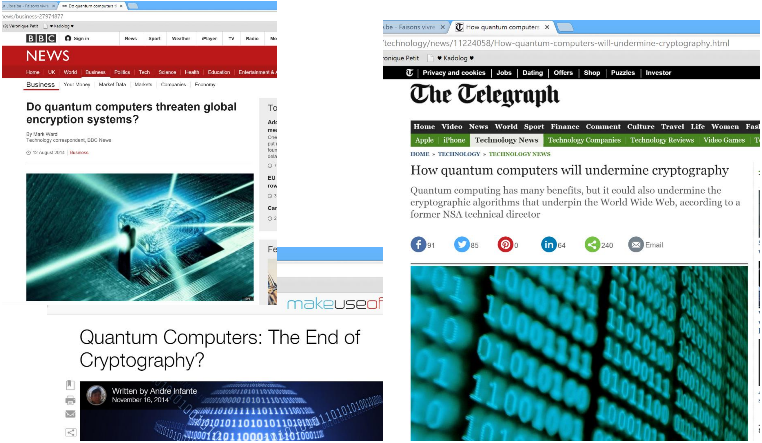
Post-quantum cryptography based on isogeny problems?

Christophe Petit

University of Oxford

The threat of quantum computers



Isogeny Problems

- Recently proposed for post-quantum cryptography
- Classical and quantum algorithms still exponential time
- Some history, e.g. David Kohel's PhD thesis in 1996
- Natural problems from a number theory point of view

Outline

Motivation

Isogenies and Cryptographic Protocols

Hard and Easy Isogeny Problems

Computing Isogenies using Torsion Point Images

Conclusion

Outline

Motivation

Isogenies and Cryptographic Protocols

Hard and Easy Isogeny Problems

Computing Isogenies using Torsion Point Images

Conclusion

Isogenies

- Let p be a prime. Up to isomorphism, any supersingular elliptic curve is defined over $\mathbb{F}_{p^2}$
- An isogeny from a curve E_0 is a morphism $\phi : E_0 \rightarrow E_1$ sending 0 to 0
- In Weierstrass affine coordinates we can write

$$\phi : E_0 \rightarrow E_1 : \phi(x, y) = \left(\frac{\varphi(x)}{\psi^2(x, y)}, \frac{\omega(x, y)}{\psi^3(x, y)} \right)$$

  where ψ^2 only depends on x, and $\omega/\psi^3 = ys(x)/t(x)$
- Isogeny degree is $\deg \phi = \max\{\deg \varphi, \deg \psi^2\}$
- Often we write $E_1 = E_0/G$ where $G = \ker \phi$

Isogeny problems

- Isogeny problems with potential interest for cryptography are about "computing" isogenies between two curves, or some variant of this problem

Isogeny problems

- Isogeny problems with potential interest for cryptography are about "computing" isogenies between two curves, or some variant of this problem
- For these problems to be "hard" these isogenies must have "large" degree
- So representation as a rational map not efficient enough
- Can often assume degree is smooth hence can return isogeny as a composition of low degree isogenies

Isogeny problems

- Isogeny problems with potential interest for cryptography are about "computing" isogenies between two curves, or some variant of this problem
- For these problems to be "hard" these isogenies must have "large" degree
- So representation as a rational map not efficient enough
- Can often assume degree is smooth hence can return isogeny as a composition of low degree isogenies
- Attacker sometimes given extra information on isogenies

Isogeny graphs

- Over $\bar{K}$ the ℓ-torsion $E[\ell]$ (points of order dividing ℓ) is isomorphic to $\mathbb{Z}_\ell \times \mathbb{Z}_\ell$
- There are $\ell + 1$ cyclic subgroups of order ℓ ; each one is the kernel of a degree ℓ isogeny
- ℓ-isogeny graph : each vertex is a j-invariant over $\bar{K}$, each edge corresponds to one degree ℓ isogeny
- Undirected graph : to every $\phi : E_1 \to E_2$ corresponds a dual isogeny $\hat{\phi} : E_2 \to E_1$ with $\phi\hat{\phi} = [\deg \phi]$
- In supersingular case all j and isogenies defined over $\mathbb{F}_{p^2}$ and graphs are Ramanujan (optimal expansion graphs)
- Isogeny problems $\sim$ finding paths in these graphs

Hash function

$$H : \{0,1\}^* \to \{0,1\}^n$$

- **Collision resistance :**
  hard to find m, m' such that $H(m) = H(m')$
- **Preimage resistance :**
  given h, hard to find m such that $H(m) = h$
- **Second preimage resistance :**
  given m, hard to find m' such that $H(m') = h$

- Popular ones use block cipher like compression functions and Merkle-Damgård ; not based on maths problems

Charles-Goren-Lauter hash function

Hash of the Future?

