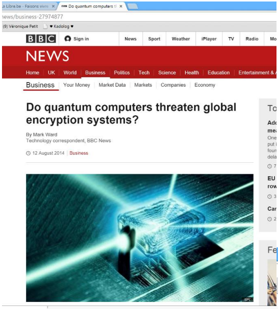
Post-quantum cryptography based on isogeny problems?

Christophe Petit

Université libre de Bruxelles
University of Birmingham

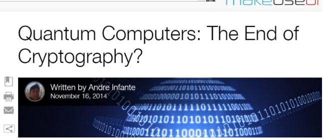
The threat of quantum computers



Do quantum computers threaten global encryption systems?

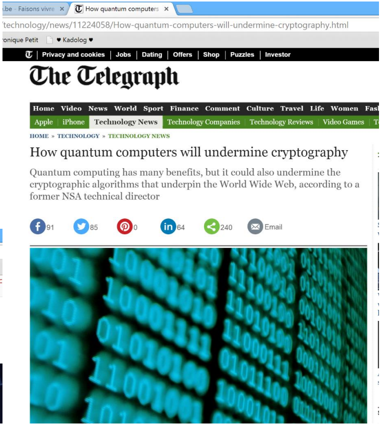
By Mark Ward
Technology correspondent, BBC News

12 August 2014 | Business



Quantum Computers: The End of Cryptography?

Written by Andre Infante
November 16, 2014


How quantum computers
technology/news/11224058/How-quantum-computers-will-undermine-cryptography.html
Privacy and cookies | Jobs | Dating | Offers | Shop | Puzzles | Investor
The Telegraph
Home Video News World Sport Finance Comment Culture Travel Life Women
Apple | iPhone | Technology News | Technology Companies | Technology Reviews | Video Games
HOME » TECHNOLOGY » TECHNOLOGY NEWS
91
85
0
64
240
Email

How quantum computers will undermine cryptography

Quantum computing has many benefits, but it could also undermine the cryptographic algorithms that underpin the World Wide Web, according to a former NSA technical director

Isogeny Problems

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

Outline

Motivation

Isogeny-based Cryptography

Hard and Easy Isogeny Problems

Computing Isogenies using Torsion Point Images

Conclusion

Outline

Motivation

Isogeny-based Cryptography

Hard and Easy Isogeny Problems

Computing Isogenies using Torsion Point Images

Conclusion

Isogenies

- Let K be a finite field and E_0 be an elliptic curve over K
- An isogeny from a curve E_0 is a non trivial morphism $\phi : E_0 \to E_1$ sending 0 to 0
- In Weierstrass affine coordinates we can write

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

- 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

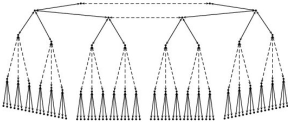
- Isogeny problems with potential interest for cryptography are about “computing” isogenies between two curves, or some variant of this problem
- For these problems to be “hard” these isogenies must have “large” degree
- So representation as a rational map not efficient enough
- Can often assume degree is smooth hence isogeny can be returned 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 $\phi\hat{\phi} = [\deg \phi]$
- Isogeny problems $\sim$ finding paths in these graphs

Isogeny graph structure

- In supersingular case all j and isogenies defined over $\mathbb{F}_{p^2}$ and graphs are Ramanujan (optimal expansion graphs)
- In ordinary case, isogeny graphs have “volcano” structure



Picture credit : Josep Miret

- In some contexts supersingular isogeny graphs restricted to curves and isogenies defined over $\mathbb{F}_p$ (e.g. CSIDH). The restriction then also has a volcano structure, with at most two levels

