EPFL

Implementing countermeasures for attacks on Supersingular Isogeny Diffie-Hellman (SIDH)

Malo RANZETTI

Semester Project Presentation

June 2023

Ecole Polytechnique Fédérale de Lausanne

AWARDS

The two challenge instances were generated during a live and recorded talk at NIST's 3rd standardization conference on June 9, 2021. The instances (and the source code used to generate them) can be found **here**.

- The award for solving the $IKEp182 challenge was $5,000 USD and was claimed on August 28, 2021. $IKEp182 was solved by Aleksei Udovenko and Giuseppe Vitto.
- The award for solving the $IKEp217 challenge was $50,000 USD and was claimed on July 22, 2022. $IKEp217 was solved by Wouter Castryck and Thomas Decru.

Fig.1: Microsoft $IKE challenge

AN EFFICIENT KEY RECOVERY ATTACK ON SIDH (PRELIMINARY VERSION)

WOUTER CASTRYCK AND THOMAS DECRU

imec-COSIC, KU Leuven

ABSTRACT. We present an efficient key recovery attack on the Supersingular Isogeny Diffie–Hellman protocol (SIDH), based on a "glue-and-split" theorem due to Kani. Our attack exploits the existence of a small non-scalar endomorphism on the starting curve, and it also relies on the auxiliary torsion point information that Alice and Bob share during the protocol. Our Magma implementation breaks the instantiation `SIKEp434`, which aims at security level 1 of the Post-Quantum Cryptography standardization process currently ran by NIST, in about one hour on a single core. This is a preliminary version of a longer article in preparation.

Fig.2: Paper by Castryck and Decru

- Supersingular elliptic curves, isogenies, post-quantum cryptography: modern and rapidly evolving field
- Introduction to Supersingular Isogeny Diffie-Hellman (SIDH) key exchange in 2011 (part of SIKE standard)
- Discovery of a polynomial time attack against SIDH in 2022 by Castryck and Decru
- Introduction of the M-SIDH scheme as a countermeasure

Project objective: Implementation and evaluation of M-SIDH in terms of performance

Supersingular Isogeny Diffie-Hellman

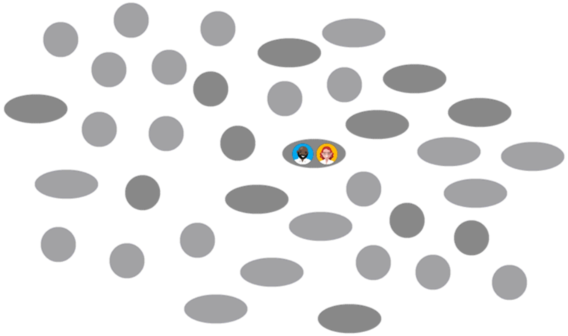
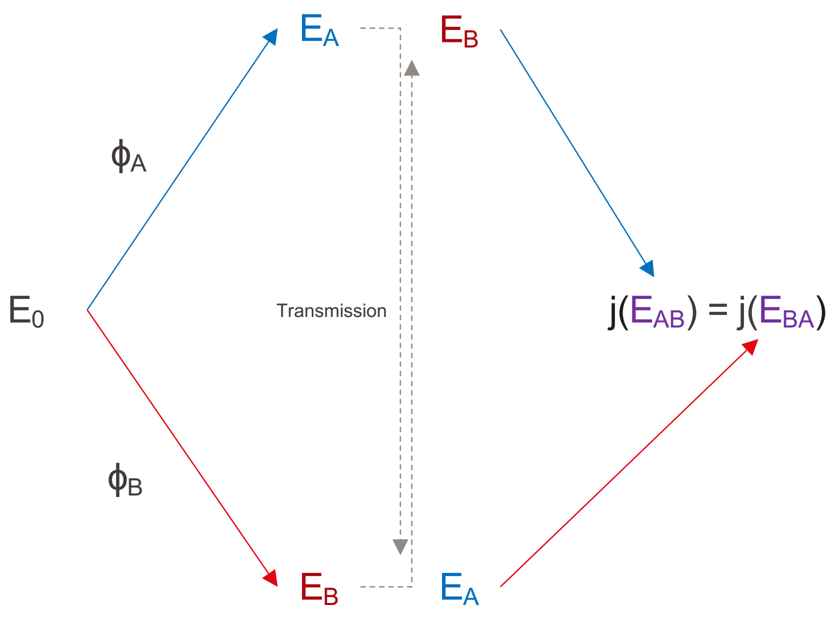