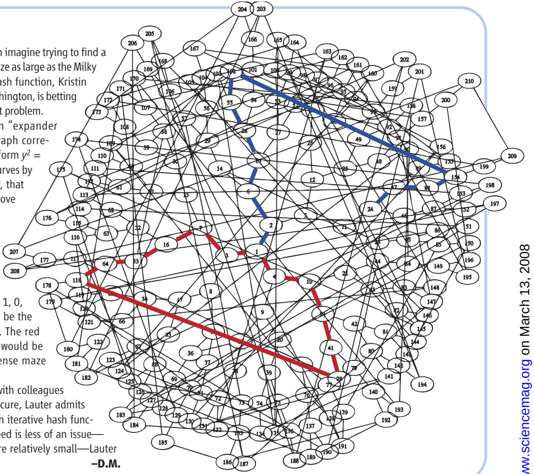
Have you ever struggled to solve a maze? Then imagine trying to find a path through a tangled, three-dimensional maze as large as the Milky Way. By incorporating such a maze into a hash function, Kristin Lauter of Microsoft Research in Redmond, Washington, is betting that neither you nor anyone else will solve that problem.

Technically, Lauter's maze is called an "expander graph" (see figure, right). Nodes in the graph correspond to elliptic curves, or equations of the form $y^2 = x^3 + ax + b$. Each curve leads to three other curves by a mathematical relation, now called isogeny, that Pierre de Fermat discovered while trying to prove his famous Last Theorem.

To hash a digital file using an expander graph, you would convert the bits of data into directions: 0 would mean "turn right," 1 would mean "turn left." In the maze illustrated here, after the initial step 1-2, the blue path encodes the directions 1, 0, 1, 1, 0, 0, 0, 0, 1, ending at point 24, which would be the digital signature of the string 101100001. The red loop shows a collision of two paths, which would be practically impossible to find in the immense maze envisioned by Lauter.

Although her hash function (developed with colleagues Denis Charles and Eyal Goren) is provably secure, Lauter admits that it is not yet fast enough to compete with iterative hash functions. However, for applications in which speed is less of an issue—for example, where the files to be hashed are relatively small—Lauter believes it might be a winner.

–D.M.



Properties

- **Uniform output distribution** for large enough messages
- **Preimage problem for CGL hash function :** Let E_0 and E_1 be two supersingular elliptic curves over $\mathbb{F}_{p^2}$ with $|E_0(\mathbb{F}_{p^2})| = |E_1(\mathbb{F}_{p^2})|$. Find $e \in \mathbb{N}$ and an isogeny of degree ℓ^e from E_0 to E_1.
- **Collision problem for CGL hash function :** Let E_0 be a supersingular elliptic curve over $\mathbb{F}_{p^2}$. Find $e_1, e_2 \in \mathbb{N}$, a supersingular elliptic curve E_1 and two distinct isogenies (i.e. with distinct kernels) of degrees respectively ℓ^{e_1} and ℓ^{e_2} from E_0 to E_1.

Key agreement

- Alice and Bob want to agree on a common secret key
- They only exchange public messages
- Eve can see all messages exchanged, yet she should not be able to infer the secret key

Diffie-Hellman key agreement

- Choose g generating a cyclic group
- Alice picks a random a and sends g^a
- Bob picks a random b and sends g^b
- Alice computes $(g^b)^a = g^{ab}$
- Bob computes $(g^a)^b = g^{ab}$
- Eve cannot compute a, b or g^{ab} from g^a and g^b (discrete logarithm, Diffie-Hellman problems)

