


Applying Castryck-Decru Attack on the Masked Torsion Point Images SIDH variant

Jesús-Javier Chi-Domínguez¹ 

Cryptography Research Centre, Technology Innovation Institute, Abu Dhabi, UAE
jesus.dominguez@tii.ae

Keywords: Cryptanalysis · Castryck-Decru Attack · Isogeny-based cryptography · Masked-SIDH

Abstract. This paper illustrates that masking the torsion point images does not guarantee Castryck-Decru attack does not apply. Our experiments over SIDH primes hint that any square root concerning the Weil pairing on the masked public key helps to recover Bob’s private key via the Castryck-Decru attack.

1 Introduction

Castryck and Decru provided in [2] a heuristically polynomial SIDH key-recovery Attack, which relies on the knowledge of

- The isogeny degree;
- The image of coprime torsion points; and
- The endomorphism ring of the isogeny domain curve.

Maino and Martindale in [7] gave an algorithm that works without knowing the endomorphism ring of the domain curve. In contrast, Robert demonstrated the existence of a polynomial key-recovery attack on SIDH [10]. The results in [7,10] are still theoretical, but [2] shared a Magma code of their attack improved by the `sagemath` code of Oudompheng and Pope in [9].

To mitigate the Castryck-Decru attack, Fouotsa and Moriya independently proposed solutions for SIDH. Fouotsa suggested masking the torsion point images [4] and Moriya hiding the isogeny degree [8]. This works only analyze Fouotsa’s countermeasure given in [4].

2 SIDH framework

We strongly recommend that the readers go through [5,3,1] and [2] for details concerning SIDH and the Castryck-Decru attack. Let us first center on the following SIDH setup. Let \mathbb{F}_{p^2} be a quadratic field extension of \mathbb{F}_p along with $p = 2^a 3^b - 1$. We set as starting supersingular curve $E_0: y^2 = x^3 + 6x^2 + x$ ¹.

¹ We choose the same E_0 as in SIKE proposal [1], but it can be any different curve with known endomorphism ring.

Let $\{P_A, Q_A\}$ a basis for the 2^a -torsion subgroup $E_0[2^a]$, and $\{P_B, Q_B\}$ for the 3^b -torsion subgroup $E_0[3^b] = \langle P_B, Q_B \rangle$.

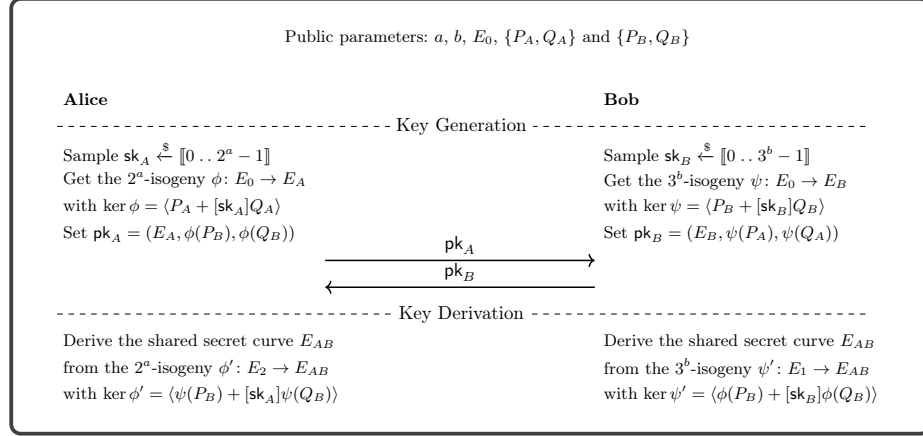


Fig. 1: General description of the SIDH protocol.

3 The masked SIDH from [4]

The countermeasure from [4] proposes masking the torsion point images in pk_A and pk_B as

- Alice samples a random integer $r_A \in \llbracket 0 \dots 3^b - 1 \rrbracket$ coprime to 3. Next, she sets a her public key $pk_A = (E_A, [r_A]\phi(P_B), [r_A]\phi(Q_B))$.
- Bobs samples a random integer $r_B \in \llbracket 0 \dots 2^a - 1 \rrbracket$ coprime to 2. Next, he sets a his public key $pk_B = (E_B, [r_B]\psi(P_A), [r_B]\psi(Q_A))$.

