Qlapoti and qt-Pegasis:

Simpler and faster ideal-to-isogeny translation

Setting:

"Effective primitive embedding"

- E elliptic curve
- $\mathrm{End}(E) \supseteq O$ quadratic order OR maximal quaternion order
- $I = O\langle N, \alpha \rangle$ a (primitive, invertible) ideal of with $\mathrm{nrd}(I) = N$

Goal:

- Compute ϕ_I

$I = \langle N, \alpha \rangle$

Some preliminaries

ϕ_I is **defined by** $\ker \phi_I = \{P \in E \mid \beta(P) = 0, \forall \beta \in I\}$
$= E[N] \cap \ker \alpha$

$I = \langle N, \alpha \rangle$

Some preliminaries

ϕ_I is **defined by** $\ker \phi_I = \{P \in E \mid \beta(P) = 0, \forall \beta \in I\}$
$= E[N] \cap \ker \alpha$

We are free to replace ϕ_I by $\phi_{I'}$ where $I' = I\beta$

$$\begin{tikzcd} & E_I \arrow[d, phantom, sloped, "\cong"] \\ E \arrow[ur, bend left, "\phi_I"] \arrow[r, bend right, "\phi_{I'}"'] & E_{I'} \end{tikzcd}$$

$I = \langle N, \alpha \rangle$

Some preliminaries

ϕ_I is **defined by** $\ker \phi_I = \{P \in E \mid \beta(P) = 0, \forall \beta \in I\}$
$= E[N] \cap \ker \alpha$

We are free to replace ϕ_I by $\phi_{I'}$ where $I' = I\beta$

$$\begin{tikzcd} E \arrow[r, bend left, "\phi_I"] \arrow[dr, bend right, "\phi_{I'}"'] & E_I \arrow[d, phantom, "\wr|"] \\ & E_{I'} \end{tikzcd}$$

First idea:

- Assume N_I smooth
- Recover $\ker \phi_I$

$$\boxed{I = \langle N, \alpha \rangle}$$

"Recover $\ker \phi_I$"

$$\begin{tikzcd} E \arrow[r, "\phi_I"] & E_I \arrow[l, bend left, "\gamma"] \end{tikzcd} \qquad \alpha = \gamma \circ \phi_I$$

Idea: Project $E_I[N]$ onto $\ker \phi_I$

$$I = \langle N, \alpha \rangle$$

"Recover $\ker \phi_I$"

$$\begin{tikzcd} E \arrow[rr, "\phi_I"] & & E_I \arrow[ll, bend left, "\gamma"] \end{tikzcd} \qquad \alpha = \gamma \circ \phi_I$$

Idea: Project $E_I[N]$ onto $\ker \phi_I$

$$\begin{aligned} \ker \phi_I &= \{ \widehat{\phi_I}(P) \mid P \in E_I[N] \} \\ &= \{ \widehat{\phi_I}(\hat{\gamma}(P)) \mid P \in E[N] \} \\ &= \{ \hat{\alpha}(P) \mid P \in E[N] \} \end{aligned}$$

$$I = \langle N, \alpha \rangle$$

"Recover $\ker \phi_I$"

$$\begin{tikzcd} E \arrow[r, "\phi_I"] & E_I \arrow[l, bend left, "\gamma"] \end{tikzcd} \qquad \alpha = \gamma \circ \phi_I$$

Idea: Project $E_I[N]$ onto $\ker \phi_I$

$$\begin{aligned} \ker \phi_I &= \{ \widehat{\phi_I}(P) \mid P \in E_I[N] \} \\ &= \{ \widehat{\phi_I}(\hat{\gamma}(P)) \mid P \in E[N] \} \\ &= \{ \hat{\alpha}(P) \mid P \in E[N] \} \end{aligned}$$

"Often" enough to take a single point of order N

$I = \langle N, \alpha \rangle$

"Assume N is smooth"

I quaternion ideal: 🌈🪄KLPT🪄🌈

$I = \langle N, \alpha \rangle$

"Assume N is smooth"

I quaternion ideal: 🌈🪄KLPT🪄🌈

Requires $N > p^3$,
creates many complications

Historically the main building block for
generic ideal-to-isogeny translation,
e.g. (old) SQIsign, Deuring for the people etc.

$I = \langle N, \alpha \rangle$

"Assume N is smooth"

I quaternion ideal: 🌈🪄KLPT🪄🌈

Requires $N > p^3$,
creates many complications

Historically the main building block for
generic ideal-to-isogeny translation,
e.g. (old) SQIsign, Deuring for the people etc.