Isogeny-based Diffie-Hellman

- Choose a prime p, and $N_A, N_B \in \mathbb{N}$ with $\gcd(N_A, N_B) = 1$
  Choose E_0 a supersingular curve over $\mathbb{F}_{p^2}$
- Alice picks a cyclic subgroup $G_A \subset E_0[N_A]$ defining an isogeny $\phi_A : E_0 \to E_A = E_0/G_A$ and she sends E_A to Bob
- Bob picks a cyclic subgroup $G_B \subset E_0[N_B]$ defining an isogeny $\phi_A : E_0 \to E_B = E_0/G_B$ and he sends E_B to Alice

$$\begin{tikzcd}
 & E_A = E_0/G_A \arrow[dr, "\phi'_B"] & \\
E_0 \arrow[ur, "\phi_A"] \arrow[dr, "\phi_B"'] & & E_0/\langle G_A, G_B\rangle \\
 & E_B = E_0/G_B \arrow[ur, "\phi'_A"] &
\end{tikzcd}$$

- Shared key is $E_0/\langle G_A, G_B\rangle = E_B/\phi_B(G_A) = E_A/\phi_A(G_B)$

Isogeny-based Diffie-Hellman (2)

- To compute the shared key Alice will need $\phi_B(G_A)$. This is achieved as follows :
  - Let $G_A = \langle \alpha_A P_A + \beta_A Q_A \rangle$ where $\langle P_A, Q_A \rangle = E_0[N_A]$ and at least one of α_A, β_A coprime to N_A
  - Bob reveals $\phi_B(P_A)$ and $\phi_B(Q_A)$ in first round
  - Alice computes $\phi_B(G_A) = \langle \alpha_A \phi_B(P_A) + \beta_A \phi_B(Q_A) \rangle$

Isogeny-based Diffie-Hellman (2)

- To compute the shared key Alice will need $\phi_B(G_A)$. This is achieved as follows :
  - Let $G_A = \langle \alpha_A P_A + \beta_A Q_A \rangle$ where $\langle P_A, Q_A \rangle = E_0[N_A]$ and at least one of α_A, β_A coprime to N_A
  - Bob reveals $\phi_B(P_A)$ and $\phi_B(Q_A)$ in first round
  - Alice computes $\phi_B(G_A) = \langle \alpha_A \phi_B(P_A) + \beta_A \phi_B(Q_A) \rangle$
- Can compute ϕ_A efficiently if N_A smooth
- Can represent torsion points efficiently if either
  - $N_A = \prod \ell_i^{e_i}$ with $\ell_i^{e_i}$ bounded
  - $N_A | p - 1$

Supersingular key agreement protocol

$$P_A, Q_A, {\color{blue}R_A} \qquad P_B, Q_B, {\color{red}R_B}$$

$$\begin{tikzcd}
 & E_0/\langle R_A \rangle \arrow[dr, red, "\phi'_B"] & \\
E_0 \arrow[ur, blue, "\phi_A"] \arrow[dr, red, "\phi_B"'] & & E_0/\langle R_A, R_B \rangle \\
 & E_0/\langle R_B \rangle \arrow[ur, blue, "\phi'_A"'] &
\end{tikzcd}$$

$$\phi_A(P_B), \phi_A(Q_B) \qquad {\color{red}\phi_A(R_B)}$$

$$\phi_B(P_A), \phi_B(Q_A) \qquad {\color{blue}\phi_B(R_A)}$$

- Jao-De Feo chose $N_i = \ell_i^{e_i}$ and $p = N_A N_B f + 1$
- A priori safer to use arbitrary primes and $N_i \approx p^2$

Identification protocol / proof of knowledge

- Prover wants to prove knowledge of a secret to Verifier without revealing it (can be used for authentication)

Identification protocol / proof of knowledge

- Prover wants to prove knowledge of a secret to Verifier without revealing it (can be used for authentication)
- Security requirements :
  - Correctness : if Prover knows the secret then Prover can convince Verifier
  - Soundness : if Prover convinces Verifier then Prover must know the secret
  - Zero-knowledge : nothing is leaked about the secret

Jao-De Feo-Plût identification protocol

- Proof of knowledge of an isogeny ϕ between two given curves E_0 and E_1

$$E_0 \xrightarrow{\phi} E_1$$

Jao-De Feo-Plût identification protocol

- Proof of knowledge of an isogeny ϕ between two given curves E_0 and E_1

$$\begin{tikzcd} E_0 \arrow[r, "\phi"] \arrow[d, "\psi"'] & E_1 \arrow[d, "\psi'"] \\ E_2 \arrow[r, "\phi'"] & E_3 \end{tikzcd}$$