[4] claims the above masking proposal is enough for the Castryck-Decru attack to fail; we need precisely the image of the torsion points to mount the Castryck-Decru attack. In concrete, an attacker can get both $(r_A)^2 \bmod 3^b$ and $(r_B)^2 \bmod 2^b$ via discrete logarithms concerning the following Weil pairing equations

$$e_{3^b}([r_A]\phi(P_B), [r_A]\phi(Q_B)) = \left(e(P_B, Q_B)^{2^a} \right)^{(r_A)^2} \text{ and}$$

$$e_{2^a}([r_B]\psi(P_A), [r_B]\psi(Q_A)) = \left(e(P_A, Q_A)^{3^b} \right)^{(r_B)^2}.$$

On that basis, [4] suggests a large prime such that the number of square roots for $(r_A)^2$ and $(r_B)^2$ is about 2^λ , thus finding the correct r_A and r_B ensure λ -bits of security.

4 Applying Castryck-Decru Attack

For simplicity, we center on analyzing Bob's public key (just as in [2]), but it easily extends to Alice's scenario. This section justifies and experimentally illustrates that any square root of $r = (r_B)^2$ helps to make the Castryck-Decru works.

Let r' be a square root of r , and let $\vartheta_{r'} : P \mapsto [\tilde{r}r_B]\psi(P)$ be the isogeny being \tilde{r} the multiplicative inverse of r' modulo 2^a . Next, we let $\hat{\vartheta}_{r'} : P \mapsto [\tilde{r}r_B]\hat{\psi}(P)$ describes the dual isogeny of $\vartheta_{r'}$. Consequently, we get $\vartheta_{r'} \circ \hat{\vartheta}_{r'} = [3^b(\tilde{r}r_B)^2]$ and $\hat{\vartheta}_{r'} \circ \vartheta_{r'} = [3^b(\tilde{r}r_B)^2]$.

Lemma 1. *The isogeny $\vartheta_{r'}$ and its dual $\hat{\vartheta}_{r'}$ satisfy $\vartheta_{r'} \circ \hat{\vartheta}_{r'} = [3^b] = \hat{\vartheta}_{r'} \circ \vartheta_{r'}$.*

Proof. Let E be either E_0 or E_B . Since we analyze curves with $E(\mathbb{F}_{p^2}) \cong \mathbb{Z}_{p+1} \times \mathbb{Z}_{p+1}$, we only need to prove the equality over the 2^a -torsion and 3^b -torsion subgroups (any point in $E(\mathbb{F}_{p^2})$ splits as the sum of order- 2^a and order- 3^b points).

If P be an order- 3^b point on E , $[3^b(\tilde{r}r_B)^2]P = \mathcal{O} = [3^b]P$ and then the equality holds. If P be an order- 2^a point on E , we have $(\tilde{r}r_B)^2 = 1 \pmod{2^a}$ and thus $[3^b(\tilde{r}r_B)^2]P = [3^b]P$. \square

Lemma 1 implicitly says $\vartheta_{r'} : E_0 \rightarrow E_B$ and its dual $\hat{\vartheta}_{r'} : E_B \rightarrow E_0$ looks like 3^b -isogenies over \mathbb{F}_{p^2} . We point out that $\vartheta_{r'}$ and $\hat{\vartheta}_{r'}$ does not behave as 3^b -isogenies over extensions fields of \mathbb{F}_{p^2} but we do not care about that for attacking the masked SIDH construction. In particular, locally over \mathbb{F}_{p^2} , we get a high chance that the Castryck-Decru attack succeeds for $\vartheta_{r'}$.