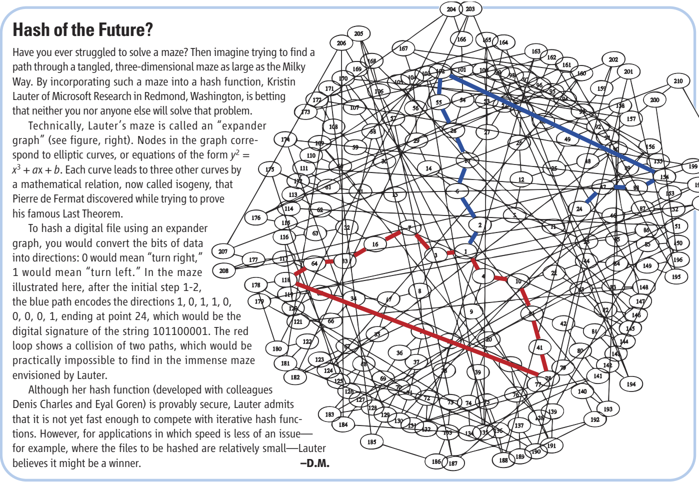
Hash function

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

- **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 $H(m') = h$

- Used in cryptography for integrity, authentication, ...

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.

www.sciencemag.org on March 13, 2008

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)

Diffie-Hellman from abelian group actions [C06,S10]

- Let $\tau : G \times S \to S : (g, s) \to \tau_g(s)$ be a group action, where G is an abelian group
- Choose $s_0 \in S$
- Alice picks a random $a \in G$ and sends $s_a := \tau_a(s_0)$
- Bob picks a random $b \in G$ and sends $s_b := \tau_b(s_0)$
- Alice computes $\tau_a(s_b) = \tau_{ab}(s_0)$
- Bob computes $\tau_b(s_a) = \tau_{ab}(s_0)$

- Examples : standard DH, CSIDH

Commutative Supersingular Isogeny Diffie-Hellman (CSIDH)

- Choose a prime p
- Let $S = \{\text{supersingular curves defined over } \mathbb{F}_p\}$ (up to isomorphisms, so in fact a set of j-invariants)
- For $E \in S$ let $\mathrm{End}_{\mathbb{F}_p}(E)$ be the set of endomorphisms defined over $\mathbb{F}_p$
- Then $\mathrm{End}_{\mathbb{F}_p}(E) \approx \mathbb{Z}[\sqrt{-p}]$ or $\mathrm{End}_{\mathbb{F}_p}(E) \approx \mathbb{Z}[\frac{\sqrt{-p}+1}{2}]$ (restrict to either case to get either CSIDH or CSURF)
- Let G be the class group of $\mathrm{End}_{\mathbb{F}_p}(E)$
- Efficiency : choose p so that all computations over $\mathbb{F}_{p^2}$

Supersingular Isogeny Diffie-Hellman (SIDH)

- 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 $\phi_B : E_0 \to E_B = E_0/G_B$ 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 $E_0/\langle G_A, G_B\rangle = E_B/\phi_B(G_A) = E_A/\phi_A(G_B)$

Supersingular Isogeny 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 addition to E_B
    - Alice computes $\phi_B(G_A) = \langle \alpha_A \phi_B(P_A) + \beta_A \phi_B(Q_A) \rangle$
- Can represent ϕ_A efficiently if N_A smooth
- Can represent torsion points efficiently if either
    - $N_A \mid p - 1$
    - $N_A = \prod \ell_i^{e_i}$ with $\ell_i^{e_i}$ small

Supersingular Isogeny Diffie-Hellman (3)

$$\begin{tikzcd}
 & & \begin{matrix} E_0/\langle R_A \rangle \\ \phi_A(P_B), \phi_A(Q_B) \\ \phi_A(R_B) \end{matrix} \arrow[dr, "\phi'_B"] & \\
\begin{matrix} P_A, Q_A, R_A \\ P_B, Q_B, R_B \end{matrix} & E_0 \arrow[ur, "\phi_A"] \arrow[dr, "\phi_B"'] & & E_0/\langle R_A, R_B \rangle \\
 & & \begin{matrix} E_0/\langle R_B \rangle \\ \phi_B(P_A), \phi_B(Q_A) \\ \phi_B(R_A) \end{matrix} \arrow[ur, "\phi'_A"'] &
\end{tikzcd}$$