Fig.3: Animation illustrating an isogeny based DH [Microsoft]

Basic idea:

Use isogenies between curves as the trapdoor function in a Diffie-Hellman scheme

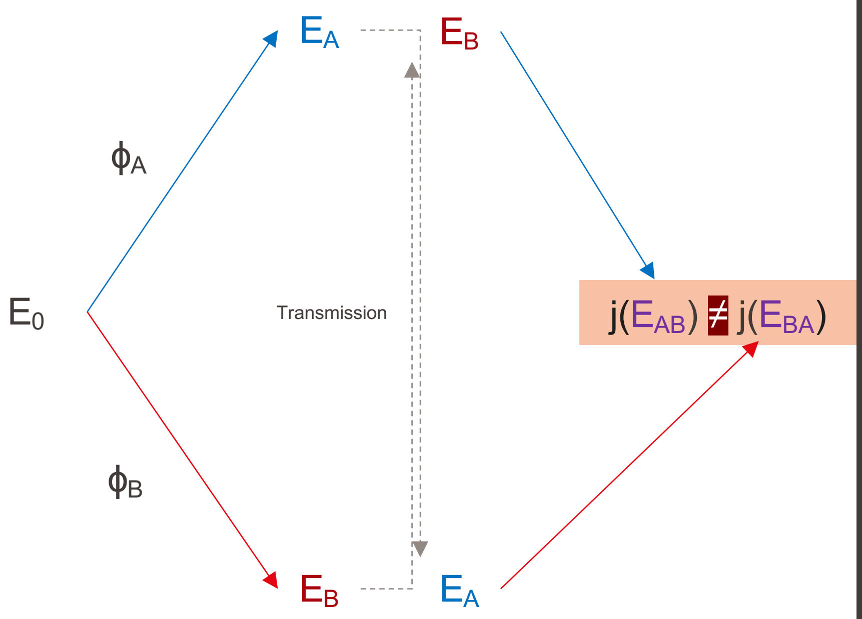


Supersingular Isogeny Diffie-Hellman

Standard DH layout with:

- Starting curve E_0
- Secret isogenies ϕ_A and ϕ_B
- Resulting curves $E_{AB} = E_{BA}$ have the same j-invariant

Malo RANZETTI

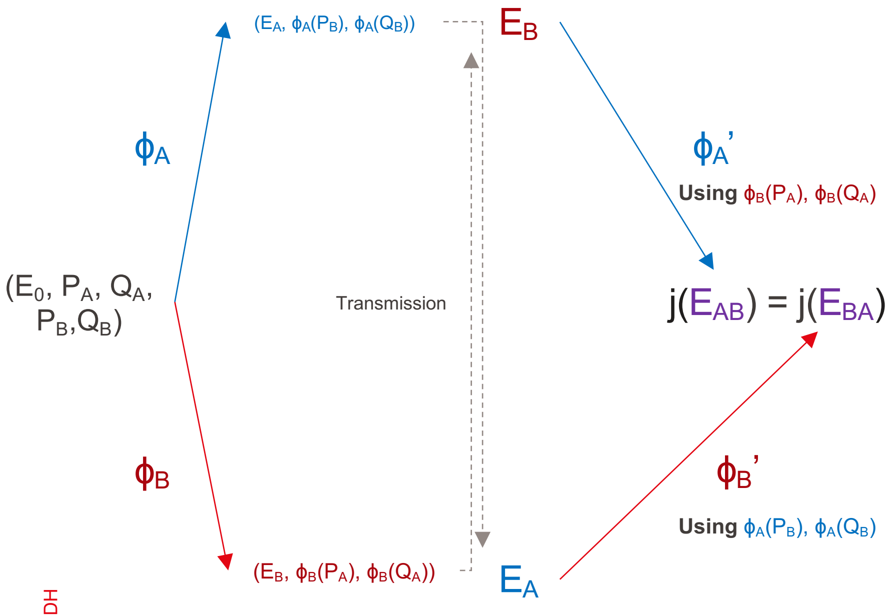


Supersingular Isogeny Diffie-Hellman

Standard DH layout with:

- Starting curve E_0
- Secret isogenies ϕ_A and ϕ_B
- Resulting curves $E_{AB} = E_{BA}$ have the same j-invariant

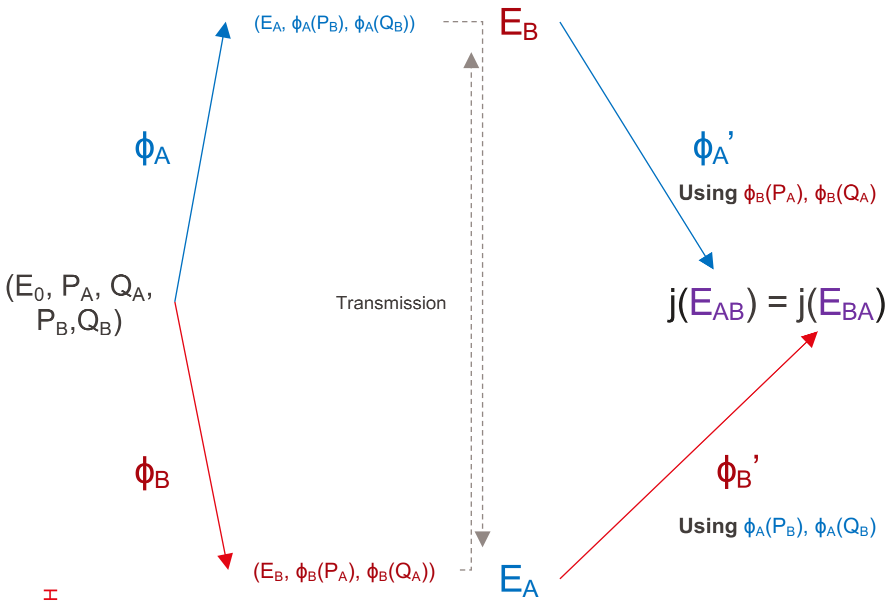
Problem: Isogenies on supersingular curves are not commutative



Supersingular Isogeny Diffie-Hellman

Making isogenies commutative:

We need to fix the kernel of the isogenies, so we need to transmit the images of the torsion points.



Castryck & Decru Attack

Attack is based on:

1. Knowledge of isogeny degrees
2. Knowledge of the images of the torsion points

EPFL

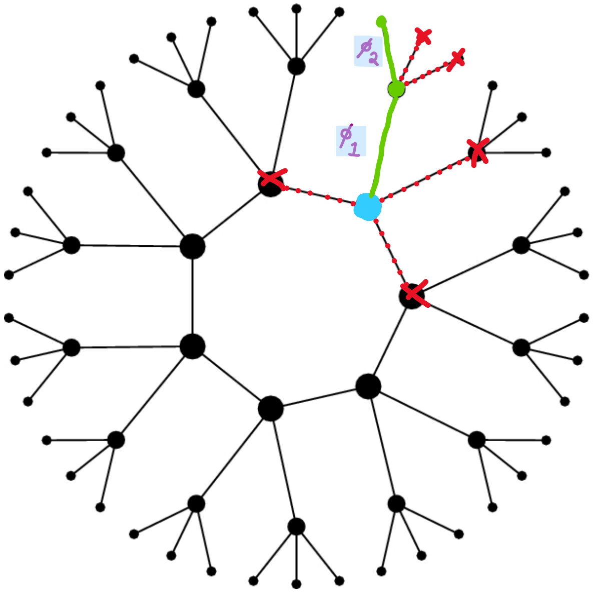
Castryck & Decru Attack

Isogeny ϕ of degree 3^k: $E \rightarrow E'$

Isogeny ϕ_i of degree 3: $E_i \rightarrow E_{i+1}$

$$\phi = \phi_1 \circ \cdots \circ \phi_k$$

An isogeny of degree n has n+1 possible co-domains



High level idea:

1. Decompose isogeny in prime steps
2. Use Kani's theorem to construct an efficient Oracle
3. Use Oracle to find correct isogeny for each step

Kani's theorem:

Theorem 1. *Let (ψ, H_1, H_2) be an isogeny diamond configuration of order $N \geq 2$ between two elliptic curves C and E. Let $d = \gcd(\#H_1, \#H_2)$, let $n = N/d$ and let $k_i = \#H_i/d$ for $i = 1, 2$. Then ψ factors uniquely over $[d]$, i.e. $\psi = \psi' \circ [d]$ and there is a unique reducible anti-isometry $\iota : C[N] \to E[N]$ such that*

$$\iota(k_1 R_1 + k_2 R_2) = \psi'(R_2 - R_1) \text{ for all } R_i \in [n]^{-1} H_i \ (i = 1, 2). \tag{3}$$

Moreover, if $N \leq p$ then every reducible anti-isometry $C[N] \to E[N]$ is of this form.

