A Powerful Active Attack on Supersingular Isogeny Diffie-Hellman (SIDH) Key Exchange

Ankit Kumar Misra Aman Singh

CS 741: Advanced Network Security and Cryptography,
Autumn 2022, IIT Bombay

December 5, 2022

Introduction

- Supersingular Isogeny Key Exchange or SIKE is a proposed post-quantum cryptographic algorithm
- Smallest key sizes among its contenders
- Perfect forward secrecy

Supersingular curves

- SIKE works in finite fields of the form $\mathbb{F}_{p^2} = \mathbb{F}_p(i)$ with elements represented as $u + vi$ where $u, v \in \mathbb{F}_p$

Definition

The curve $E_\lambda : y^2 = x(x-1)(x-\lambda)$ is supersingular if and only if λ is a root of $H_p(x) = \sum_{i=0}^{(p-1)/2} \binom{(p-1)/2}{i}^2 x^i$ such that $\lambda \neq 0, 1$

- We are interested performing operations over j-invariants of supersingular elliptic curves in $\mathbb{F}_{p^2}$

j-invariant

Definition

For a curve in Montgomery form $E_a : y^2 = x^3 + ax^2 + x$,

$$j(E_a) = 256\frac{(a^2-3)^3}{(a^2-4)}$$

- Isomorphic elliptic curves have equal j-invariant and curve with equal j-invariants are isomorphic
- Number of supersingular j-invariants in $\mathbb{F}_{p^2}$ is approximately $\lfloor p/12 \rfloor$

Isogenies

- Isogenies are group homomorphisms on elliptic curves i.e they preserve the group structure. For an isogeny $\phi : E \to E'$

$$\phi(A + B) = \phi(A) + \phi(B)$$

- Unlike isomorphisms, isogenies do not conserve the j-invariant
- The kernel $ker(\phi)$ of an isogeny contains the group elements that are mapped to the identity element $\mathcal{O}$ under ϕ
- The order of an isogeny is the number of elements in its kernel

Definition

The d-torsion isogeny is the multiplication-by-d map represented as $[d] : E \to E'$ such that $[d]P = dP$. $E[d]$ is the kernel of $[d]$ in E.

Isogeny graph

Theorem

$E[d] \cong \mathbb{Z}_d \times \mathbb{Z}_d$

Theorem

If $\phi : E \to E'$ *is a separable isogeny of degree* d, $ker(\phi) \subseteq E[d]$

- To proceed further, we would like to compute 2-isogeny and 3-isogeny graphs starting from a fixed curve E
- A d-isogeny graph has vertices representing an equivalence class of elliptic curves with same j-invariant and an edge if there is an isogeny that connects them.
- Due to the above theorems, $\langle P, Q \rangle = E[2]$ and there exist three 2-isogenies from each E with kernels $\langle P \rangle, \langle Q \rangle, \langle P + Q \rangle$

Composing isogenies

- To implement SIDH we would need to compute isogenies of degree 2^d where $d \approx 200$
- This is done by composing d 2-isogenies
- We wish to compute $\phi : E \to E/\langle S \rangle$ where S is of order 2^d
- As the first step, we calculate $S' = [2^{d-1}]S$ which has order 2. Associated with S' is the isogeny $\phi' : E \to E/\langle S' \rangle$ of degree 2 (i.e. a step in the 2-isogeny graph)
- Now, $0 = \phi'(S') = \phi'(2^{d-1}S) = 2^{d-1}\phi'(S)$. So, $\phi'(S)$ is a point of order $d-1$ in $E/\langle S' \rangle$
- By continuing this process, we can get an isogeny of degree 2^d in d steps.

SIDH protocol

- Prime p is chosen to be of the form $2^{e_A}3^{e_B} - 1$ with $2^{e_A} \approx 3^{e_B}$
- **Protocol parameters**: E, P_A, Q_A, P_B, Q_B where $\langle P_A, Q_A \rangle = E[2^{e_A}]$ and $\langle P_B, Q_B \rangle = E[3^{e_B}]$
- Alice and Bob choose secret integers $k_A \in [0, 2^{e_A} - 1], k_B \in [0, 3^{e_B} - 1]$ and compute the isogeny ϕ_A and ϕ_B such that $ker(\phi_A) = \langle P_A + [k_A]Q_A \rangle, ker(\phi_B) = \langle P_B + [k_B]Q_B \rangle$
- **Public keys**: $PK_A = (\phi_A(E) = E_A, \phi_A(P_B), \phi_A(Q_B))$ and $PK_B = (\phi_B(E) = E_B, \phi_B(P_A), \phi_B(Q_A))$
- **Shared secret**: Alice computes $S_{BA} = \phi_B(P_A + [k_A]Q_A) = \phi_B(P_A) + [k_A]\phi_B(Q_A)$ and hence also $E_{BA} = \phi_B(E)/\langle S_{BA} \rangle$. Similarly, Bob calculates E_{AB}

How can we attack SIDH?

- Galbraith et al., 2016.
- If a party (say, Alice) does not change her private key k_A, her full private key can be recovered in $\mathcal{O}(\log p)$ interactions.
- For the attack, interactions with Alice are modeled in terms of accessing an oracle O:
  - Input: $E_B, \phi_B(P_A), \phi_B(Q_A), E_{AB}$.
  - Output: 1 if the j-invariant of E_{AB} equals that of $E_B/\langle \phi_B(P_A) + [k_A]\phi_B(Q_A) \rangle$, and 0 otherwise.
- Validation checks in SIDH can prevent basic attacks.
  - Check that public key points $\phi_B(P_A), \phi_B(Q_A)$ have full order of 2^n, to prevent small subgroup attacks.
  - Weil pairing check for independence: $e_{2^n}(\phi_B(P_A), \phi_B(Q_A)) = e_{2^n}(P_A, Q_A)^{3^m}$.

