About these slides

- Temporary version of the slides
- Please check the course website for updates
- Please report any typo and error found !

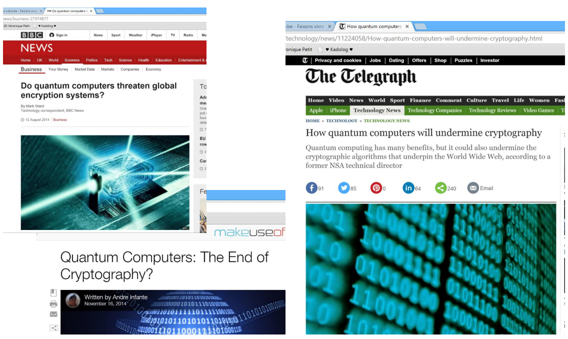
Advanced Cryptography

Isogeny-based Cryptography

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

Isogeny Problems

Existing Cryptographic Protocols

Cryptanalysis Results

Conclusion

Outline

Motivation

Isogeny Problems

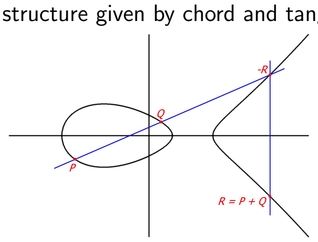
Existing Cryptographic Protocols

Cryptanalysis Results

Conclusion

Elliptic curves

- Set of rational points satisfying some cubic equation
- Group structure given by chord and tangent rule



Elliptic curve discrete logarithm problem (ECDLP)

- Given an elliptic curve E over a finite field K,
  Given $P \in E(K)$, given $Q \in G :=< P >$,
  Find $x \in \mathbb{Z}$ such that $Q = xP$.
- Underlies strongest cryptosystems today
  Elliptic Curve Diffie-Hellman, ECDSA, ...
- Best solvers are generic DLP algorithms in general
- But : easily broken with a quantum computer

Isogenies

- Rational maps from one curve to another

$$\phi : E_0 \to E_1 : (x, y) \to \phi(x, y)$$

- Group homomorphisms

$$\phi(P + Q) = \phi(P) + \phi(Q)$$

- If $E_1 = E_0$ we say ϕ is an endomorphism of E_0
  - Examples : scalar multiplications, Frobenius

Isogenies

- In fact we can write

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

where ψ^2 only depends on x, and $\omega/\psi^3 = ys(x)/t(x)$
- $\deg \phi = \max\{\deg \varphi, \deg \psi^2\}$
- Kernel $\ker \phi = \{P \in E_0 : \phi(P) = O\}$
  - $(x, y) \in \ker \phi \setminus \{O\} \Leftrightarrow \psi(x, y) = 0$
  - $G = \ker \phi$ is a cyclic subgroup of $E_0[\deg \phi]$
  - Often we write $E_1 = E_0/G$
  - For separable isogenies $\deg \phi = \# \ker \phi$

First computational aspects

- 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
- Often the isogeny required has large (exponential) degree, so need some non trivial representation
  - If $\deg \phi = n_1 n_2$ then $\phi = \phi_1 \circ \phi_2$ with $n_i = \deg \phi_i$

Structure of the endomorphism ring

- Ring structure : if ϕ_1, ϕ_2 are endomorphisms of E then so are $\phi_1 + \phi_2$ and $\phi_1 \circ \phi_2$
- Ordinary curves : order in a quadratic imaginary field K
  - $K = \mathbb{Q}(\pi)$ with $\pi^2 + t\pi + p = 0$ where $\Delta = t^2 - 4p < 0$
  - Contains scalar multiplications and the Frobenius π
- Supersingular curves : maximal order in the quaternion algebra $B_{p,\infty}$ ramified at p (characteristic of K) and $\mathbb{R}$
  - $B_{p,\infty} = \mathbb{Q}(i, j)$ with $i^2 = -q$, $j^2 = -p$, $k = ij = -ji$
  - q prime and under GRH we can take $q = O(\log p)$.
  - Contains scalar multiplications, the Frobenius π and a third element ϕ such that $\phi\pi \neq \pi\phi$