I quadratic ideal: No polynomial time algorithm in practice

$$I = \langle N, \alpha \rangle$$

"Assume N is smooth"

I quaternion ideal: 🌈🪄KLPT🪄🌈

Requires $N > p^3$,
creates many complications

Historically the main building block for
generic ideal-to-isogeny translation,
e.g. (old) SQIsign, Deuring for the people etc.

I quadratic ideal: No polynomial time algorithm in practice

CSIDH: Only sample I smooth

SCALLOP and friends:
Exponential time precomputation

$$
\begin{tikzcd}
E_{\bar{I}} \arrow[r] & E \\
E \arrow[u] \arrow[r, "\phi_I"] & E_I \arrow[u, "\widehat{\phi_J}"']
\end{tikzcd}
\qquad\qquad
\Phi :
\begin{tikzcd}
E \arrow[r, "\phi_I"] \arrow[dr] & E_I \\
E \arrow[ur, "\phi_J"] \arrow[r] & E_{\bar{I}}
\end{tikzcd}
$$

$$
\begin{tikzcd}
E_{\bar{I}} \arrow[r] & E \\
E \arrow[u] \arrow[r, "\phi_I"] & E_I \arrow[u, "\widehat{\phi_J}"']
\end{tikzcd}
\qquad
\Phi : \begin{tikzcd}
E \arrow[r, "\phi_I"] \arrow[dr] & E_I \\
E \arrow[ur, "\phi_J"] \arrow[r] & E_{\bar{I}}
\end{tikzcd}
$$

New idea:

- Assume $I \sim J$ with $\mathrm{nrd}(I) + \mathrm{nrd}(J) = 2^e$
- Recover $\ker \Phi$

$$
\begin{tikzcd}
E_{\bar{I}} \arrow[r] & E \\
E \arrow[u] \arrow[r, "\phi_I"] & E_I \arrow[u, "\widehat{\phi_J}"']
\end{tikzcd}
$$

$$
\Phi : \begin{tikzcd}
E \arrow[r, "\phi_I"] \arrow[dr] & E_I \\
E \arrow[ur, "\phi_J"] \arrow[r] & E_{\bar{I}}
\end{tikzcd}
$$

$$
\begin{aligned}
\ker \Phi &= \{(\widehat{\phi_I}(P), \widehat{\phi_J}(Q)) \mid P, Q \in E_I[2^e]\} \\
&= \{(\phi_I \circ \widehat{\phi_I}(P), \phi_I \circ \widehat{\phi_J}(Q)) \mid P, Q \in E[2^e]\} \\
&= \{([N_I]P, \gamma(Q)) \mid P, Q \in E[2^e]\}
\end{aligned}
$$

New idea:

- Assume $I \sim J$ with $\mathrm{nrd}(I) + \mathrm{nrd}(J) = 2^e$
- Recover $\ker \Phi$

$$
\begin{tikzcd}
F \arrow[rr] & & E_v \\
& & E \arrow[u, "\phi_v"'] \\
E_u \arrow[uu] & E \arrow[l, "\phi_u"'] \arrow[r, "\phi_I"] & E_I \arrow[u, "-\widehat{\phi_J}"']
\end{tikzcd}
$$

$$
\Phi : \begin{tikzcd}
E_u \arrow[r, "\phi_I \circ \widehat{\phi_u}"] \arrow[dr] & E_I \\
E_v \arrow[ur, "\phi_J \circ \widehat{\phi_v}" near start] \arrow[r] & F
\end{tikzcd}
$$

$$
\begin{tikzcd}
F \arrow[rr] & & E_v \\
 & & E \arrow[u, "\phi_v"'] \\
E_u \arrow[uu] & E \arrow[l, "\phi_u"'] \arrow[r, "\phi_I"] & E_I \arrow[u, "-\widehat{\phi_J}"']
\end{tikzcd}
$$

$$
\Phi : \begin{tikzcd}
E_u \arrow[r, "\phi_I \circ \widehat{\phi_u}"] \arrow[dr] & E_I \\
E_v \arrow[ur, "\phi_J \circ \widehat{\phi_v}"] \arrow[r] & F
\end{tikzcd}
$$

$$\ker \Phi = \{(\widehat{\phi_u}([N_I]P), \widehat{\phi_v}(\gamma(Q))) \mid P, Q \in E[2^e]\}$$

New idea:

- Assume $I \sim J$ and $u, v \in \mathbb{N}$ with $uN_I + vN_J = 2^e$
- Recover $\ker \Phi$

$$
\begin{tikzcd}
F \arrow[rr] & & E_v \\
& & E \arrow[u, "\phi_v"'] \\
E_u \arrow[uu] & E \arrow[l, "\phi_u"'] \arrow[r, "\phi_I"] & E_I \arrow[u, "-\widehat{\phi_J}"']
\end{tikzcd}
$$

$$
\Phi : \begin{tikzcd}
E_u \arrow[r, "\phi_I \circ \widehat{\phi_u}"] \arrow[dr] & E_I \\
E_v \arrow[ur, "\phi_J \circ \widehat{\phi_v}" near start] \arrow[r] & F
\end{tikzcd}
$$

New idea:

- Assume $I \sim J$ and $u, v \in \mathbb{N}$ with $uN_I + vN_J = 2^e$
- Recover $\ker \Phi$

$$\ker \Phi = \{(\widehat{\phi_u}([N_I]P), \widehat{\phi_v}(\gamma(Q))) \mid P, Q \in E[2^e]\}$$

Requires computing **random** isogenies of prescribed degree u, v

The norm equation

Given $I \subset \mathcal{O}_0$ find $\beta_1, \beta_2 \in I$ and $u, v \in \mathbb{Z}_{\geq 0}$,
such that $u \cdot n(\beta_1) + v \cdot n(\beta_2) = 2^e \cdot n(I)$

The norm equation

Given $I \subset \mathcal{O}_0$ find $\beta_1, \beta_2 \in I$ and $u, v \in \mathbb{Z}_{\geq 0}$,
such that $u \cdot n(\beta_1) + v \cdot n(\beta_2) = 2^e \cdot n(I)$

Step 1: Find the smallest $\beta_1, \beta_2 \in I$ of coprime norm
(so in particular, β_1, β_2 must be independent)

The norm equation

Given $I \subset \mathcal{O}_0$ find $\beta_1, \beta_2 \in I$ and $u, v \in \mathbb{Z}_{\geq 0}$,
such that $u \cdot n(\beta_1) + v \cdot n(\beta_2) = 2^e \cdot n(I)$

Step 1: Find the smallest $\beta_1, \beta_2 \in I$ of coprime norm
(so in particular, β_1, β_2 must be independent)
Step 2: Solve for u, v

The norm equation

Given $I \subset \mathcal{O}_0$ find $\beta_1, \beta_2 \in I$ and $u, v \in \mathbb{Z}_{\geq 0}$,
such that $u \cdot n(\beta_1) + v \cdot n(\beta_2) = 2^e \cdot n(I)$

Step 1: Find the smallest $\beta_1, \beta_2 \in I$ of coprime norm
(so in particular, β_1, β_2 must be independent)
Step 2: Solve for u, v

Often a bit larger :(

Expected to find $n(\beta_1)/n(I) \approx n(\beta_2)/n(I) \approx \sqrt{p}$,
and solution is guaranteed when $2^e > n(\beta_1)n(\beta_2)/n(I)^2$

Must be a few bits
smaller than p

Clapoti Issues - Quaternion ideals

The current way of solving the norm equation fails with non-negligible probability

Clapoti Issues - Quaternion ideals

The current way of solving the norm equation fails with non-negligible probability

Leads to a complicated rerandomisation procedure to bring failure probability down to 2^{-60}

Still not negligible in security parameter leads to gap in security proof

Clapoti Issues - Quaternion ideals

The current way of solving the norm equation fails with non-negligible probability

Leads to a complicated rerandomisation procedure to bring failure probability down to 2^{-60}

Still not negligible in security parameter leads to gap in security proof

Random isogenies of degree u and v: QFESTA, done by computing an isogeny in dimension 2.

Clapoti Issues - Quaternion ideals

The current way of solving the norm equation fails with non-negligible probability

Leads to a complicated rerandomisation procedure to bring failure probability down to 2^{-60}

Still not negligible in security parameter leads to gap in security proof

Random isogenies of degree u and v: QFESTA, done by computing an isogeny in dimension 2.

