Some remarks on the security of isogeny-based cryptosystems

Sergey Grebnev

QApp

Outline of the talk

- An intro to isogenies and supersingular curves.
- A brief description of SIDH.
- Forsythia – an instance of SIDH.
- A review of cryptanalytical attacks.
- Security estimates with respect to success probability for SIKE and Forsythia.

Isogenies

Isogeny — a rational homomorphism between two elliptic curves.

Theorem 1.

(Serre-Tate) Two curves are isogenous over K iff they have the same number of points.

The complexity of computation of an isogeny of degree l (that is, with the kernel's size of l) is $O(l)$ operations in K (**Vélu's formulae**).
An isogeny of a smooth degree decomposes into a composition of isogenies of small degrees.

Supersingular curves

An elliptic curve defined over K with the characteristic p is **supersingular**, if #E $\equiv 1 \pmod{p}$.

Theorem 2.

(Deuring) If E is supersingular, then

1. E is isomorphic to a curve defined over GF(p^2)
2. every endomorphism of E is defined over GF(p^2)

This means that we may only consider supersingular curves over GF(p^2). There are about $p/12$ such curves.

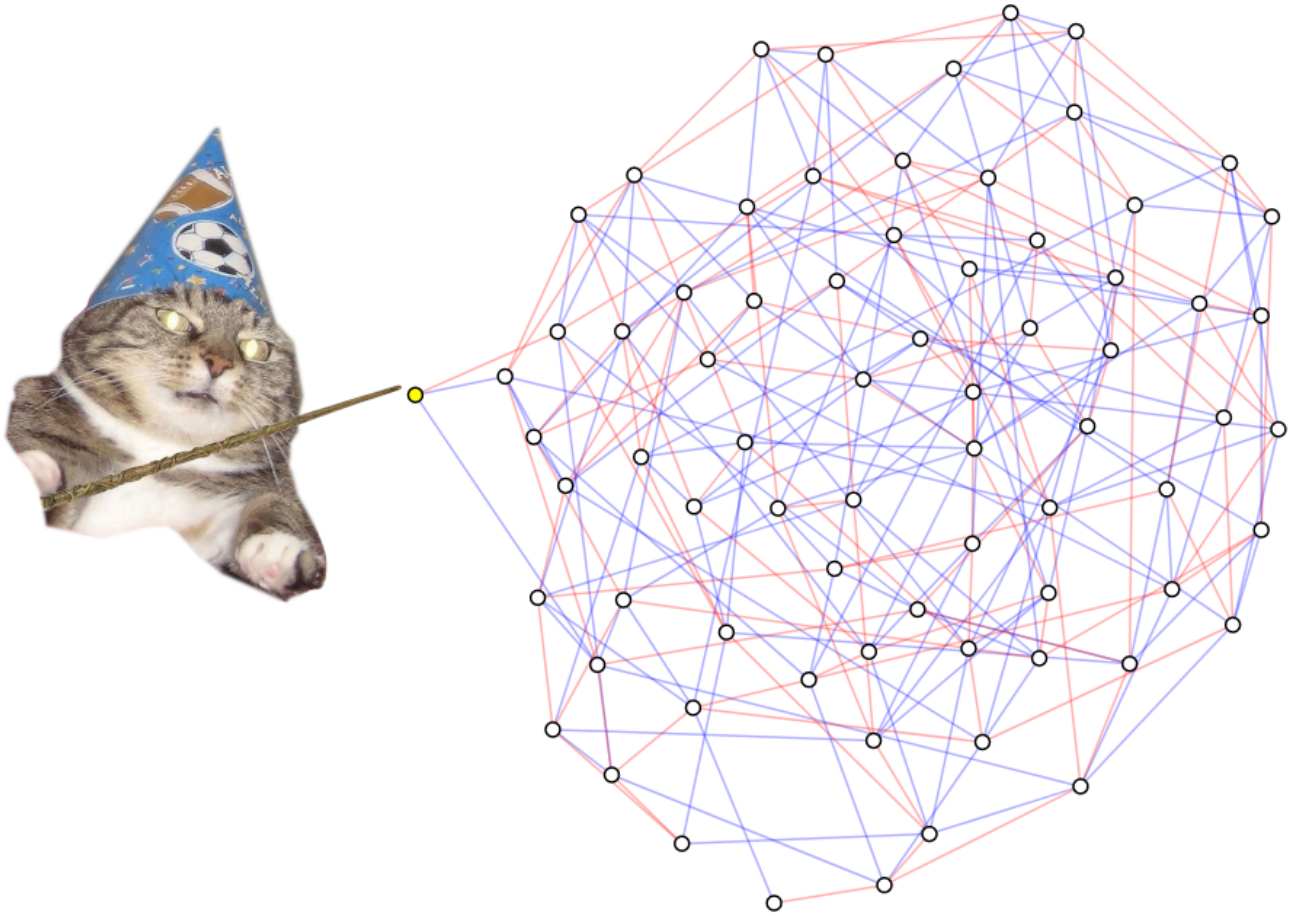
Supersingular curves

Isogeny graph: a graph, vertices of which are the classes of isomorphism of elliptic curves, and they are connected by an edge iff the corresponding representatives are isogenous. Consider only isogenies of degree l to obtain the l**-isogeny graph**.
For a prime $l \neq p$ and supersingular elliptic curves the l-isogeny graph is:

- $(l + 1)$-regular
- contains an unique connected component (all the supersingular curves)
- **an expander**

An example

$p = 2^5 \cdot 3^3 - 1 = 863$; 73 supersingular j-invariants.