Endomorphism ring computation

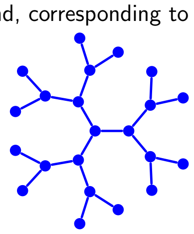
- Ring structure : if ϕ_1, ϕ_2 are endomorphisms of E then so are $\phi_1 + \phi_2$ and $\phi_1 \circ \phi_2$
- **Endomorphism ring computation :** Given an elliptic curve E defined over a finite field K, compute the endomorphism ring of E
- Output = some efficient representation of basis elements
- Problem considered by David Kohel in his PhD thesis (Berkeley 1996)
- Explicit version of Deuring correspondence (1931)

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 corresponding to one 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)

Kohel's algorithm for supersingular curves

- From now on only supersingular curves, defined over $\mathbb{F}_{p^2}$
- Fix a small ℓ. Given a curve E, compute all its neighbors in the graph. Compute all neighbors of neighbors, etc, until a loop is found, corresponding to an endomorphism



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

Isogeny computation

- **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 couple of rational maps
- Same hardness as endomorphism ring computation, at least heuristically (see later)
- 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

Special isogeny problems

- In Jao-de Feo-Plût protocols special problems are used
  1. A special prime p is chosen so that $p = 2^{e_2}3^{e_3}f \pm 1$ with $2^{e_2} \approx 3^{e_3} \approx \sqrt{p}$
  2. There exists an isogeny of degree $O(\sqrt{p})$ power of 2/3 instead of $O(p)$ in general
  3. Extra information provided : search for $\phi : E_0 \to E_1$ of degree 2^{e_2} knowing $\phi(P)$ for all $P \in E_0[3^{e_3}]$
- Point 2 improves tree-based attacks to $O(p^{1/4})$
- Point 3 allows adaptive attacks on key exchange protocol

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 φ = norm of I

Explicit Deuring correspondence

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

Quaternion ℓ power isogeny algorithm

- 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 a maximal order O_0 and a O_0 left ideal I, one can translate the ideal into an isogeny provided
  - We know E_0 and a basis for $\mathrm{End}(E_0) \approx O_0$
  - The norm of I is powersmooth

  (achieved by comparing kernels modulo prime powers)
- Reverse operation also possible under same conditions
- This constructs Deuring correspondence : given O_1,
  1. Compute an ideal between O_0 and O_1
  2. Apply quaternion powersmooth isogeny algorithm
  3. Translate powersmooth ideal to isogeny

Endomorphism ring vs Isogeny computation

- Given an algorithm to compute isogenies between random curves, given E
  1. Perform 2 random walks from E to E_1 and E_2
  2. Compute an isogeny from E_1 and E_2
  3. Composition gives an endomorphism of E
  4. Heuristically 3 endomorphisms give a small index subring
- Given an algorithm to compute endomorphism ring of random curves, given E_1 and E_2
  1. Perform 2 random walks from E_1 and E_2 to E_1' and E_2'
  2. Compute endomorphism ring of E_1' and E_2'
  3. Deduce endomorphism ring of E_1 and E_2
  4. Use quaternion isogeny algorithms to compute a powersmooth isogeny between them

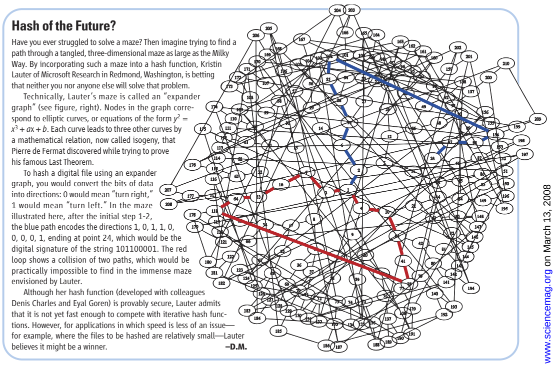
Outline

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.