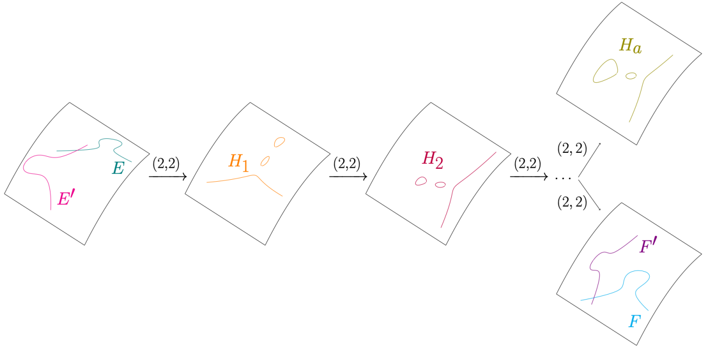


Fig.4: Decision strategy based on Kani's reducibility criterion

Countermeasures

Kani’s theorem:

- The degree A of the secret supersingular isogeny $\phi : E_0 \to E$ must be known.
- The images $\phi(P), \phi(Q)$ of the torsion basis $\langle P, Q \rangle$ of the B-torsion $E_0[B]$ (where $B > A$ and are coprime) must be known.

Option 1: Mask the isogeny degree (MD-SIDH)

Option 2: Mask the images of the torsions points (M-SIDH)

M-SIDH has smaller key sizes and parameters for the same level of security as MD-SIDH

Countermeasures

Modifications in M-SIDH

Masking the images of torsion points

Multiply the points by a random non-zero α sampled from $\mathbb{Z}/n\mathbb{Z}$ with $n = A, B$.

The exchange succeeds as the scaled points generate the same kernel.

New problem: There exists an efficient algorithm to find α^2 in groups of smooth order[7].

Therefore recovering the direct images is efficient.

Modifications in M-SIDH

Masking the images of torsion points

Multiply the points by a random α sampled from:

$$\alpha \in_R \mu_2(N) := \{x \in \mathbb{Z}/N\mathbb{Z} \text{ st. } x^2 \equiv 1 \pmod N\}$$

The exchange is shown to succeed as the scaled points generate the same kernel.

New problem: We need $|\mu_2(N)|$ to be large!

Solution: We construct A, B such that α^2 has at least 2^λ roots in $\mu_2(A)$ and $\mu_2(B)$

$$p_{128} = 2^2 \cdot \ell_1 \cdots \ell_{256} \cdot 59 - 1 \qquad p = 2^2 ABf - 1 \qquad A = \ell_1 \cdot \ell_3 \cdots \ell_{2\lambda - 1}$$

$$p_{192} = 2^2 \cdot \ell_1 \cdots \ell_{384} \cdot 102 - 1 \qquad l_i = \text{i-th prime} \qquad B = \ell_2 \cdot \ell_4 \cdots \ell_{2\lambda}$$

Malo RANZETTI

Modifications in M-SIDH

AES	NIST	p (in bits)	secret key	public key	compressed pk
128	level 1	5911	$\approx$ 369 bytes	4434 bytes	$\approx$ 2585 bytes
192	level 3	9382	$\approx$ 586 bytes	7037 bytes	$\approx$ 4103 bytes
256	level 5	13000	$\approx$ 812 bytes	9750 bytes	$\approx$ 5687 bytes

[8]

Table 1. Suggested parameters for 128, 192 and 256 bits of security.

Implementation of M-SIDH

Implementation challenges:

Naïve implementation takes many hours to compute.