2-isogenies, 3-isogenies, $E_0 : y^2 = x^3 + x$.



(c) `https://isogenies.enricflorit.com/visualizations/graph.html`

SIDH in a slide

Public parameters:

- a large prime $p = l_A^{e_A} l_B^{e_B} \cdot f - 1$, and a supersingular $E_0(GF(p^2))$
- bases $\{P_A, Q_A\}$ and $\{P_B, Q_B\}$ of $E_0[l_A^{e_A}]$ and $E_0[l_B^{e_B}]$

A	B
$n_A \in_R \mathbb{Z}/l_A^{e_A}\mathbb{Z}$,	$n_B \in_R \mathbb{Z}/l_B^{e_B}\mathbb{Z}$,
$K_A := \langle P_A + [n_A]Q_A \rangle$	$K_B := \langle P_B + [n_B]Q_B \rangle$
compute	compute
$\varphi_A : E_0 \to E_0/K_A = E_A$	$\varphi_B : E_0 \to E_0/K_B = E_B$
$E_A, \varphi_A(P_B), \varphi_A(Q_B)$	$E_B, \varphi_B(P_A), \varphi_B(Q_A)$
compute	compute
$K'_A = \langle \varphi_B(P_A) + [n_A]\varphi_B(Q_A) \rangle$	$K'_B = \langle \varphi_A(P_B) + [n_B]\varphi_A(Q_B) \rangle$
$s = j((E/\langle K_B \rangle)/\langle K'_A \rangle)$	$s = j((E/\langle K_A \rangle)/\langle K'_B \rangle)$

Forsythia: a brief description

- based upon the original SIDH protocol;
- supports ephemeral-only key exchange;
- session-key secure;
- has its own starting curve $E_{19}(GF(p)) : y^2 = x^3 - 2^3 \cdot 19x + 2 \cdot 19^2$;
- three security levels;
- Montgomery representation.

Форзиция (Forsythia) is a genus of flowering plants in the olive family (Oleaceae).
`https://github.com/s-v-grebnev/Forsythia`

Analysis: the problem

Problem 3.

Computational Supersingular Isogeny – CSSI: let $\phi_1 : E_0 \to E_1$ – an isogeny with the kernel $R_1 + [n_1]S_1$, where n_1 is chosen uniformly at random from the interval $[1, l_1^{e_1}]$. Given E_1 and images $\phi_1(R_2), \phi_1(S_2)$ of the points, find the generator of $\langle R_1 + [n_1]S_1 \rangle$.