Charles-Goren-Lauter hash function

$H : \{1, \ldots, \ell\}^* \rightarrow \{$supersingular j-invariants over $\mathbb{F}_{p^2}\}$

- Let p, ℓ be prime numbers, $\ell \neq p$, $p = 1 \bmod 12$
- For every j, define its neighbour set N_j
- For two neighbours j_{i-1}, j_i and for $m_{i+1} \in \{1, \ldots, \ell\}$, define a rule $\sigma(j_{i-1}, j_i, m_{i+1}) = j_{i+1} \in N_{j_i} \setminus \{j_{i-1}\}$
- Let $j_0 \in \mathbb{F}_{p^2}$ be a supersingular j-invariant, and let j_{-1} be one of its neighbours
- To hash a message, start from j_{-1}, j_0, compute j_{i+1} with σ recursively, return last j-invariant

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 $|E_0(\mathbb{F}_{p^2})| = |E_1(\mathbb{F}_{p^2})| = (p+1)^2$.
  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.

Cryptanalysis

- Collision algorithm for special j_0 (see later)
- Trapdoor collision attack : NSA can choose parameters such that they can compute collisions without solving the hard problem (however the collision will leak the trapdoor)
- Still secure for random and honestly generated j_0 : relies on endomorphism ring computation

Outline

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

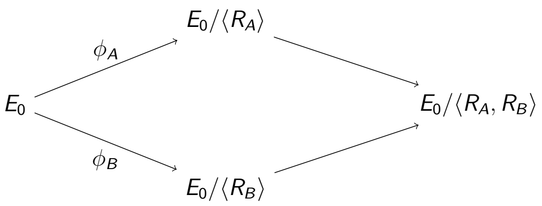
Diffie-Hellman Key Exchange

- 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)

Supersingular key agreement protocol

- Choose ℓ_A, ℓ_B small, distinct primes.
  Choose $p = \ell_A^{e_A}\ell_B^{e_B}f \pm 1$ prime and $E_0/\mathbb{F}_{p^2}$ supersingular.
  For $i = A, B$ choose P_i, Q_i with $\langle P_i, Q_i\rangle = E_0[\ell_i^{e_i}]$.
- Alice chooses $R_A = a_A P_A + b_A Q_A$ with order $\ell_A^{e_A}$; she computes $\phi_A : E_0 \to E_A = E_0/\langle P_A\rangle$ and sends E_A to Bob.
  She also computes and sends $\varphi_A(P_B)$ and $\varphi_A(Q_B)$.
  Bob proceeds similarly.
- Upon receiving E_B, $\varphi_B(P_A)$ and $\varphi_B(Q_A)$, Alice computes $\varphi_B(R_A) = a_A\varphi_B(P_A) + b_A\varphi_B(R_A)$, then she computes $E_{AB} = E_B/\langle\varphi_B(R_A)\rangle = E_0/\langle R_A, R_B\rangle = E_A/\langle\varphi_A(R_B)\rangle$

Supersingular key agreement protocol



Remarks

- Choice of p ensures that $E_0[\ell_i^{e_i}]$ is defined over $\mathbb{F}_{p^2}$, can be generalized at an efficiency cost
- There is ϕ_i of "small" degree $\ell_i^{e_i} \approx \sqrt{p}$ from E_0 to E_i, implies more efficient isogeny tree attacks ; can be avoided at an efficiency cost
- Extra data $\phi_A(P_B), \phi_A(Q_B)$ leads to active attacks (Galbraith-P-Shani-Ti, Asiacrypt 2016) ; impact on passive attacks remains unclear

Public Key Encryption

- Alice chooses keys SK, PK
- She publishes PK but keeps SK secret
- Boc can use PK to encrypt messages for Alice
- Alice can decrypt using SK
- Eve sees PK, yet they cannot distinguish encryptions of any two chosen messages

Public Key Encryption

- Diffie-Hellman-like key exchange protocol leads to ElGamal-like public key encryption
- R_A is secret key and $(E_A, \phi_A(P_B), \phi_A(Q_B))$ is public key
- Encryption of m is (c_1, c_2) where
  - $c_1 = (E_B, \varphi_B(P_A), \varphi_B(Q_A))$
  - c_2 is some one-time pad of m with shared key E_{AB}
- To decrypt : first recompute the shared key then undo one-time pad