Now we sketch how to recover Bob's secret 3^b -isogeny $\psi : E_0 \rightarrow E_B$. Given the masked public key $\mathbf{pk}_B = (E_B, [r_B]\psi(P_A), [r_B]\psi(Q_A))$, we proceed as follows:

1. Parse $(E_B, P', Q') \leftarrow \mathbf{pk}_B$;
2. Get $r = (r_B)^2$ from the Weil pairing equation in Section 3;
3. Compute any square root r' of r ;
4. Calculate the multiplicative inverse \tilde{r} of r' modulo 2^a ;
5. Set $\mathbf{pk}'_B = (E_B, [\tilde{r}]P', [\tilde{r}]Q')$;
6. Feed the Castryck-Decru attack with \mathbf{pk}'_B to find the isogeny $\vartheta_{r'} : P \mapsto [\tilde{r}r_B]\psi(P)$.
7. Derive \mathbf{sk}_B from $\vartheta_{r'}$.

We validate the above procedure using the sagemath code from [9] along with the below code ² and illustrate that we do not need the correct square root r_B of $r = (r_B)^2$.

² We take the script `baby_SIDH.sage` from [9] as a baseline and replace it according to SIKE parameters.

```

import public_values_aux
from public_values_aux import *

load('castryck_decru_shortcut.sage')

# SIKEpXXX parameters
# a, b = 33, 19
# a, b = 191, 117
a, b = 273, 172

# Set the prime, finite fields and starting curve
# with known endomorphism
p = 2^a*3^b - 1
public_values_aux.p = p

Fp2.<i> = GF(p^2, modulus=x^2+1)
R.<x> = PolynomialRing(Fp2)

E_start = EllipticCurve(Fp2, [0,6,0,1,0])
E_start.set_order((p+1)^2) # Speeds things up in Sage

# Generation of the endomorphism 2i
two_i = generate_distortion_map(E_start)

# Generate public torsion points, for SIKE implementations
# these are fixed but to save loading in constants we can
# just generate them on the fly
P2, Q2, P3, Q3 = generate_torsion_points(E_start, a, b)
check_torsion_points(E_start, a, b, P2, Q2, P3, Q3)

# Generate Bob's key pair
bob_private_key, EB, PB, QB = gen_bob_keypair(E_start, b, P2, Q2, P3, Q3)
solution = Integer(bob_private_key).digits(base=3)

print(f"Running the attack against SIDHp{p.bit_length()} parameters,
→ which has a prime: 2^{a}*3^{b} - 1")
print(f"If all goes well then the following digits should be found:
→ {solution}")

def mask(pk):
    (EB, PB, QB) = pk
    N = 2^a
    rB = 2 * randint(1, N // 2) + 1
    print(f'mask:\t{rB}')
    return EB, rB * PB, rB * QB

# =====
# ===== ATTACK =====
# =====
def unmask(pk):

```

```

(EB, PB, QB) = pk
e = P2.weil_pairing(Q2, 2^a)
e_ = PB.weil_pairing(QB, 2^a)
N = e.order()
r = discrete_log(e_, e^(3^b))
assert (e ^ (r * (3^b))) == e_
assert N == e_.order()
N = 2^a
R = IntegerModRing(N)
square_roots = R(r).sqrt(all=True)
other = int(square_roots[0])
dec, tilde, _ = xgcd(other, N)
assert dec == 1
tilde = int(R(tilde))
assert tilde * other % (N) == 1
PB_ = tilde * PB
QB_ = tilde * QB
assert e^(3^b) == (PB_).weil_pairing(QB_, 2^a)
print(f'unmask:\t{other}')
return EB, PB_, QB_

def RunAttack(num_cores):
    return CastryckDecruAttack(E_start, P2, Q2, EB, PB, QB, two_i,
        ↪ num_cores=num_cores)

EB, PB, QB = mask((EB, PB, QB))
EB, PB, QB = unmask((EB, PB, QB))

if __name__ == '__main__' and '__file__' in globals():
    if '--parallel' in sys.argv:
        # Set number of cores for parallel computation
        num_cores = os.cpu_count()
        print(f"Performing the attack in parallel using {num_cores}
            ↪ cores")
    else:
        num_cores = 1
    recovered_key = RunAttack(num_cores)