Our case

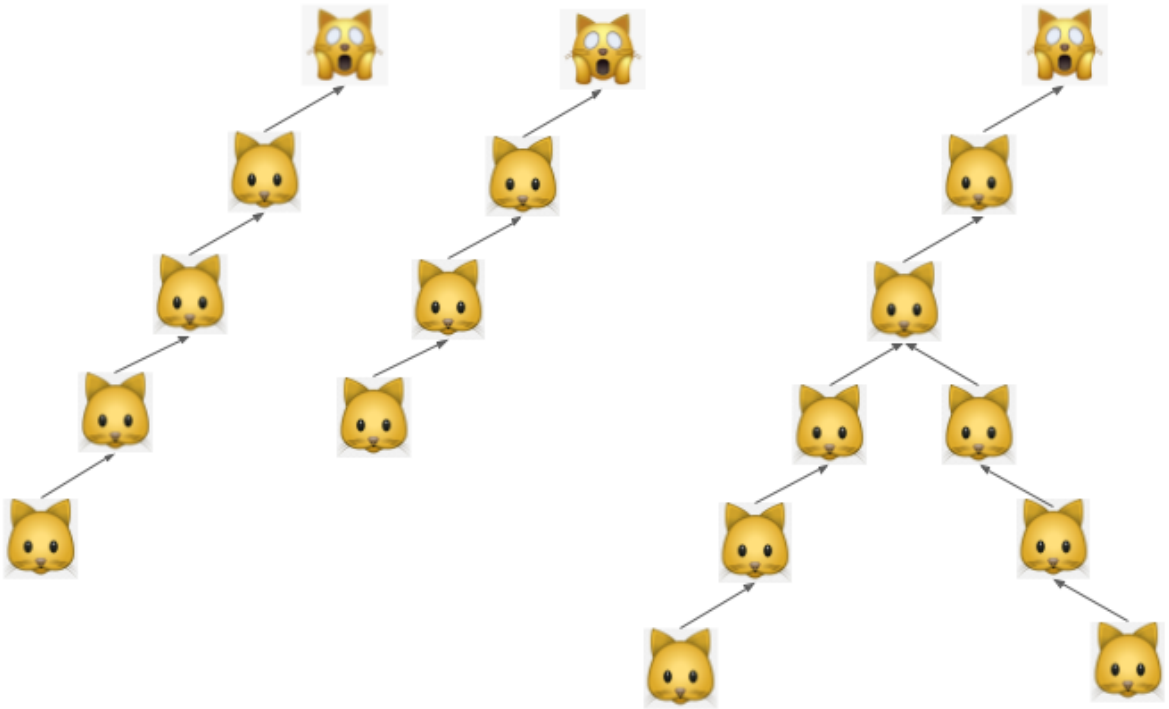
A simplified case of the CSSI problem may be stated as follows. Let us have a (secret) l^e-isogeny $\phi : E \to E/G$ for a subgroup $G \subset E$ of the order $l^e \approx p^{1/2}$. The problem is to find the generator of G (or, equivalently, the isogeny ϕ).

Analysis: classical computer

Brute-force: every supersingular elliptic curve $E(GF(p^2))$ has $(l+1)l^{e-1}$ cyclic subgroups of order l^e; thus, brute-force requires $O(l^e)$ or $O(p^{1/2})$ samplings.
Meet-in-the-middle: We construct a pair of trees such that the leaves of the first define classes of isomorphisms of the curves, $l^{e/2}$-isogenous to E; leaves of the second – classes of isomorphisms of the curves, $l^{e/2}$-isogenous to E/G. Each set contains no more that $(l+1)l^{e/2-1}$ classes.
By brute force testing we detect $l^{e/2}$-isogenies $\phi_1 : E \to E'$ and $\phi_2 : E/G \to E''$ such that there exists an isomorphism $\psi : E' \to E''$. At last, we have l^e-isogeny $\phi = \widehat{\phi_2} \circ \psi \circ \phi_1$.
memory – $O(p^{1/4})$, time – $O(p^{1/4})$.
(Adj et al., eprint 2018/313)