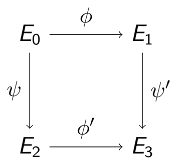
Outline

Identification protocol / proof of knowledge

- Prover wants to prove knowledge of a secret to Verifier without revealing it (can be used for authentication)
- Often 3-round protocol, with **commitment**, **challenge** and **answer** messages
- Security requirements :
  - **Correctness** : if Prover knows the secret then he can convince Verifier
  - **Soundness** : if Prover convinces the Verifier then he must know the secret
  - **Zero-knowledge** : nothing is leaked about the secret

Jao-de Feo-Plût identification protocol

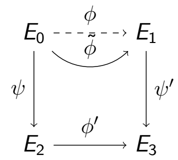
- Proof of knowledge of an isogeny ϕ between two given curves E_0 and E_1
- Proof inspired by classical proof for graph isomorphism, and commutative diagram in key agreement protocol



- 3-round protocol : Prover commits with E_2 and E_3 ; Verifier answers with one bit ; depending on this bit Prover either reveals ϕ' or Prover reveals both ψ and ψ'

Jao-de Feo-Plût identification protocol

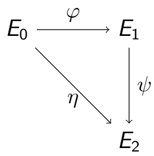
- Correctness : clear
- 2-special soundness : answer for both bit values gives $\tilde{\phi} = \hat{\psi}' \circ \phi' \circ \hat{\psi}$. Compute $\ker \phi = E_0[\ell_A^{e_A}] \cap \ker(\tilde{\phi})$.



- Zero-knowledge : relies on ad hoc isogeny problems

New protocol based on endomorphism ring computation (Galbraith-P-Silva Vélon)

- Goal is to rely solely on the endomorphism ring computation problem
- Proof is actually closer to graph isomorphism proof

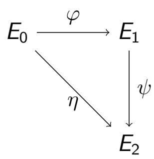


New identification protocol

- Choose E_0 special such that End(E_0) is known
- Choose φ of degree large enough such that E_1 is uniformly distributed
- Secret : knowledge of isogeny φ between E_0 and E_1. Equivalently, knowledge of the endomorphism ring of E_1
- Prover chooses random ψ with degree large enough so that E_2 is uniformly distributed, and commits with E_2. Verifier challenges with one bit. Depending on this bit Prover answers either with ψ or with an isogeny $\eta : E_0 \to E_2$

Security Properties

- Correctness is clear
- Soundness based on a "standard" isogeny problem
- Note that the isogeny $\tilde{\eta} = \psi \circ \varphi$ cannot be returned by Prover, as it would reveal the secret φ



- To achieve zero-knowledge Prover needs to compute a "fresh" isogeny from E_0 to E_2, independent of φ and ψ

Achieving Zero-Knowledge

- Algorithm to compute η :
  1. Let $O_0 \approx \text{End}(E_0)$ with $O_0 \subset B_{p,\infty}$
  2. Compute O_0-left ideal I corresponding to $\tilde{\eta} = \psi \circ \varphi$
  3. Apply quaternion powersmooth isogeny algorithm (variant of ANTS 2014) to get another O_0-left ideal J in the same class as I
  4. Compute isogeny η corresponding to J
- Remarks
  - Steps 2 and 4 use knowledge of End(E_0)
  - Powersmooth requirement for efficiency
  - We prove η is independent of $\tilde{\eta}$, except for the fact that they connect the same curves

Signature schemes

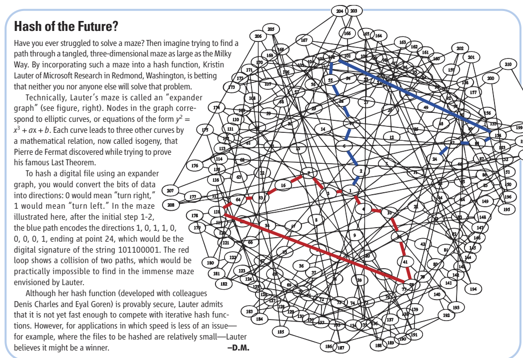
- Alice chooses two keys PK and SK
- She publishes PK and keeps SK secret
- She signs messages with SK
- Signatures can be verified with PK
- Security property : existential unforgeability under chosen message attacks

Signature schemes

- Can use Fiat-Shamir transform (or any alternative) to turn the above ID protocols into signature schemes, in the random oracle model
- Secret key is isogeny φ ; public key is E_1
- Signature on m : repeat the identification protocol, with challenge bits replaced by the hash of the message and commitments. The signature contains the commitments and the responses. (Or the hash and responses.)
- To verify, recompute the hash and check all responses

Outline

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.