- 3-round protocol :
  - Prover commits with E_2 and E_3
  - Verifier challenges Prover with one bit b
  - Prover reveals ψ and ψ' if $b = 0$, and ϕ' if $b = 1$

Public Key Encryption and Signatures

- **Public Key Encryption** $\sim$ digital lock : everybody can lock/encrypt but one needs private key to unlock/decrypt
- Diffie-Hellman-like key exchange protocol leads to ElGamal-like public key encryption

Public Key Encryption and Signatures

- **Public Key Encryption** $\sim$ digital lock : everybody can lock/encrypt but one needs private key to unlock/decrypt
- Diffie-Hellman-like key exchange protocol leads to ElGamal-like public key encryption

- **Digital signatures** are analog to real signatures
- Identification protocols lead to digital signatures using the Fiat-Shamir transform (or any alternative)
- In [Galbraith-P-Silva 2017] we build an alternative identification protocol and signature scheme

Outline

Motivation

Isogenies and Cryptographic Protocols

Hard and Easy Isogeny Problems

Computing Isogenies using Torsion Point Images

Conclusion

Isogeny from kernel

- Given $G = \ker \phi$ can compute ϕ with Vélu's formulae

$$\phi(P) = \left(x_P + \sum_{Q \in G \setminus \{O\}} (x_{P+Q} - x_Q), \quad y_P + \sum_{Q \in G \setminus \{O\}} (y_{P+Q} - y_Q) \right)$$

  using $O(\#G)$ operations

Isogeny from kernel

- Given $G = \ker \phi$ can compute ϕ with Vélu's formulae

$$\phi(P) = \left(x_P + \sum_{Q \in G \setminus \{O\}} (x_{P+Q} - x_Q), \quad y_P + \sum_{Q \in G \setminus \{O\}} (y_{P+Q} - y_Q) \right)$$

  using $O(\#G)$ operations

- If $\#G$ is composite then better to write ϕ as a composition of prime degree isogenies
- If $\#G = \prod \ell_i^{e_i}$ write $G = \prod G_i$ with $\#G_i = \ell_i^{e_i}$

Endomorphism ring computation

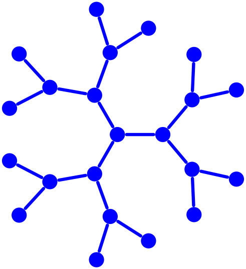
- Given an elliptic curve E defined over a finite field K, compute the endomorphism ring of E

Endomorphism ring computation

- Given an elliptic curve E defined over a finite field K, compute the endomorphism ring of E
- We focus on the supersingular case so $\mathrm{End}(E)$ is a maximal order in the quaternion algebra $B_{p,\infty}$
- Output $=$ some efficient representation of basis elements
- Problem considered by David Kohel in his PhD thesis (Berkeley 1996)

Kohel's algorithm for supersingular curves

- Fix a small ℓ. Given a curve E, compute all its neighbors in isogeny graph. Compute all neighbors of neighbors, etc, until a loop is found, corresponding to an endomorphism



- Complexity $\tilde{O}(\sqrt{p})$

Isogeny computation

- Given elliptic curves E_0, E_1 defined over a finite field K, compute an isogeny $\phi : E_0 \to E_1$

Isogeny computation

- Given elliptic curves E_0, E_1 defined over a finite field K, compute an isogeny $\phi : E_0 \to E_1$
- For the problem to be hard then $\deg \phi$ must be large, so ϕ cannot be returned as a rational map
- Same hardness as endomorphism ring computation, at least heuristically
- May impose some conditions on the degree, for example $\deg \phi = \ell^e$ for some e, with same hardness heuristically
- Can be solved in $O(\sqrt{p})$ with two trees from E_0 and E_1 in the isogeny graph