So currently, translating an ideal to curve requires one $(2^e, 2^e)$-isogeny and two $(2^f, 2^f)$-isogenies $(f \approx e/2)$

Clapoti Issues - Quadratic ideals

Random isogenies of degree u and v: Can only be done in dimension 2, which even assumes u, v can be written as sums of squares

Clapoti Issues - Quadratic ideals

Random isogenies of degree u and v: Can only be done in dimension 2, which even assumes u, v can be written as sums of squares

Leads to embedding ϕ_I, ϕ_J in (diagonal) 2-dimensional isogenies, and the final isogeny must be computed in dimension 4

Clapoti Issues - Quadratic ideals

Random isogenies of degree u and v: Can only be done in dimension 2, which even assumes u, v can be written as sums of squares

Leads to embedding ϕ_I, ϕ_J in (diagonal) 2-dimensional isogenies, and the final isogeny must be computed in dimension 4

Failure probability so high, extra tricks must be used to make it work (see: PEGASIS)

Qlapoti:

Simple and Efficient Translation of Quaternion Ideals to Isogenies

Joint work with: Giacomo Borin, Maria Corte-Real Santos, Riccardo Invernizzi, Marzio Mula, Sina Schaeffler and Frederik Vercauteren

Jonathan Komada Eriksen,
COSIC, KU Leuven

$$
\begin{tikzcd}
E_{\bar{I}} \arrow[r] & E \\
E \arrow[u] \arrow[r, "\phi_I"] & E_I \arrow[u, swap, "\widehat{\phi_J}"]
\end{tikzcd}
\qquad\qquad
\Phi :
\begin{tikzcd}
E \arrow[r, "\phi_I"] \arrow[dr] & E_I \\
E \arrow[ur, "\phi_J"] \arrow[r] & E_{\bar{I}}
\end{tikzcd}
$$

"Assume $I \sim J$ with $\mathrm{nrd}(I) + \mathrm{nrd}(J) = 2^e$"

Given $I \subset \mathcal{O}_0$ find $\beta_1, \beta_2 \in I$ such that $n(\beta_1) + n(\beta_2) = 2^e \cdot n(I)$

Idea: Solve equation directly

Given $I = \mathcal{O}_0\langle N, \alpha\rangle$ find $\beta_1, \beta_2 \in I$ such that $n(\beta_1) + n(\beta_2) = 2^e \cdot n(I)$

Idea: Solve equation directly

Given $I = \mathcal{O}_0\langle N, \alpha \rangle$ find $\beta_1, \beta_2 \in I$ such that $n(\beta_1) + n(\beta_2) = 2^e \cdot n(I)$

Very easy algorithm that sort of works: Same as u, v method, but restrict u, v to be sums of squares

Failure probability goes from bad to worse...

Idea: Solve equation directly

Given $I = \mathcal{O}_0\langle N, \alpha \rangle$ find $\beta_1, \beta_2 \in I$ such that $n(\beta_1) + n(\beta_2) = 2^e \cdot n(I)$

~~Very easy algorithm that sort of works: Same as u, v method, but restrict u, v to be sums of squares~~

~~Failure probability goes from bad to worse...~~

Idea: Solve equation directly

Given $I = \mathcal{O}_0\langle N, \alpha\rangle$ find $\beta_1, \beta_2 \in I$ such that $n(\beta_1) + n(\beta_2) = 2^e \cdot n(I)$

Key: Look for $\beta_k = (a_k + ib_k) \cdot N + \alpha$, for $k = 1,2$

Idea: Solve equation directly

Given $I = \mathcal{O}_0\langle N, \alpha\rangle$ find $\beta_1, \beta_2 \in I$ such that $n(\beta_1) + n(\beta_2) = 2^e \cdot n(I)$

Key: Look for $\beta_k = (a_k + ib_k) \cdot N + \alpha$, for $k = 1,2$

$$N(a_1^2 + b_1^2 + a_2^2 + b_2^2) + 2n(\alpha)/N + 2(a_\alpha(a_1 + a_2) + b_\alpha(b_1 + b_2)) = 2^e$$

Idea: Solve equation directly

Given $I = \mathcal{O}_0\langle N, \alpha\rangle$ find $\beta_1, \beta_2 \in I$ such that $n(\beta_1) + n(\beta_2) = 2^e \cdot n(I)$

Key: Look for $\beta_k = (a_k + ib_k) \cdot N + \alpha$, for $k = 1,2$

$$N(a_1^2 + b_1^2 + a_2^2 + b_2^2) + 2n(\alpha)/N + 2(a_\alpha(a_1 + a_2) + b_\alpha(b_1 + b_2)) = 2^e$$

Step 1: Find short A, B such that $2(a_\alpha A + b_\alpha B) \equiv 2^e - 2n(\alpha)/N \pmod{N}$

Idea: Solve equation directly

Given $I = \mathcal{O}_0\langle N, \alpha\rangle$ find $\beta_1, \beta_2 \in I$ such that $n(\beta_1) + n(\beta_2) = 2^e \cdot n(I)$

