion:** The algorithm *eventually* generates a ring of the form \(\mathbb{Z} + M \cdot \text{End}(E)\)

*From a generating set of \(\mathbb{Z} + M \cdot \text{End}(E)\), one can find a basis of \(\text{End}(E)\)*

👍
Stabilization

Idea 2: Prove that the ring generated by $(\alpha_i)_i$ eventually stabilizes

**Theorem 1:** The probability distribution of $\text{Rich}^\circ(E)$ is stable under conjugation

*In essence: any output $\alpha$ is as likely as any conjugate $\beta^{-1}\alpha\beta$

**Theorem 2:** Subrings of $\text{End}(E)$ stable under conjugation are $\mathbb{Z} + M \cdot \text{End}(E)$ for $M \in \mathbb{Z}$

**Conclusion:** The algorithm **eventually** generates a ring of the form $\mathbb{Z} + M \cdot \text{End}(E)$

*From a generating set of $\mathbb{Z} + M \cdot \text{End}(E)$, one can find a basis of $\text{End}(E)$

"Eventually" = exponential time

👍 "Eventually" = exponential time

👎
Stabilization

Idea 2: Prove that the ring generated by \((\alpha_i)\) eventually stabilizes

**The tough part!**

**Theorem 1:** The probability distribution of \(\text{Rich}^0(E)\) is stable under conjugation

In essence: any output \(\alpha\) is as likely as any conjugate \(\beta^{-1}\alpha\beta\)

**Theorem 2:** Subrings of \(\text{End}(E)\) stable under conjugation are \(\mathbb{Z} + M \cdot \text{End}(E)\) for \(M \in \mathbb{Z}\)

**Conclusion:** The algorithm **eventually** generates a ring of the form \(\mathbb{Z} + M \cdot \text{End}(E)\)

From a generating set of \(\mathbb{Z} + M \cdot \text{End}(E)\), one can find a basis of \(\text{End}(E)\)

"Eventually" = exponential time
Stabilization

**Idea 2:** Prove that the ring generated by \((\alpha_i)\) eventually stabilizes.

**The tough part!**

**Theorem 1:** The probability distribution of \(\text{Rich}^\circ(E)\) is stable under conjugation.

In essence: any output \(\alpha\) is as likely as any conjugate \(\beta^{-1}\alpha\beta\).

**Deuring correspondence**

**Theorem 2:** Subrings of \(\text{End}(E)\) stable under conjugation are \(\mathbb{Z} + M \cdot \text{End}(E)\) for \(M \in \mathbb{Z}\).

**Conclusion:** The algorithm *eventually* generates a ring of the form \(\mathbb{Z} + M \cdot \text{End}(E)\).

From a generating set of \(\mathbb{Z} + M \cdot \text{End}(E)\), one can find a basis of \(\text{End}(E)\).

"Eventually" = exponential time.
Idea 2: Prove that the ring generated by \((\alpha_i)_i\) eventually stabilizes.

Theorem 1: The probability distribution of \(\text{Rich}^\circ(E)\) is stable under conjugation.

In essence: any output \(\alpha\) is as likely as any conjugate \(\beta^{-1}\alpha\beta\).

The tough part!

Deuring correspondence

Theorem 2: Subrings of \(\text{End}(E)\) stable under conjugation are \(\mathbb{Z} + M \cdot \text{End}(E)\) for \(M \in \mathbb{Z}\).

Jacquet-Langlands correspondence

Conclusion: The algorithm eventually generates a ring of the form \(\mathbb{Z} + M \cdot \text{End}(E)\).

From a generating set of \(\mathbb{Z} + M \cdot \text{End}(E)\), one can find a basis of \(\text{End}(E)\).

"Eventually" = exponential time.
Idea 2: Prove that the ring generated by \((\alpha_i)\) eventually stabilizes

Theorem 1: The probability distribution of \(\text{Rich}^\sigma(E)\) is stable under conjugation

In essence: any output \(\alpha\) is as likely as any conjugate \(\beta^{-1}\alpha\beta\)

**The tough part!**

**Deuring correspondence**

Theorem 2: Subrings of \(\text{End}(E)\) stable under conjugation are \(\mathbb{Z} + M \cdot \text{End}(E)\) for \(M \in \mathbb{Z}\)

**Jacquet-Langlands correspondence**

**Conclusion:** The algorithm **eventually** generates a ring of the form \(\mathbb{Z} + M \cdot \text{End}(E)\)

From a generating set of \(\mathbb{Z} + M \cdot \text{End}(E)\), one can find a basis of \(\text{End}(E)\)

"Eventually" = exponential time

**Deligne's bound on coefficients of modular forms**
Reducing EndRing to OneEnd

Outline of the reduction:

1. Initialize $S = \{ 1 \}$
2. While $S$ does not generate a ring of the form $\mathbb{Z} + M \cdot \text{End}(E)$, do:
   3. Sample $\alpha \leftarrow \text{Rich}^\theta(E)$
   4. $\alpha \leftarrow \text{LazyReduce}(\alpha)$ (Idea 3)
   5. Add $\alpha$ to $S$
3. Extract from $S$ a basis of $\text{End}(E)$, and return it
Reducing EndRing to OneEnd

Outline of the reduction:

1. Initialize $S = \{1\}$

2. While $S$ does not generate a ring of the form $\mathbb{Z} + M \cdot \text{End}(E)$, do:
   
   3. Sample $\alpha \leftarrow \text{Rich}^\emptyset(E)$
   
   4. $\alpha \leftarrow \text{LazyReduce}(\alpha)$ (Idea 3)
   
   5. Add $\alpha$ to $S$

6. Extract from $S$ a basis of $\text{End}(E)$, and return it

Terminates! 👍
Reducing EndRing to OneEnd

Outline of the reduction:

1. Initialize $S = \{1\}$
2. While $S$ does not generate a ring of the form $\mathbb{Z} + M \cdot \text{End}(E)$, do:
   3. Sample $\alpha \leftarrow \text{Rich}^0(E)$
   4. $\alpha \leftarrow \text{LazyReduce}(\alpha)$ (Idea 3)
   5. Add $\alpha$ to $S$
3. Extract from $S$ a basis of $\text{End}(E)$, and return it

Terminates! 👍

In exponential time... 👎
Reducing EndRing to OneEnd

Outline of the reduction:

1. Initialize $S = \{1\}$
2. While $S$ does not generate a ring of the form $\mathbb{Z} + M \cdot \text{End}(E)$, do:
   3. Sample $\alpha \leftarrow \text{Rich}^\varnothing(E)$
   4. $\alpha \leftarrow \text{LazyReduce}(\alpha)$ \textbf{(Idea 3)}
   5. Add $\alpha$ to $S$
3. Extract from $S$ a basis of $\text{End}(E)$, and return it
Reducing EndRing to OneEnd

Outline of the reduction:
1. Initialize $S = \{1\}$
2. While $S$ does not generate a ring of the form $\mathbb{Z} + M \cdot \text{End}(E)$, do:
   3. Sample $\alpha \leftarrow \text{Rich}^0(E)$
   4. $\alpha \leftarrow \text{LazyReduce}(\alpha)$ (Idea 3)
   5. Add $\alpha$ to $S$
3. Extract from $S$ a basis of $\text{End}(E)$, and return it

Polynomial time!
OneEnd to find them all