- Finding generators of the A,B-torsion basis
  - Use the structure of the curve group to generate points of order p+1
  - Check the independence of points using Weil pairings
  - Adjust points until we obtain generators of the curve group
  - Use the generators to obtain torsion points
- Sampling elements of $\mu_2(N)$
  - Use the known prime factorization of A and B to uniformly and independently choose a root of unity for each prime factor (1 or p_i-1).

$$\alpha \in_R \mu_2(N) := \{x \in \mathbb{Z}/N\mathbb{Z} \text{ st. } x^2 \equiv 1 \pmod N\}$$

Implementation of M-SIDH

Bugs found along the way:

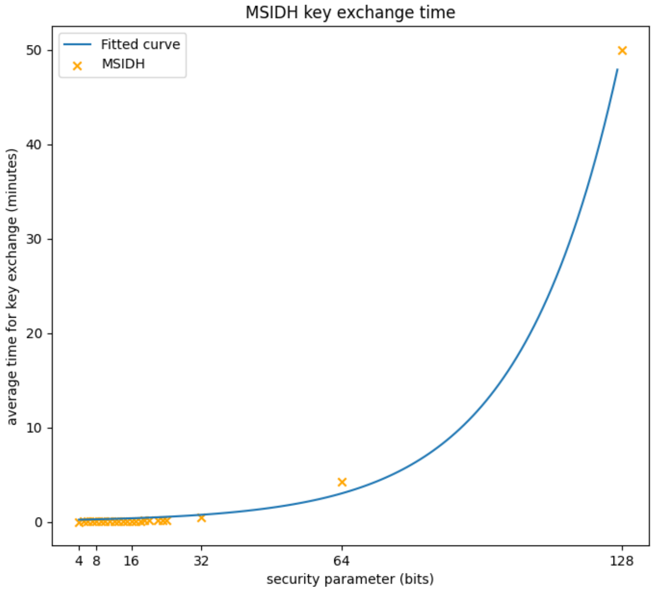
Some bugs compromised the security and efficiency

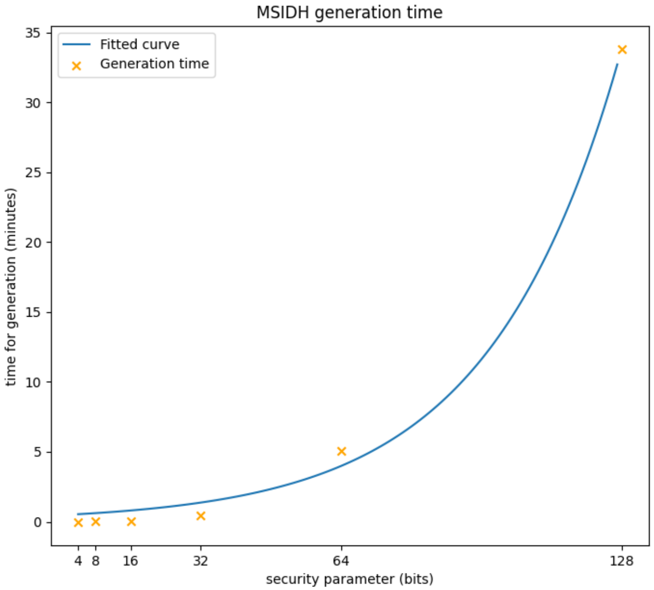
- Efficiency problem with Sagemath
  - Proofs turned off to prevent is_prime computation on large p
  - The velusqrt algorithm seems to have efficiency problem (“naïve” Vélu is faster even for large n > 3000)
  - Factored isogeny did not leverage velusqrt (arguably lucky)
  - Factored isogeny recomputes order of kernel point at every step
- Slight correction to proposed M-SIDH algorithm

“We then check the value $t - n + 1$ described in subsection 4.1. If $\lambda < t - n + 1$, we restart with a larger t”