Key: Look for $\beta_k = (a_k + ib_k) \cdot N + \alpha$, for $k = 1,2$

$$N(a_1^2 + b_1^2 + a_2^2 + b_2^2) + 2n(\alpha)/N + 2(a_\alpha(a_1 + a_2) + b_\alpha(b_1 + b_2)) = 2^e$$

Step 1: Find short A, B such that $2(a_\alpha A + b_\alpha B) \equiv 2^e - 2n(\alpha)/N \pmod{N}$

$$a_1^2 + b_1^2 + (A - a_1)^2 + (B - b_1)^2 = M$$

$$\frac{2^e - 2n(\alpha)/N - 2(a_\alpha A + b_\alpha B))}{N}$$

Idea: Solve equation directly

Given $I = \mathcal{O}_0\langle N, \alpha\rangle$ find $\beta_1, \beta_2 \in I$ such that $n(\beta_1) + n(\beta_2) = 2^e \cdot n(I)$

Key: Look for $\beta_k = (a_k + ib_k) \cdot N + \alpha$, for $k = 1,2$

$$N(a_1^2 + b_1^2 + a_2^2 + b_2^2) + 2n(\alpha)/N + 2(a_\alpha(a_1 + a_2) + b_\alpha(b_1 + b_2)) = 2^e$$

Step 1: Find short A, B such that $2(a_\alpha A + b_\alpha B) \equiv 2^e - 2n(\alpha)/N \pmod{N}$

$$a_1^2 + b_1^2 + (A - a_1)^2 + (B - b_1)^2 = M$$

Step 2: Use Cornacchia to solve

$$(2a_1 - A)^2 + (2b_1 - B)^2 = 2M - A^2 - B^2$$

Idea: Solve equation directly

Minkowski: $N < 2\sqrt{2p}/\pi$

Given $I = \mathcal{O}_0\langle N, \alpha\rangle$ find $\beta_1, \beta_2 \in I$ such that $n(\beta_1) + n(\beta_2) = 2^e \cdot n(I)$

Choose $n(\alpha)/N < 2^e$ (Not restrictive, expect to find $n(\alpha)/N \approx \sqrt{p}$)

Expect to find A, B with $A \approx B \approx \sqrt{N}$

Step 1: Find short A, B such that $2(a_\alpha A + b_\alpha B) \equiv 2^e - 2n(\alpha)/N \pmod{N}$

$$\frac{2^e - 2n(\alpha)/N - 2(a_\alpha A + b_\alpha B))}{N}$$

Step 2: Use Cornacchia to solve

$$(2a_1 - A)^2 + (2b_1 - B)^2 = 2M - A^2 - B^2$$

So all we need is $A^2 + B^2 \lesssim 2^e/N$, and we try new α until this is satisfied

Failure probability for SQIsign parameters

NIST level	p	c	e	upper bound on failure rate
I	$2^{248} \cdot 5 - 1$	2185	246	2^{-197}
III	$2^{376} \cdot 65 - 1$	38495	374	2^{-312}
V	$2^{500} \cdot 27 - 1$	21484	498	2^{-438}

Table 3. The final upper bound of the failure rate of Qlapoti applied to the SQIsign parameters.

Results in SageMath

NIST level	Previous work [5]	This work	Improvement
I	0.415s	0.160s	x2.595
III	0.768s	0.346s	x2.222
V	1.060s	0.467s	x2.269

Table 5. Timings comparing IdealToIsogeny using the technique currently used in SQIsign and the one presented in this work, given in wall-clock time. The final column represents the improvement factor.

Results in SageMath

Protocol	Algorithm	Previous work	This work	Improvement
SQIsign-LVLI	KeyGen	0.489s	0.249s	x1.961
	Signing	1.010s	0.522s	x1.935
PRISM-LVLI	KeyGen	0.484s	0.252s	x1.929
	Signing	0.593s	0.322s	x1.673
PRISM-LVL3	KeyGen	0.915s	0.544s	x1.682
	Signing	1.328s	0.808s	x1.644
PRISM-LVL5	KeyGen	1.436s	0.758s	x1.894
	Signing	2.017s	1.426s	x1.415

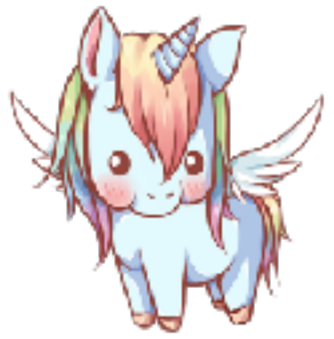
Table 6. Preliminary benchmarks in SageMath to measure the impact of Qlapoti on the signature schemes SQIsign and PRISM. The comparison with PRISM is with the implementation from [5], while the comparison with SQIsign uses a preliminary proof-of-concept implementation privately shared by the authors.