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

Public Key Encryption protocols

- Diffie-Hellman-like key exchange protocol leads to ElGamal-like public key encryption
- Séta : alternative encryption scheme based on a new trapdoor mechanism [DKPS19] (see later)
- SiGamal, C-SiGamal : variants of CSIDH, trading random oracle for new complexity assumption [MOT20]

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

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

- 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 $b = 1$

Digital Signatures

- Identification protocols lead to digital signatures using the Fiat-Shamir transform (or any alternative)
- In [GPV17] we build an alternative identification protocol and signature scheme (see later)
- CSIDH versions : SeaSign [FG18], CSIFish [BKV19]
- SQISign [FKLPW20] : drastic improvements of [GPV17]
  sign 204B, sk 16B, pk 64B, keygen 0.6s, sign 2.5s, verif 50ms

And more !

- Undeniable signatures [JS14]
- Oblivious transfer [DOPS18, V18, BNOB18]
- Verifiable delay functions [FMPS19]
- Trapdoor DDH groups [KPS20]
- See also [AFMP20] at Asiacrypt 2020

Outline

Motivation

Isogeny-based Cryptography

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

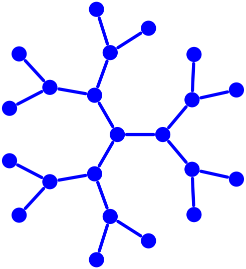
- If $\#G$ is composite then better to write ϕ as a composition of prime degree isogenies
- If $\#G = \prod \ell_i^{e_i}$ write $G = \prod G_i$ with $\#G_i = \ell_i^{e_i}$
- Prime degree isogenies : $O(\ell)$ seems a lower bound, but codomain and point image only cost $O(\sqrt{\ell})$ operations [BFLS20]

Endomorphism ring computation

- Given an elliptic curve E defined over a finite field K, compute the endomorphism ring of E
- We know $\text{End}(E)$ is
  - A maximal order in the quaternion algebra $B_{p,\infty}$ (supersingular curves)
  - An order in a quadratic imaginary number field (ordinary curves)
- Output $=$ some efficient representation of basis elements (several natural representations are equivalent [PL17])
- 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 $\tilde{O}(\sqrt{p})$

Ordinary case and CSIDH

- Ordinary case : subexponential time (Bisson-Sutherland)

- CSIDH [CLMPR18] and variants : we have

$$\mathbb{Z}[\pi] \subseteq \mathrm{End}(E) \subseteq \mathbb{Z}\left[\frac{\pi+1}{2}\right]$$

  where $\pi : (x, y) \to (x^p, y^p)$,

  CSIDH parameters such that $\mathrm{End}(E) = \mathbb{Z}[\pi]$

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

Isogeny computation (2)

- Supersingular case
  - 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 $\tilde{O}(\sqrt{p})$ with two trees from E_0 and E_1 in the isogeny graph

- Ordinary and CSIDH cases : subexponential *quantum* algorithm

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)

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

- Under this correspondence, translate isogeny $\varphi : E_1 \rightarrow E_2$ 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 $\mathrm{End}(E_1) \approx O_1$
- 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
- Can be improved to $e \approx 3 \log p$ [PS18]
- New algorithms for any maximal order O_0 in [FKLPW20]

Remark : GPS signatures [GPS17]

- Fiat-Shamir transform on (parallel instances of) the following identification protocol

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

where secret is φ, commitment is E_1, challenge is ψ, response η is obtained by applying [KLPT14] to $\psi \circ \varphi$

Remark : SQISign [FKLPW20]

- Fiat-Shamir transform on (a single instance of) the following identification protocol

$$\begin{tikzcd} & E_0 \arrow[r, "\psi"] \arrow[dl, dashed, "\tau"'] & E_1 \arrow[d, "\varphi"] \\ E_A \arrow[rr, "\sigma"] & & E_2 \end{tikzcd}$$

where secret is τ ; commitment is E_1 ; challenge is φ ; response σ computed with a new KLPT generalization