```

Our experiments randomly generate instances with the following SIDH parameters:

- Baby SIDHp64 from [9]: $a = 33$ and $b = 19$,
- \$IKEp217: $a = 110$ and $b = 67$;
- SIKEp377 from [6]: $a = 191$ and $b = 117$;
- SIKEp546 from [6]: $a = 273$ and $b = 172$;

Remark 1. If we replace the prime $p = 2^a 3^b - 1$ by $p = 4AB - 1$ with $(A, B) = 1$, and the isogeny degrees 2^a and 3^b by A and B , respectively. Then, a similar

reasoning from above holds. We can use the Castryck-Decru attack to recover the private key sk_B from the proposal in [4]; this time, we should find the isogeny $\vartheta_{r'} : P \mapsto [\tilde{r}r_B]\psi(P)$ with $\tilde{r}r' = 1 \pmod A$ for any square root r' of $(r_B)^2$ modulo A .

Open questions: Could the above attack work for primes $p = 4AB - 1$? Or do we need exactly the square root r_B ?

Acknowledgements We thank Benjamin Wesolowski, Luca De Feo, and Peter Kutas for their comments and discussion on the applicability of masked SIDH parameters with P.

References

1. Azarderakhsh, R., Campagna, M., Costello, C., De Feo, L., Hess, B., Jalali, A., Jao, D., Koziel, B., LaMacchia, B., Longa, P., Naehrig, M., Pereira, G., Renes, J., Soukharev, V., Urbanik, D.: Supersingular Isogeny Key Encapsulation. Third Round Candidate of the NIST’s post-quantum cryptography standardization process (2020), available at: <https://sike.org/>
2. Castryck, W., Decru, T.: An efficient key recovery attack on SIDH (preliminary version). IACR Cryptol. ePrint Arch. p. 975 (2022), <https://eprint.iacr.org/2022/975>
3. De Feo, L., Jao, D., Plüt, J.: Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. *Journal of Mathematical Cryptology* **8**(3), 209–247 (2014). <https://doi.org/10.1515/jmc-2012-0015>, <https://doi.org/10.1515/jmc-2012-0015>
4. Fouotsa, T.B.: SIDH with masked torsion point images (2022), <https://eprint.iacr.org/2022/1054>
5. Jao, D., De Feo, L.: Towards Quantum-Resistant Cryptosystems from Supersingular Elliptic Curve Isogenies. In: Yang, B. (ed.) *Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011, Taipei, Taiwan, November 29 - December 2, 2011. Proceedings. Lecture Notes in Computer Science*, vol. 7071, pp. 19–34. Springer (2011). https://doi.org/10.1007/978-3-642-25405-5_2, https://doi.org/10.1007/978-3-642-25405-5_2
6. Longa, P., Wang, W., Szefer, J.: The Cost to Break SIKE: A Comparative Hardware-Based Analysis with AES and SHA-3. In: Malkin, T., Peikert, C. (eds.) *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16–20, 2021, Proceedings, Part III. Lecture Notes in Computer Science*, vol. 12827, pp. 402–431. Springer (2021). https://doi.org/10.1007/978-3-030-84252-9_14, https://doi.org/10.1007/978-3-030-84252-9_14
7. Maino, L., Martindale, C.: An attack on SIDH with arbitrary starting curve. IACR Cryptol. ePrint Arch. p. 1026 (2022), <https://eprint.iacr.org/2022/1026>
8. Moriya, T.: Masked-degree SIDH (2022), <https://eprint.iacr.org/2022/1019>
9. Oudompheng, R., Pope, G.: A Note on Reimplementing the Castryck-Decru Attack and Lessons Learned for SageMath (2022), <https://eprint.iacr.org/2022/1283>
10. Robert, D.: Breaking SIDH in polynomial time. IACR Cryptol. ePrint Arch. p. 1038 (2022), <https://eprint.iacr.org/2022/1038>