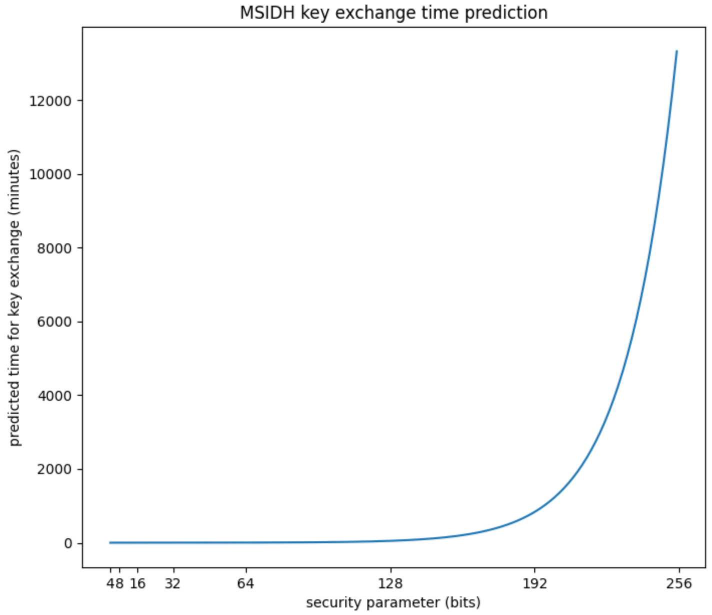
```
if security_parameter > (t - n + 1):
    # We have to restart with a larger t
    print(f"retrying with t={t+1}")
    self.__init__(security_parameter, force_t=t+1)
    return
```

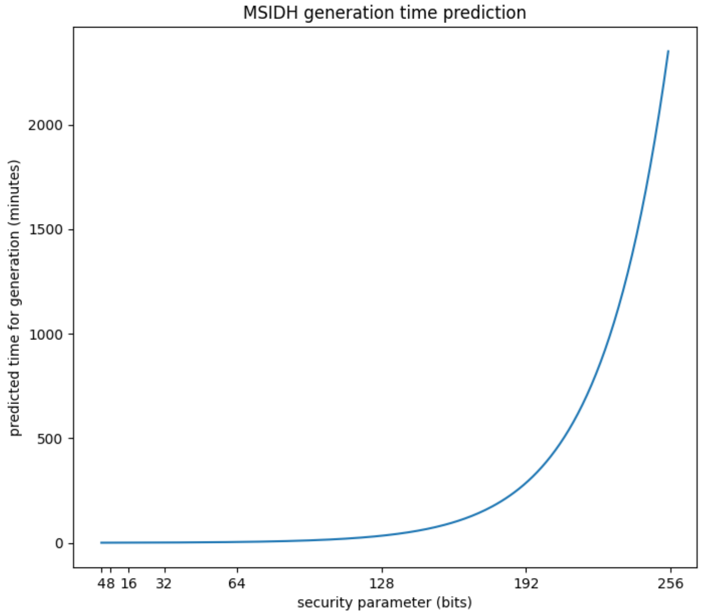
Evaluation of M-SIDH





Evaluation of M-SIDH





Evaluation of M-SIDH

M-SIDH Implementation results

λ (bits)	Key exchange time (s)
4	0.1681 ± 0.0028
8	0.8171 ± 0.013
16	4.183 ± 0.070
32	28.27 ± 0.47
64	252.5 ± 4.2
128	3000 ± 50

Reference results (same machine)

Scheme & Classical security level	Key exchange time (s)
SIDH, NIST 1	2.852
SIDH, NIST 2	4.043
SIDH, NIST 3	6.563
SIDH, NIST 5	11.26
CSIDH, NIST 1	17.63
CSIDH, NIST 5	30.02

Related works

Public key compression in M-SIDH

Kaizhan Lin, Jianming Lin, Shiping Cai, Weize Wang, and Chang-An Zhao

- Similar performance (slightly faster)
- Uses low-level optimizations (still in Sagemath)
- Not generalizable

CSIDH: An Efficient Post-Quantum Commutative Group Action

Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes

- Another proposed scheme, not affected by the Castryck & Decru attack
- Slightly slower than SIDH, but much better performance than M-SIDH

λ (bits)	Key exchange time (s)
4	0.1681 ± 0.0028
8	0.8171 ± 0.013
16	4.183 ± 0.070
32	28.27 ± 0.47
64	252.5 ± 4.2
128	3000 ± 50

Scheme & Classical security level	Key exchange time (s)
SIDH, NIST 1	2.852
SIDH, NIST 2	4.043
SIDH, NIST 3	6.563
SIDH, NIST 5	11.26
CSIDH, NIST 1	17.63
CSIDH, NIST 5	30.02

Summary of key results

Conclusion

- M-SIDH is currently suffering from performance issues
- Main bottleneck is isogeny computation
- Next step: Multi-threading, low-level implementation,TPC for curve

Rapidly developing field, but should we rather focus on other alternatives?