Deuring correspondence

- Deuring correspondence (1931) : bijection from supersingular curves over $\bar{\mathbb{F}}_p$ (up to Galois conjugacy) to maximal orders in the quaternion algebra $B_{p,\infty}$ (up to conjugation)

$$E \to O \approx \mathrm{End}(E)$$

Deuring correspondence

- Deuring correspondence (1931) : bijection from supersingular curves over $\bar{\mathbb{F}}_p$ (up to Galois conjugacy) to maximal orders in the quaternion algebra $B_{p,\infty}$ (up to conjugation)

$$E \to O \approx \mathrm{End}(E)$$

- Under this correspondence translate isogeny $\varphi : E_1 \to E_2$ into ideal I, both left ideal of O_1 and right ideal of O_2, with degree $\varphi =$ norm of I

Quaternion isogeny computation

- Input : two maximal orders O_0 and O_1 in $B_{p,\infty}$
- Output : a O_0-left ideal $J = Iq$ with ℓ-power norm, where I is a O_0-left ideal and a O_1-right ideal, and $q \in B^*_{p,\infty}$
- Following Deuring's correspondence this corresponds to computing an isogeny $\varphi : E_0 \to E_1$ with power of ℓ degree where $\mathrm{End}(E_0) \approx O_0$ and $\mathrm{End}(E_1) \approx O_1$

Quaternion isogeny computation

- Input : two maximal orders O_0 and O_1 in $B_{p,\infty}$
- Output : a O_0-left ideal $J = Iq$ with ℓ-power norm, where I is a O_0-left ideal and a O_1-right ideal, and $q \in B_{p,\infty}^*$
- Following Deuring's correspondence this corresponds to computing an isogeny $\varphi : E_0 \to E_1$ with power of ℓ degree where $\mathrm{End}(E_0) \approx O_0$ and $\mathrm{End}(E_1) \approx O_1$
- ANTS 2014 heuristic algorithm (Kohel-Lauter-P-Tignol) solves the problem with $e = \log_\ell n(I) \approx \frac{7}{2} \log p$
- Can be adapted to powersmooth norms

Explicit Deuring correspondence

- Given supersingular invariant, return corresponding order

  $=$ Endomorphism ring computation problem

  $\rightarrow$ Believed to be hard

Explicit Deuring correspondence

- Given supersingular invariant, return corresponding order
  - $=$ Endomorphism ring computation problem
  - $\rightarrow$ Believed to be hard
- Given a maximal order, compute corresponding invariant
  - $=$ Inverse endomorphism ring computation problem
  - $\rightarrow$ Heuristic polynomial time algorithm

Explicit Deuring correspondence

- Given supersingular invariant, return corresponding order
  - $=$ Endomorphism ring computation problem
  - $\rightarrow$ Believed to be hard
- Given a maximal order, compute corresponding invariant
  - $=$ Inverse endomorphism ring computation problem
  - $\rightarrow$ Heuristic polynomial time algorithm
- Candidate one-way function !

Special isogeny problems

- In Jao-De Feo-Plût protocols special problems are used
  1. A special prime p is chosen so that $p = N_1 N_2 \pm 1$ with $N_1 \approx N_2 \approx \sqrt{p}$
  2. There are $\approx p/12$ supersingular invariants but only $N_1 \approx \sqrt{p}$ possible choices for E_1
  3. **Extra information provided** : compute $\phi : E_0 \to E_1$ of degree N_1 **knowing** $\phi(P)$ **for all** $P \in E_0[N_2]$
- Point 2 improves tree-based attacks to $O(p^{1/4})$
- We now focus on Point 3

Outline

Motivation

Isogenies and Cryptographic Protocols

Hard and Easy Isogeny Problems

Computing Isogenies using Torsion Point Images

Conclusion

Motivation

- Attack on Jao-De Feo-Plût protocol : compute an isogeny $\phi_1 : E_0 \to E_1$ of degree N_1 **given action of** ϕ_1 **on** $E_0[N_2]$
- How useful is this additional information ?