Analysis: classical computer

Parallel collision search



Analysis: classical computer

Let $S = \{0,1\} \times \{0,\ldots,(l+1)l^{e/2-1}-1\}$, $E_0 = E$, $E_1 = E/G$. Each pair $(i,y) \in S$ defines a subgroup of elliptic curve E_i.

Example 4.

For $l = 2$ the correspondence between the pairs $(i,y) = (i,(b,k)) \in \{0,1\} \times \{0,1,2\} \times \{0,\ldots,l^{e/2-1}-1\}$ and cyclic subgroups $\langle R_i \rangle \subset E_i$ is given by

$$R_i = \begin{cases} P_i + [b2^{e/2-1} + k]Q_i, & \text{if } b = 0,1 \\ [2k]P_i + Q_i, & \text{if } b = 2, \end{cases}$$

where $\langle P_i, Q_i \rangle = E_i[2^{e/2}]$.

Analysis: classical computer

Let $h : S \to E_0(GF(p^2)) \cup E_1(GF(p^2))$, $h : (i, y) \mapsto R_i$, and let the iteration function $f : S \to S$ be a function that on an input pair (i, y) computes an $l^{e/2}$-isogeny with the kernel $\langle R_i \rangle$, computes the j-invariant $j(E_i / \langle R_i \rangle)$ and maps it to S by some pseudorandom function $g : GF(p^2) \to S$.
Average time:

$$O\left(\frac{p^{3/8}}{m\, w^{1/2}} t\right)$$

(m – the number of processors, w – memory, t – the complexity of iteration function)
(Costello et al., eprint 2019/298)

Analysis: quantum computer

Claw-finding algorithm (Tani): let $g_1 : X_1 \to Y$, $g_2 : X_2 \to Y$, find x_1, x_2: $g_1(x_1) = g_2(x_2)$.
Let $\#X_1 \approx \#X_2 \approx N$, $\#Y \gg N$, time is – $O(N^{2/3})$.
We have time – $O(p^{1/6})$ (and memory $O(p^{1/6})$).
Grover's method: time – $O(p^{1/4})$, memory – $O(1)$.
(Jacques et Schanck, eprint 2019/103)
Quantum golden collision search: $O(p^{3/14})$ gates and $O(p^{1/14})$ memory
(Jacques et Schrottenloher, SAC 2020)

Other attacks

- Static-key attacks (Galbraith et al., eprint 2016/859)
- Unbalanced primes (de Quehen et al., eprint 2020/633)
- Torsion points attacks (de Quehen et al., eprint 2020/633)

Cannot be applied:

- Ephemeral-only key exchange
- Primes are well balanced, no “backdoor” staring curves are known.

Choice of parameters

- p – balanced;
- $\left(\frac{-19}{p}\right) = -1$ (for the starting curve E_{19} to be supersingular).

p	Formula	Classical	Quantum security	
			Grover	Golden coll.
p_{271}	$2^{132} \cdot 3^{85} \cdot 11 - 1$	82	130	119
p_{415}	$2^{208} \cdot 3^{129} \cdot 5 - 1$	135	202	180
p_{754}	$2^{372} \cdot 3^{239} \cdot 7 - 1$	262	371	326

Success probability

Definition 5.

Let X be a computational problem, with instances x produced by an algorithm Gen(1^λ). Let ε_0 be some fixed upper bound on success probabilities of interest (possibly a function of λ). Then X has λ-bit security level if, for every adversary A, if A(x) runs in time t and succeeds with probability $\varepsilon < \varepsilon_0$ we have $t/\varepsilon \geq 2^\lambda$.