Strategy to break CGL hash

- 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)
$$E \to O \approx \text{End}(E)$$
- Strategy to break CGL : constructive correspondence
  - Translate collision and preimage resistance properties in the quaternion world
  - Break collision and preimage resistance properties in the quaternion world
  - Translate the attacks (as much as possible) back to the elliptic curve world

CGL attack on special initial points

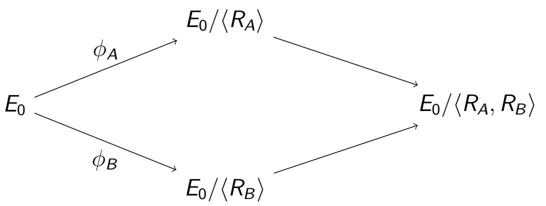
- What : collision attack for special parameters compute an endomorphism of E_0 of degree ℓ^e when $\text{End}(E_0)$ is known
- Compute $\alpha \in O_0$ of norm ℓ^e
- Deduce $I_i = O_0\alpha + O_0\ell^i$, $i = 1, \ldots, e$
- For each i
  - Compute $J_i \approx I_i$ with powersmooth norm
  - Compute corresponding isogeny φ_i and j-invariant j_i
- Deduce a collision path $(j_0, j_1, \ldots, j_e = j_0)$

A trapdoor collision attack

- What : compute genuine-looking parameters together with a collision trapdoor
- Choose a random path from j_0, ending at j_1
- Reveal j_1 as initial point in the graph
- Keep the path as a trapdoor
- Use collision attack on j_0
- Combine paths to produce collision on j_1
- Note : using the trapdoor will reveal it

Outline

Supersingular key agreement protocol



Adaptive attack on supersingular key agreement (Galbraith-P-Shani-Ti)

- What : if Alice uses static secret key $R_A = a_A P_A + b_A Q_A$, run key agreement protocol several times and deduce R_A
- Normal execution : on input $(E_B, \phi_B(P_A), \phi_B(Q_A))$, Alice returns $E_{AB} = E_B/\langle a_A\phi_B(P_A) + b_A\phi_b(Q_A)\rangle$
- Adaptive attack : make Alice compute $E_B/\langle a_A U_i + b_A V_i\rangle$ for well-chosen U_i, V_i, and recover the secret piecewise
- Sometimes Alice only returns a hash of $j(E_{AB})$: hence adversary does not get corresponding E_{AB} but can still vary inputs and observe when outputs change

Attack when $\ell_A = 2$

- Can assume $R_A = P_A + \alpha Q_A$ with $\alpha = \sum \alpha_i 2^i \in (\mathbb{Z}_{2^{e_2}})^*$
- Send $U_i = \phi_B(a_i P_A + b_i Q_A)$ and $V_i = \phi_B(c_i P_A + d_i Q_A)$ in query i such that
  1. $\langle U_i + \alpha V_i\rangle = \langle (a_i + \alpha c_i)P_A + (b_i + \alpha d_i)Q_A\rangle$ is equal to $\langle P_A + \alpha Q_A\rangle$ if and only if $\alpha_i = 0$
  2. U_i and V_i both have order 2^n
  3. $e_{2^n}(U_i, V_i) = e_{2^n}(\phi_B(P_A), \phi_B(Q_A)) = e_{2^n}(P_A, Q_A)^{3^m}$
- First condition to distinguish $\alpha_i = 0$ from $\alpha_i = 1$; second and third conditions to pass validity checks
- See Asiacrypt paper for how to choose a_i, b_i, c_i, d_i

Other results on key agreement

- The degree condition on the isogeny problems could a priori have made them harder to break. We prove this is not the case : computing the endomorphism rings of both curves is enough to break the isogeny problems in supersingular key agreement protocol.
- Side-channel attack recovering a static key from partial leakage of shared keys

Outline

Conclusion

- 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 ?