Extracting the LSB of k_A

- For simplicity, let $R = \phi_B(P_A)$ and $S = \phi_B(Q_A)$.
- Attacker queries the oracle on $(E_B, R, S + [2^{n-1}]R, E_{AB})$.
- Oracle returns 1 if and only if E_{AB} and $E_B / \langle R + k_A(S + [2^{n-1}]R)\rangle$ are isomorphic, i.e., $\langle R + k_A(S + [2^{n-1}]R)\rangle = \langle R + k_A S\rangle$.

Lemma

Let $R, S \in E[2^n]$ be linearly independent points of order 2^n, and $k_A \in \mathbb{Z}_{2^n}$. Then, $\langle R + k_A(S + [2^{n-1}]R)\rangle = \langle R + k_A S\rangle$ if and only if k_A is even.

Extracting the LSB of k_A

Lemma

Let $R, S \in E[2^n]$ be linearly independent points of order 2^n, and $k_A \in \mathbb{Z}_{2^n}$. Then, $\langle R + [k_A](S + [2^{n-1}]R) \rangle = \langle R + [k_A]S \rangle$ if and only if k_A is even.

Proof.

($\Longrightarrow$) Groups generated by $R + [k_A](S + [2^{n-1}]R)$ and $R + [k_A]S$ are equal, so there exists $\lambda \in \mathbb{Z}^*_{2^n}$ such that

$$\lambda(R + [k_A](S + [2^{n-1}]R)) = R + [k_A]S.$$

By linear independence of R and S, we have $\lambda = 1$, and thus $[k_A][2^{n-1}]R = 0$. Since R has order 2^n, k_A must be even.

($\Longleftarrow$) If k_A is even, then $[k_A][2^{n-1}]R = 0$. Hence proved. $\square$

Extracting the LSB of k_A

- The lemma just described implies that, for query $(E_B, R, S + [2^{n-1}]R, E_{AB})$, the oracle returns
  - 1 if and only if k_A is even.
  - 0 if and only if k_A is odd.
- LSB of k_A has been determined by a single oracle access.
  - LSB $= 1-$oracle's response
- A similar strategy is adopted for the higher-order bits.

Strategy to extract an arbitrary bit of k_A

- Let $k_A = k_0 + 2^1 k_1 + \cdots + 2^{n-1} k_{n-1} = \mathcal{K}_i + 2^i k_i + 2^{i+1} k'$, where the i LSBs (given by $\mathcal{K}_i$) are already known.
- The attacker queries $(E_B, [a]R + [b]S, [c]R + [d]S, E_{AB})$ for appropriately chosen a, b, c, d.
- Conditions to be satisfied:
  - The oracle's response should reveal k_i.
  - Attack should pass undetected through order checking and Weil pairing validation checks.

Designing the attacker's query

- More formally, the following have to be satisfied:
  - If $k_i = 0$ then $\langle [a + k_A c]R + [b + k_A d]S \rangle = \langle R + k_A S \rangle$.
  - If $k_i = 1$ then $\langle [a + k_A c]R + [b + k_A d]S \rangle \neq \langle R + k_A S \rangle$.
  - $[a]R + [b]S$ and $[c]R + [d]S$ both have order 2^n.
  - $e_{2^n}([a]R + [b]S, [c]R + [d]S) = e_{2^n}(R, S) = e_{2^n}(P_A, Q_A)^{3^m}$.
- The first three are satisfied by

$$a = 1 \qquad b = -2^{n-i-1}\mathcal{K}_i$$
$$c = 0 \qquad d = 1 + 2^{n-i-1}$$

- Thus, $R' = [\theta](R - [2^{n-i-1}\mathcal{K}_i]S)$ and $S' = [\theta][1 + 2^{n-i-1}]S$.

Designing the attacker's query

- For the last condition, we need $e_{2^n}(R', S') = e_{2^n}(R, S)^{\theta^2(1+2^{n-i-1})} = e_{2^n}(R, S)$.
- So we use $\theta = \sqrt{(1 + 2^{n-i-1})^{-1}} \pmod{2^n}$.
- This square root can be shown to exist for $n - i - 1 \geq 3$, i.e., $i \leq n - 4$.
- Oracle queries are used for $i = 0, 1, \ldots, n - 4$, followed by brute force search over all 8 possibilities for $i = n - 3, n - 2, n - 1$, checking whether $E/\langle R + k_A S\rangle$ equals E_A in each case.

Implementation

- We implemented SIDH and the attack in Sage, and verified their working for small values of e_A and e_B.
- Code available at `https://github.com/ankitkmisra/SIDH`.

References

1. Galbraith et al., *On the Security of Supersingular Isogeny Cryptosystems*, Asiacrypt 2016.
2. PQCRYPTO's *Summer School on Post-Quantum Cryptography* in Eindhoven, 2017.
3. Craig Costello, *Supersingular Isogeny Key Exchange for Beginners*, Selected Areas in Cryptography (SAC) 2019.
4. William Borgeaud's blog post on *Isogeny-Based Crypto Part 1: The SIDH Protocol*, 2020.
5. Sage documentation.