Bibliography

[1] Daniel J. Bernstein, Luca De Feo, Antonin Leroux, and Benjamin Smith. Faster computation of isogenies of large prime degree. Algorithmic Number Theory Symposium 2020, Document ID: 44d5ade1c1778d86a5b035ad20f880c08031a1dc. Homepage: https://velusqrt.isogeny.org/. 2020. U R L: https://velusqrt.isogeny.org/velusqrt-20200616.pdf.

[2] Fabio Campos, Jorge Chavez-Saab, Jesús-Javier Chi-Domínguez, Michael Meyer, Krijn Reijnders, Francisco Rodríguez-Henríquez, Peter Schwabe, and Thom Wiggers. On the Practicality of Post-Quantum TLS Using Large-Parameter CSIDH. Cryptology ePrint Archive, Paper 2023/793. https://eprint.iacr.org/2023/793. 2023. U R L: https://eprint.iacr.org/2023/793.

[3] Wouter Castryck and Thomas Decru. An efficient key recovery attack on SIDH. Cryptology ePrint Archive, Paper 2022/975. 2022. U R L: https://eprint.iacr.org/2022/975.

[4] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: An Efficient Post-Quantum Commutative Group Action. Cryptology ePrint Archive, Paper 2018/383. https://eprint.iacr.org/2018/383. 2018. U R L: https://eprint.iacr.org/2018/383.

[5] Craig Costello, Patrick Longa, and Michael Naehrig. Efficient algorithms for supersingular isogeny Diffie-Hellman. Cryptology ePrint Archive, Paper 2016/413. 2016. U R L: https://eprint.iacr.org/2016/413.

[6] Luca De Feo, David Jao, and Jérôme Plût. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. Cryptology ePrint Archive, Paper 2011/506. 2011. U R L: https://eprint.iacr.org/2011/506.

[7] Tako Boris Fouotsa. SIDH with masked torsion point images. Cryptology ePrint Archive, Paper 2022/1054. 2022. U R L: https://eprint.iacr.org/2022/1054.

[8] Tako Boris Fouotsa, Tomoki Moriya, and Christophe Petit. M-SIDH and MD-SIDH: countering SIDH attacks by masking information. Cryptology ePrint Archive, Paper 2023/013. 2023. U R L: https://eprint.iacr.org/2023/013.

[9] Gora Adj, Jesús-Javier Chi-Domínguez, and Francisco Rodríguez-Henríquez. SIBC Python library. 2021. U R L: https://github.com/JJChiDguez/sibc/.

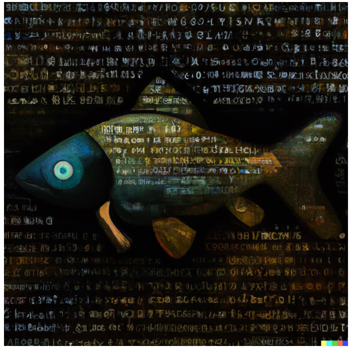
[10] Kaizhan Lin, Jianming Lin, Shiping Cai, Weize Wang, and Chang-An Zhao. Public-key Compression in M-SIDH. Cryptology ePrint Archive, Paper 2023/136. 2023. U R L: https://eprint.iacr.org/2023/136.

[11] University of Duisburg-EssenSVG version: Flugaal Original schema: A.J.jacquie MacLaine/Vinck. File:Diffie-Hellman Key Exchange.svg - Wikimedia Commons. 2020. U R L: https://commons.wikimedia.org/wiki/File:Diffie-Hellman_Key_Exchange.svg.

[12] Damien Robert. Breaking SIDH in polynomial time. Cryptology ePrint Archive, Paper 2022/1038.2022. U R L: https://eprint.iacr.org/2022/1038.

[13] The Sage Developers. SageMath, the Sage Mathematics Software System (Version 9.8). 2023.U R L: https://www.sagemath.org.

Monkeys doing Elliptic Curve Cryptography



Cryptographic fish (Rembrandt)

Fig.4: Generated illustrations [DALL-E-2]

Questions?