Explicit Deuring correspondence

- Given supersingular invariant, return corresponding order

  $\approx$ Endomorphism ring computation problem

  $\rightarrow$ Believed to be hard

- Given a maximal order, compute corresponding invariant

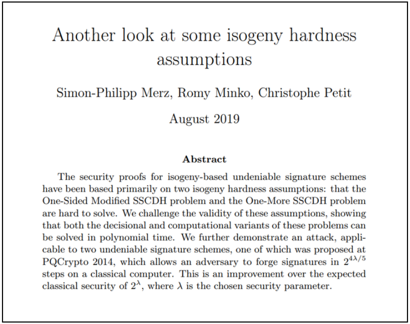
  $\approx$ Inverse endomorphism ring computation problem

  $\rightarrow$ Heuristic polynomial time algorithm [PL17]

- Candidate one-way function

Special isogeny problems (1)

- While above problems are natural and have some history, many protocols rely on “variants” for security
- Not all of these variants are hard. . .



Another look at some isogeny hardness assumptions

Simon-Philipp Merz, Romy Minko, Christophe Petit

August 2019

Abstract

The security proofs for isogeny-based undeniable signature schemes have been based primarily on two isogeny hardness assumptions: that the One-Sided Modified SSCDH problem and the One-More SSCDH problem are hard to solve. We challenge the validity of these assumptions, showing that both the decisional and computational variants of these problems can be solved in polynomial time. We further demonstrate an attack, applicable to two undeniable signature schemes, one of which was proposed at PQCrypto 2014, which allows an adversary to forge signatures in $2^{4\lambda/5}$ steps on a classical computer. This is an improvement over the expected classical security of 2^{λ}, where λ is the chosen security parameter.

Special isogeny problems (2)

- 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

Isogeny-based Cryptography

Hard and Easy Isogeny Problems

Computing Isogenies using Torsion Point Images

- Building endomorphisms of E_0
- A quantum hidden shift attack

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 ?
  - 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]

- Next slides : passive attacks (eavesdropping only)
  - Building endomorphisms of E_0 [P17,KMPPS20]
  - Quantum hidden shift attack [KMPW20]

Outline

Motivation

Isogeny-based Cryptography

Hard and Easy Isogeny Problems

Computing Isogenies using Torsion Point Images

- Building endomorphisms of E_0
- A quantum hidden shift attack

Conclusion

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 $N_2 > 2\sqrt{N_1}$.
- 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 $\deg\psi = a^2\deg\phi + b^2 + ab\mathrm{Tr}(\phi)$
  $$= \left(b + a\tfrac{\mathrm{Tr}\phi}{2}\right)^2 + a^2\left(\deg\phi - \left(\tfrac{\mathrm{Tr}\phi}{2}\right)^2\right)$$
- We want $\deg\psi = N_2N_1'$ 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.)

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

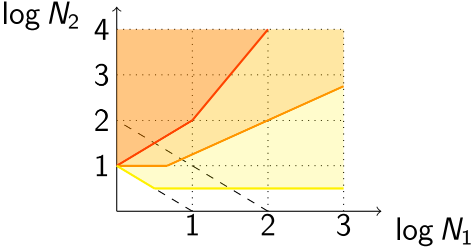
- For $\theta \in \mathrm{End}(E_0)$ consider $\phi = \phi_1\theta\hat{\phi}_1 \in \mathrm{End}(E_1)$
- Evaluate ϕ on the N_2 torsion
- Apply techniques from above on ϕ
- Compute $\ker\phi \cap E_1[N_1]$
- Deduce $\ker\hat{\phi}_1$, then $\hat{\phi}_1$ and ϕ_1

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 θ. Both aspects are useful in attacks.