Results in C

Coming soon...

NIST level	Previous work [10]	This work
I	$75,5$ KiB	$33,5$ KiB
III	337 KiB	$49,2$ KiB
V	347 KiB	$64,6$ KiB

Table 7. Heap usage by a reference/Release build of the SQIsign NIST2 implementation with and without Qlapoti. Average over 10 runs. Measures were taken with the sqisign_test_scheme_lvl[x] executable for level x.

qt-PEGASIS: 

Applying Qlapoti to PEGASIS

Joint work with Riccardo Invernizzi and Frederik Vercauteren

Jonathan Komada Eriksen,
COSIC, KU Leuven

$$\begin{tikzcd}
A \arrow[rr] & & E \times E \\
E \times E \arrow[u] \arrow[rr, "\Phi_{J_1,J_2}"] & & E_{\bar{I}} \times E_I \arrow[u, "\widetilde{\Phi}_{K_1,K_2}"']
\end{tikzcd}$$

$$\Phi_{J_1,J_2} : \quad \begin{tikzcd}
E \arrow[rr, "\phi_{J_1}"] \arrow[drr, "-\phi_{\overline{J_2}}"' near start] & & E_I \\
E \arrow[rr, "\phi_{\overline{J_1}}"'] \arrow[urr, "\phi_{J_2}" near start] & & E_{\bar{I}}
\end{tikzcd}$$

$$\Phi_{J_1,J_2} : \begin{tikzcd} E \arrow[r, "\phi_{J_1}"] \arrow[dr, "-\phi_{\overline{J_2}}" near start] & E_I \\ E \arrow[ur, "\phi_{J_2}" near start] \arrow[r, "\phi_{\overline{J_1}}"'] & E_{\bar{I}} \end{tikzcd}$$

$$\begin{tikzcd} A \arrow[r] & E \times E \\ E \times E \arrow[u] \arrow[r, "\Phi_{J_1,J_2}"] & E_{\bar{I}} \times E_I \arrow[u, "\widetilde{\Phi}_{K_1,K_2}"'] \end{tikzcd}$$

Kani (twice!): $\Gamma : E \times E \times E \times E \to A \times E_{\bar{I}} \times E_I$ has

$$\deg \Gamma = \deg \Phi_{J_1,J_2} + \deg \Phi_{K_1,K_2} = N(J_1) + N(J_2) + N(K_1) + N(K_2)$$

$$\begin{tikzcd}
A \arrow[r] & E \times E \\
E \times E \arrow[u] \arrow[r, "\Phi_{J_1,J_2}"] & E_{\bar{I}} \times E_I \arrow[u, "\widetilde{\Phi}_{K_1,K_2}"']
\end{tikzcd}$$

$$\Phi_{J_1,J_2} : \begin{tikzcd}
E \arrow[r, "\phi_{J_1}"] \arrow[dr, "-\phi_{\overline{J_2}}" near end] & E_I \\
E \arrow[ur, "\phi_{J_2}" near start] \arrow[r, "\phi_{\overline{J_1}}"'] & E_{\bar{I}}
\end{tikzcd}$$

Kani (twice!): $\Gamma : E \times E \times E \times E \to A \times E_{\bar{I}} \times E_I$ has

$$\deg \Gamma = \deg \Phi_{J_1,J_2} + \deg \Phi_{K_1,K_2} = N(J_1) + N(J_2) + N(K_1) + N(K_2)$$

Given $I \subset R$ find $\beta_1, \beta_2, \delta_1, \delta_2 \in I$ such that $n(\beta_1) + n(\beta_2) + n(\delta_1) + n(\delta_2) = 2^e \cdot n(I)$

Qlapoti already solves this!

As in KLaPoTi: $R = \mathbb{Z} + \frac{1+j}{2}\mathbb{Z}, j^2 = -p$

Then
$O = R + iR, i^2 = -1, ij = -ji = -p,$ is the "typical" quaternion order!

Qlapoti already solves this!

As in KLaPoTi: $R = \mathbb{Z} + \frac{1+j}{2}\mathbb{Z}, j^2 = -p$

Then
$O = R + iR, i^2 = -1, ij = -ji = -p$, is the "typical" quaternion order!

KLPT: Find $\gamma = \beta_1 + i\beta_2 \in I + iI$
such that $n(\gamma) = n(\beta_1) + n(\beta_2) = 2^e N > p^3 N$

Qlapoti already solves this!