Motivation

- Attack on Jao-De Feo-Plût protocol : compute an isogeny $\phi_1 : E_0 \to E_1$ of degree N_1 **given action of** ϕ_1 **on** $E_0[N_2]$
- How useful is this additional information ?
  - If $\gcd(N_1, N_2) \neq 1$ can recover (part of) ϕ_1

Motivation

- Attack on Jao-De Feo-Plût protocol : compute an isogeny $\phi_1 : E_0 \to E_1$ of degree N_1 **given action of ϕ_1 on $E_0[N_2]$**
- How useful is this additional information ?
  - If $\gcd(N_1, N_2) \neq 1$ can recover (part of) ϕ_1
  - Active attacks : replace $\phi_1(P_2), \phi_1(Q_2)$ by well-chosen points so that (part of) the secret is leaked in shared key [Galbraith-P-Shani-Ti 2016 + others]

Motivation

- Attack on Jao-De Feo-Plût protocol : compute an isogeny $\phi_1 : E_0 \to E_1$ of degree N_1 **given action of ϕ_1 on $E_0[N_2]$**
- How useful is this additional information ?
  - If $\gcd(N_1, N_2) \neq 1$ can recover (part of) ϕ_1
  - Active attacks : replace $\phi_1(P_2), \phi_1(Q_2)$ by well-chosen points so that (part of) the secret is leaked in shared key [Galbraith-P-Shani-Ti 2016 + others]
  - What about passive attacks (eavesdropping only) ?

Warm-up : computing endomorphisms with auxilliary information

- Let p be a prime and let E be a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$. Let ϕ be a non scalar endomorphism of E with smooth order N_1. Let N_2 be a smooth integer with $\gcd(N_1, N_2) = 1$, and let P, Q be a basis of $E[N_2]$.
- Let R be a subring of $\mathrm{End}(E)$ that is either easy to compute, or given (for example, scalar multiplications).
- Given E, P, Q, $\phi(P)$, $\phi(Q)$, $\deg \phi$, R, compute ϕ.

Warm-up : computing endomorphisms with auxilliary information

- Let p be a prime and let E be a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$. Let ϕ be a non scalar endomorphism of E with smooth order N_1. Let N_2 be a smooth integer with $\gcd(N_1, N_2) = 1$, and let P, Q be a basis of $E[N_2]$.
- Let R be a subring of End(E) that is either easy to compute, or given (for example, scalar multiplications).
- Given E, P, Q, $\phi(P)$, $\phi(Q)$, $\deg \phi$, R, compute ϕ.

- Best previous algorithm : meet-in-the-middle in $\tilde{O}(\sqrt{N_1})$

Algorithm sketch (with $R = \mathbb{Z}$)

- We know ϕ on the N_2 torsion.
  Deduce $\hat{\phi}$ on the N_2 torsion and $\mathrm{Tr}(\phi)$ if $N_2 > 2\sqrt{N_1}$.

Algorithm sketch (with $R = \mathbb{Z}$)

- We know ϕ on the N_2 torsion.
  Deduce $\hat{\phi}$ on the N_2 torsion and $\mathrm{Tr}(\phi)$ if $N_2 > 2\sqrt{N_1}$.
- Consider $\psi := a\phi + b$ for $a, b \in \mathbb{Z}$.
  Can evaluate ψ on the N_2 torsion.

Algorithm sketch (with $R = \mathbb{Z}$)

- We know ϕ on the N_2 torsion.
  Deduce $\hat{\phi}$ on the N_2 torsion and $\mathrm{Tr}(\phi)$ if $N_2 > 2\sqrt{N_1}$.
- Consider $\psi := a\phi + b$ for $a, b \in \mathbb{Z}$.
  Can evaluate ψ on the N_2 torsion.
- Find $a, b \in \mathbb{Z}$ such that

$$\deg \psi = a^2 \deg \phi + b^2 + ab\mathrm{Tr}\phi = N_2 N_1'$$

with N_1' small and smooth. Write $\psi = \psi_{N_1'} \psi_{N_2}$.

Algorithm sketch (with $R = \mathbb{Z}$)

- We know ϕ on the N_2 torsion.
  Deduce $\hat{\phi}$ on the N_2 torsion and $\mathrm{Tr}(\phi)$ if $N_2 > 2\sqrt{N_1}$.
- Consider $\psi := a\phi + b$ for $a, b \in \mathbb{Z}$.
  Can evaluate ψ on the N_2 torsion.
- Find $a, b \in \mathbb{Z}$ such that

$$\deg \psi = a^2 \deg \phi + b^2 + ab\mathrm{Tr}\phi = N_2 N_1'$$

  with N_1' small and smooth. Write $\psi = \psi_{N_1'} \psi_{N_2}$.
- Identify $\ker \psi_{N_2}$ from $\psi(E[N_2])$ and deduce ψ_{N_2}.
- Find $\psi_{N_1'}$ with a meet-in-the-middle strategy.