S. Galbraith, Security levels in cryptography, ACISP 2020.

Success probability

The probability $P(t)$ of success over a search space S with $\#S = n$ after t iterations with z memory cells:

$$P(t) = \begin{cases} 1 - \exp(-t^2/(2n)), & \text{if } t < z, \\ 1 - (1 - \frac{z}{n})^{t-z} \exp(-z^2/(2n)), & \text{otherwise.} \end{cases} \tag{1}$$

The setting

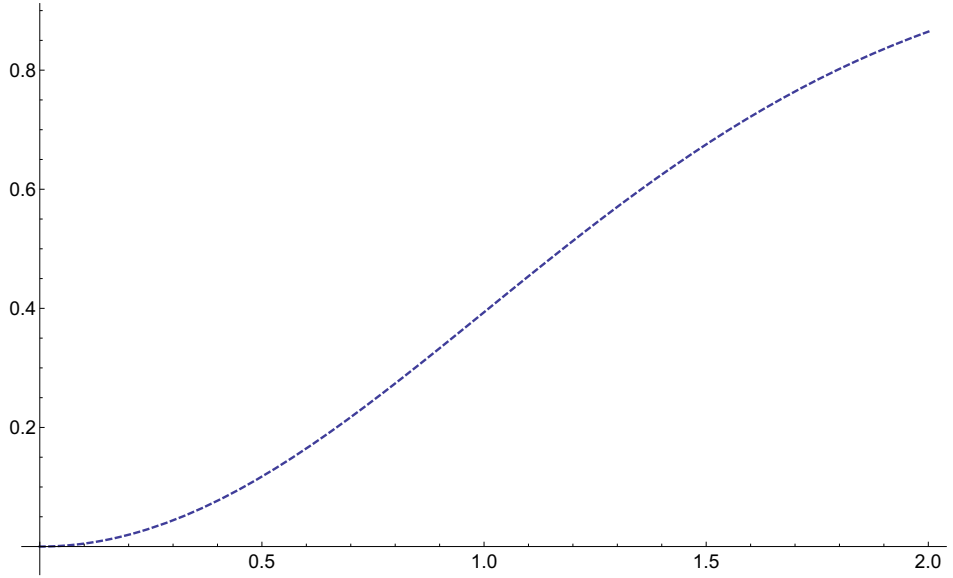
- A computer with 2^{64} memory cells
- Capable of executing $T = 2^{80}$ operations (p_{415}, p_{754}) or $T = 2^{64}$ operations for p_{271}
- Running parallel collision search
- Upper bound on success probability ε_0 now are 2^{-48}, 2^{-176} and 2^{-16}

Calculation

1. Substitute $\mathsf{n} = \mathsf{p}^{1/4}$ into equation (1) and rewrite it for a new variable $\theta = \mathsf{t}/\sqrt{\mathsf{n}}$:

$$\mathsf{P}(\theta) = \begin{cases} 1 - \exp(-\theta^2/(2)), & \text{if } \theta < \mathsf{z}/\sqrt{\mathsf{n}}, \\ 1 - (1 - \frac{\mathsf{z}}{\mathsf{n}})^{\theta\sqrt{\mathsf{n}} - \mathsf{z}} \exp(-\mathsf{z}^2/(2\mathsf{n})), & \text{otherwise.} \end{cases} \tag{2}$$

2. Solve (2) for θ with $\mathsf{P} = \varepsilon_0$.
3. Compute the bit-size of the p required to obtain the claimed security level as $\mathsf{l} = 4(\log_2 \mathsf{T} - \log_2(\theta))$.



Forsythia

Claimed security level	Forsythia prime (bit-size)	Calculated prime (bit-size)
80 bits	**271**	286
128 bits	415	414
256 bits	754	670

(cf. Grebnev, CTCrypt'2019: Limonnitsa, 963 bits)

SIKE

Claimed NIST security level	SIKE prime (bit-size)	Calculated prime (bit-size)
I	434	414
III	610	541
V	751	670

QApp

Thanks for attention.

sg@qapp.tech