As in KLaPoTi: $R = \mathbb{Z} + \frac{1+j}{2}\mathbb{Z}, j^2 = -p$

Then
$O = R + iR, i^2 = -1, ij = -ji = -p$, is the "typical" quaternion order!

KLPT: Find $\gamma = \beta_1 + i\beta_2 \in I + iI$
such that $n(\gamma) = n(\beta_1) + n(\beta_2) = 2^e N > p^3 N$

Qlapoti: Find $\gamma_1 = \beta_1 + i\beta_2 \in I + iI, \gamma_2 = \delta_1 + \delta_2 \in I + iI,$
such that $n(\gamma_1) + n(\gamma_2) = n(\beta_1) + n(\beta_2) + n(\delta_1) + n(\delta_2) = 2^e N \gtrsim pN$

Optimized Qlapoti for these ideals

Ideals of the form $I + iI \subseteq R + iR = \mathcal{O}$ are not "generic" at all

Correspond to isogenies defined over $\mathbb{F}_p$

Optimized Qlapoti for these ideals

Ideals of the form $I + iI \subseteq R + iR = O$ are not "generic" at all

Correspond to isogenies defined over $\mathbb{F}_p$

Can make an **extremely fast** Qlapoti variant for this: Let $I = R\langle N, \omega - \lambda \rangle$

$$\beta_k = b_k N + d_k(\omega - \lambda), k = 1,2$$
$$\delta_k = a_k N, k = 1,2$$

Optimized Qlapoti for these ideals

Ideals of the form $I + iI \subseteq R + iR = O$ are not "generic" at all

Correspond to isogenies defined over $\mathbb{F}_p$

Can make an **extremely fast** Qlapoti variant for this: Let $I = R\langle N, \omega - \lambda\rangle$

$$\beta_k = b_k N + d_k(\omega - \lambda), k = 1,2$$
$$\delta_k = a_k N, k = 1,2$$

$$N(a_1^2 + a_2^2 + b_1^2 + b_2^2) + (n(\omega - \lambda)/N)(d_1^2 + d_2^2) + 2\lambda(b_1 d_1 + b_2 d_2) = 2^e$$

Optimized Qlapoti for these ideals

Ideals of the form $I + iI \subseteq R + iR = O$ are not "generic" at all

Correspond to isogenies defined over $\mathbb{F}_p$

Can make an **extremely fast** Qlapoti variant for this: Let $I = R\langle N, \omega - \lambda\rangle$

Can fix $d_1 = 1$, and start with $d_2 = c$ where c is as large as possible.

$$\beta_k = b_k N + d_k(\omega - \lambda), k = 1{,}2$$
$$\delta_k = a_k N, k = 1{,}2$$

$$N(a_1^2 + a_2^2 + b_1^2 + b_2^2) + (n(\omega - \lambda)/N)(d_1^2 + d_2^2) + 2\lambda(b_1 d_1 + b_2 d_2) = 2^e$$

Optimized Qlapoti for these ideals

Ideals of the form $I + iI \subseteq R + iR = O$ are not "generic" at all

Correspond to isogenies defined over $\mathbb{F}_p$

Can make an **extremely fast** Qlapoti variant for this: Let $I = R\langle N, \omega - \lambda \rangle$

$$\beta_k = b_k N + d_k(\omega - \lambda), k = 1{,}2$$
$$\delta_k = a_k N, k = 1{,}2$$

Can fix $d_1 = 1$, and start with $d_2 = c$ where c is as large as possible.