Algorithm sketch (with $R = \mathbb{Z}$)

- We know ϕ on the N_2 torsion.
  Deduce $\hat{\phi}$ on the N_2 torsion and $\mathrm{Tr}(\phi)$ if $N_2 > 2\sqrt{N_1}$.
- Consider $\psi := a\phi + b$ for $a, b \in \mathbb{Z}$.
  Can evaluate ψ on the N_2 torsion.
- Find $a, b \in \mathbb{Z}$ such that

$$\deg \psi = a^2 \deg \phi + b^2 + ab\mathrm{Tr}\phi = N_2 N_1'$$

  with N_1' small and smooth. Write $\psi = \psi_{N_1'} \psi_{N_2}$.
- Identify $\ker \psi_{N_2}$ from $\psi(E[N_2])$ and deduce ψ_{N_2}.
- Find $\psi_{N_1'}$ with a meet-in-the-middle strategy.
- Find $\ker \phi$ by evaluating $(\psi - b)/a$ on the N_1 torsion, and deduce ϕ.

Finding (a, b) and Complexity

- We have $\deg\psi = a^2 \deg\phi + b^2 + ab\mathrm{Tr}\phi$
  $= \left(b + a\frac{\mathrm{Tr}\phi}{2}\right)^2 + a^2\left(\deg\phi - \left(\frac{\mathrm{Tr}\phi}{2}\right)^2\right)$
- We want $\deg\psi = N_2 N_1'$ and N_1' small and smooth

Finding (a, b) and Complexity

- We have $\deg \psi = a^2 \deg \phi + b^2 + ab\mathrm{Tr}\phi$
  $$= \left(b + a\frac{\mathrm{Tr}\phi}{2}\right)^2 + a^2 \left(\deg \phi - \left(\frac{\mathrm{Tr}\phi}{2}\right)^2\right)$$
- We want $\deg \psi = N_2 N_1'$ and N_1' small and smooth
- Solutions to $\deg \psi = 0 \bmod N_2$ form a dimension 2 lattice
- We compute a reduced basis, then search for a small linear combination of short vectors until N_1' smooth

Finding (a, b) and Complexity

- We have $\deg\psi = a^2 \deg\phi + b^2 + ab\mathrm{Tr}\phi$
  $$= \left(b + a\tfrac{\mathrm{Tr}\phi}{2}\right)^2 + a^2\left(\deg\phi - \left(\tfrac{\mathrm{Tr}\phi}{2}\right)^2\right)$$
- We want $\deg\psi = N_2 N_1'$ and N_1' small and smooth
- Solutions to $\deg\psi = 0 \bmod N_2$ form a dimension 2 lattice
- We compute a reduced basis, then search for a small linear combination of short vectors until N_1' smooth
- Heuristic analysis shows we can expect $N_1' \approx \sqrt{N_1}$. Revealing $\phi(E[N_2])$ leads to a near square root speedup. (Some parameter restrictions apply.)

Open problem : subfield curves

- If E is defined over $\mathbb{F}_p$ we can take $R = \mathbb{Z}[\pi]$
- Let $\phi' = \phi - \mathrm{Tr}\phi$ and consider

$$\psi = (a\phi' + b)\pi_p + c\phi' + d$$

- Let $\Delta = \deg\phi - \left(\frac{\mathrm{Tr}\phi}{2}\right)^2$. We want

$$\deg\psi = (a^2\Delta + b^2)p + (c^2\Delta + d^2) + (ad - bc)\mathrm{Tr}(\phi'\pi_p) = N_1' N_2$$

with N_1' small and smooth

Open problem : subfield curves

- If E is defined over $\mathbb{F}_p$ we can take $R = \mathbb{Z}[\pi]$
- Let $\phi' = \phi - \mathrm{Tr}\phi$ and consider

$$\psi = (a\phi' + b)\pi_p + c\phi' + d$$

- Let $\Delta = \deg\phi - \left(\frac{\mathrm{Tr}\phi}{2}\right)^2$. We want

$$\deg\psi = (a^2\Delta + b^2)p + (c^2\Delta + d^2) + (ad - bc)\mathrm{Tr}(\phi'\pi_p) = N_1' N_2$$

with N_1' small and smooth

- Heuristic analysis : when $N_2 \approx N_1 p$ we should be able to get $N_1' = O(1)$,