Impact on SIDH / SIKE

- SIKE parameters : $N_1 \approx N_2 \approx \sqrt{p}$
- First attacks in [P17] only for significantly overstreched parameters, such as $N_1 \approx p^2$ and $N_2 \approx N_1^4$
- New attacks in [KMPPS20] get closer to actual parameters ; apply to "BSIDH-like parameters"



Impact on SIDH / SIKE (2)

- "Weak instances" [KMPPS20] :
  - Can construct an alternative initial curve E_0 together with additional information allowing to break SIDH instances faster than with previous approaches
  - Special choices of parameters also lead to better attacks

Remark : Séta [DKPS19]

- Key idea : use torsion point attacks as a trapdoor mechanism, noticing that they only apply when the endomorphism ring of E_0 is known
- Key generation : secret key is random walk $E_0 \to E_s$, public key is E_s

$$\begin{tikzcd} E_0 \arrow[r, "\varphi_s"] & E_s \arrow[r, "\varphi_m"] & E_m \end{tikzcd}$$

- Encryption : take a walk from E_s depending on message. Ciphertext is final curve E_m + images of torsion points
- Decryption : use torsion point attack to recover isogeny hence message

Outline

Motivation

Isogeny-based Cryptography

Hard and Easy Isogeny Problems

Computing Isogenies using Torsion Point Images

- Building endomorphisms of E_0
- A quantum hidden shift attack

Conclusion

A quote from Jao-De Feo-Plût

- Childs-Jao-Soukharev : quantum subexponential attack for computing isogenies between **ordinary** curves, by reduction to hidden shift problem
- SIDH uses **supersingular** curves. Jao-De Feo-Plût :

  "Since the algorithm of Childs et al. depends crucially on the properties of abelian groups, we believe that no reasonable variant of this strategy would apply to supersingular curves."

Malleability oracles and one-way function

- Consider a **one-way function** $f : I \rightarrow O$
- Assume a group G acts on I
- Assume **malleability oracle** : given $f(i)$ and $g \in G$, can compute $f(g \cdot i)$ efficiently
- Then (assuming f is injective, G is finite abelian, and action is transitive and free) the function f **can be inverted in quantum subexponential time**
  - Choose $i_o \in I$
  - Define $F_0(g) = f(g \cdot i_0)$
  - Define $F_1(g) = f(g \cdot i)$
  - Find h such that $F_1(g) = F_0(hg)$
  - Compute $i = h \cdot i_0$

Application to SIDH-like protocols (1)

- Let E_0 with known endomorphism ring. Let N_1, N_2 be SIDH parameters.
- Let I be the set of cyclic subgroups of order N_1 in E_0
- Consider one-way function f sending C any cyclic subgroup of order N_1 to $E_A = E/C$
- Choose $\theta_0 \in \mathrm{End}(E_0)$ with $\deg \theta_0$ coprime to N_1, and let G be the subgroup of endomorphisms generated by θ_0 (modulo N_1, and up to scalar multiplications)
- Note that G acts on I

Application to SIDH-like protocols (2)

- To apply our framework we need to construct a malleability oracle, i.e. an efficient algorithm that given E/C and $\theta \in G$ computes $E/\theta(C)$
- Torsion point images immediately help if $\deg\theta$ divides N_2

$$\begin{tikzcd}
E_0 \arrow[rr, "\varphi"] \arrow[d, "\theta"'] & & E_A \arrow[d] \\
E_0 \arrow[r] & E_0/\theta(\ker\varphi) \arrow[r, phantom, "\cong"] & E_A/\varphi(\ker\theta)
\end{tikzcd}$$

- In general : for any $\theta = \theta_0^i \in G$, find $\theta' \in \mathrm{End}(E_0)$ with $\deg\theta' | N_2$ such that $\theta = \theta'$ on the N_1 torsion

Application to SIDH-like protocols (3)

- In general : for any $\theta = \theta_0^i \in G$, find $\theta' \in \mathrm{End}(E_0)$ with $\deg \theta' | N_2$ such that $\theta = \theta'$ on the N_1 torsion
  - This amounts to a lifting problem alike one solved in KLPT algorithm
  - Can only be done for large enough N_2
- Choice of θ, proof that group action is free and transitive, lifting algorithm + extra technicalities : see [KMPW20]
- **Quantum hidden shift attack** when roughly $N_2 > pN_1^3$

Outline

Motivation

Isogeny-based Cryptography

Hard and Easy Isogeny Problems

Computing Isogenies using Torsion Point Images

Conclusion

Conclusion

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

Thanks !

- Questions ?