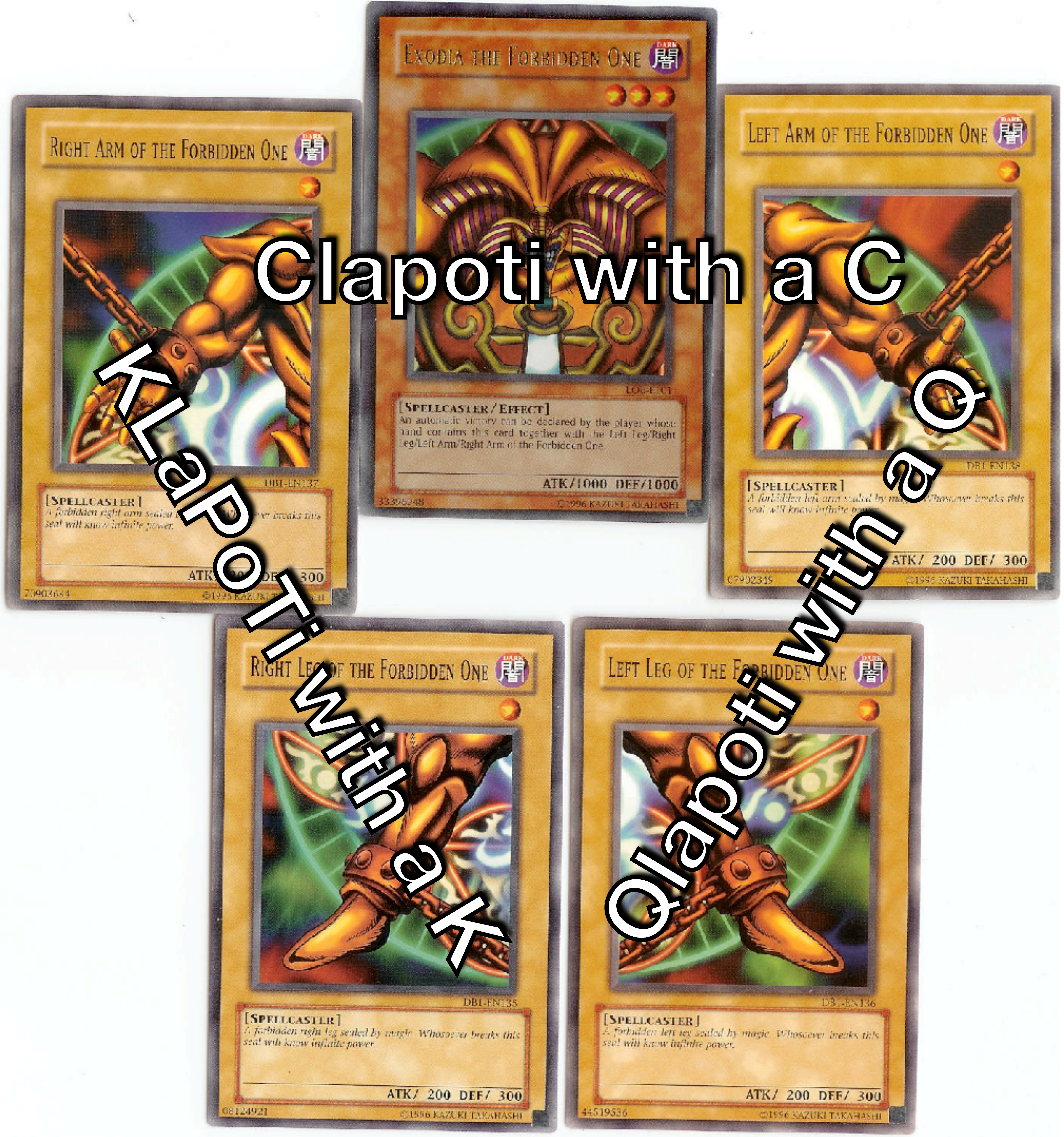
$$N(a_1^2 + a_2^2 + b_1^2 + b_2^2) + (n(\omega - \lambda)/N)(d_1^2 + d_2^2) + 2\lambda(b_1 d_1 + b_2 d_2) = 2^e$$

Decrementing d_2 makes short solutions behave predictably!

In practice, every c **only costs 2 additions mod m to test**

Results

Prime size (bits)	Prime	Variant	Time (s)				Rerand.
			Step 1	Step 2	Step 3	Total	
508	$3 \cdot 11 \cdot 2^{503} - 1$	PEGASIS	0.097	0.48	0.96	1.53	0.17
		qt-P	0.014	0.0014	-	0.97	0
1008	$3 \cdot 5 \cdot 2^{1004} - 1$	PEGASIS	0.21	1.16	2.84	4.21	0.07
		qt-P	0.023	0.0032	-	2.86	0
1554	$3^2 \cdot 2^{1551} - 1$	PEGASIS	1.19	2.85	6.49	10.5	1.53
		qt-P	0.043	0.0084	-	6.54	0
2031	$3 \cdot 17 \cdot 2^{2026} - 1$	PEGASIS	1.68	8.34	11.3	21.3	0.70
		qt-P	0.21	0.018	-	11.5	0
4089	$3^2 \cdot 7 \cdot 2^{4084} - 1$	PEGASIS	15.6	52.8	53.5	122	0.41
		qt-P	1.01	0.082	-	54.6	0



= qt-PEGASIS

Class group actions where essentially the whole cost at all security levels is a single 4-dimensional isogeny!