Open problem : subfield curves

- If E is defined over $\mathbb{F}_p$ we can take $R = \mathbb{Z}[\pi]$
- Let $\phi' = \phi - \mathrm{Tr}\phi$ and consider

$$\psi = (a\phi' + b)\pi_p + c\phi' + d$$

- Let $\Delta = \deg\phi - \left(\frac{\mathrm{Tr}\phi}{2}\right)^2$. We want

$$\deg\psi = (a^2\Delta+b^2)p+(c^2\Delta+d^2)+(ad-bc)\mathrm{Tr}(\phi'\pi_p) = N_1'N_2$$

  with N_1' small and smooth
- Heuristic analysis : when $N_2 \approx N_1 p$ we should be able to get $N_1' = O(1)$, but I cannot solve the above equation

Computing isogenies with auxilliary information

- Let p be a prime. Let $N_1, N_2 \in \mathbb{Z}$ coprime. Let E_0 be a supersingular elliptic curve over $\mathbb{F}_{p^2}$. Let $\phi_1 : E_0 \to E_1$ be an isogeny of degree N_1.
- Let R_0, R_1 be subrings of $\mathrm{End}(E_0)$, $\mathrm{End}(E_1)$ respectively. Assume R_0 contains more than scalar multiplications.
- Given N_1, E_1, R_0, R_1 and the image of ϕ_1 on the whole N_2 torsion, compute ϕ_1.

Computing isogenies with auxilliary information

- Let p be a prime. Let $N_1, N_2 \in \mathbb{Z}$ coprime. Let E_0 be a supersingular elliptic curve over $\mathbb{F}_{p^2}$. Let $\phi_1 : E_0 \to E_1$ be an isogeny of degree N_1.
- Let R_0, R_1 be subrings of $\mathrm{End}(E_0)$, $\mathrm{End}(E_1)$ respectively. Assume R_0 contains more than scalar multiplications.
- Given N_1, E_1, R_0, R_1 and the image of ϕ_1 on the whole N_2 torsion, compute ϕ_1.

- Best previous algorithm : meet-in-the-middle in $\tilde{O}(\sqrt{N_1})$

General idea

- For $\theta \in \mathrm{End}(E_0)$ consider $\phi = \phi_1 \theta \hat{\phi}_1 \in \mathrm{End}(E_1)$
- Evaluate ϕ on the N_2 torsion
- Apply techniques from above on ϕ
- Compute $\ker \hat{\phi}_1 = \ker \phi \cap E_1[N_1]$
- Deduce $\hat{\phi}_1$ and ϕ_1

Remarks

- Several authors have suggested to use $j(E_0) = 1728$ for efficiency reasons. In this case $\mathrm{End}(E_0)$ is entirely known and moreover it contains a degree 1 non scalar element. Both aspects are useful in attacks.
- The paper develops two attacks but we expect variants and improvements to come.

Impact on Key Agreement Protocol

- For $j(E_0) = 1728$ and when $N_1 \approx p^2$ and $N_2 \approx N_1^4$ this approach leads to polynomial time key recovery (heuristic analysis)
- Assuming only that $\mathrm{End}(E_0)$ has a small element, then if $\log N_2 \approx (\log^2 N_1)$, a variant of the above strategy also leads to polynomial time key recovery (heuristic analysis)
- Parameters suggested by De Feo-Jao-Plût $N_1 \approx N_2 \approx \sqrt{p}$ are not affected so far

Outline

Motivation

Isogenies and Cryptographic Protocols

Hard and Easy Isogeny Problems

Computing Isogenies using Torsion Point Images

Conclusion

Conclusion

- Revealing images of torsion points helps the resolution of (at least some) isogeny problems
- Endomorphism ring computation & pure isogeny problems are natural problems with some history but
  - More classical and quantum cryptanalysis needed
  - Beware of variants
- We can build some crypto protocols on isogeny problems (key exchange, public key encryption, signatures) with reasonable efficiency. Other protocols ?

Thanks !

- Questions ?