Cryptanalysis of Isogeny-Based Quantum Money with Rational Points

Hyeonhak Kim[1], Donghoe Heo[1] and Seokhie Hong[1]*

School of Cybersecurity, Korea University, Seoul 02841, South Korea

Keywords: Quantum Money, Quantum Lightning, Class Group Action and Elliptic Curve

Abstract. Quantum money is the cryptographic application of the quantum no-cloning theorem. It has recently been instantiated by Montgomery and Sharif (Asiacrypt '24) from class group actions on elliptic curves. In this work, we propose a concrete cryptanalysis by leveraging the efficiency of evaluating division polynomials with the coordinates of rational points, offering a speedup of $O(\log^4 p)$ compared to the brute-force attack. Since our attack still requires exponential time, it remains impractical to forge a quantum banknote. Interestingly, due to the inherent properties of quantum money, our attack method also results in a more efficient verification procedure. Our algorithm leverages the properties of quadratic twists to utilize rational points in verifying the cardinality of the superposition of elliptic curves. We expect this approach to contribute to future research on elliptic-curve-based quantum cryptography.

1 Introduction

Extensive research has been carried out to harness quantum advantage. Using a quantum computer, it is well known that the discrete logarithm problem on which many modern cryptosystems are based is easily breakable. Most of the research on quantum algorithms focuses on their efficiency in solving a classically intractable problem. Recently, there has been a growing interest in utilizing quantum computers for security purposes, notably through quantum money. Quantum money was first proposed by Wiesner in [18]. The no-cloning theorem prevents adversaries from counterfeiting the money. However, Wiesner's scheme was privately verifiable, which means that only the issuing authority (the mint) can verify the money. In [1], the publicly verifiable quantum money scheme was devised where anyone can verify the money.

The first instantiation of publicly verifiable quantum money was [2], but it was later broken by [7]. Zhandry subsequently proposed a construction of quantum money/lightning based on the assumption of indistinguishability obfuscation in [19], which was later broken by Roberts [13]. More recently, Liu,

* Corresponding author.

Montgomery and Zhandry devised a construction of quantum money which is called *walkable invariant money* in [10]. This walkable invariant money was concretely instantiated in [11] using class group actions on elliptic curves.

The instantiation of [11] uses the cardinality of elliptic curves as a serial number of a banknote. We propose a new way to forge a quantum banknote from the serial number given, more optimal than brute-force attack. We leverage the fact that computing division polynomials with rational points is more efficient than the point-counting algorithm. For an elliptic curve E defined over $\mathbb{F}_p$, a division polynomial $\psi_\ell(X, Y) \in \mathbb{F}_p[X, Y]$ is a polynomial whose roots are the ℓ-torsion points $(x, y) \in E[\ell]$. Using the recurrence relation of division polynomials, we can compute $\psi_\ell(x, y)$ in $O(\log \ell)$ multiplications in $\mathbb{F}_p$, which is significantly faster than the point-counting algorithm when p is a large prime.

1.1 Contributions

In this work, we analyze the unforgeability of the quantum money scheme proposed in [11] in a concrete manner. We identify two potential approaches to forge a quantum banknote in this scheme.

The first approach involves constructing a uniform superposition of elliptic curves with the given cardinality using the quantum random walk. In order to use the quantum random walk-based method, the index used in generating the quantum state must be discarded, which is known as the Index Erasure Problem. In Section 3, we show that Kuperberg's algorithm [9] is infeasible for solve this problem in our setting and demonstrate that the quantum random walk-based attack is significantly inefficient.

The second approach leverages quantum search techniques to sample elliptic curves of the required cardinality. A straightforward yet computationally expensive method would be to employ the point-counting algorithm as a search oracle. Instead, we propose a more efficient alternative that takes advantage of the fact that verifying the order of rational points is significantly faster than counting rational points.

Furthermore, we note the intrinsic connection between forging a quantum banknote and the verification process. By exploiting this relationship, we outline how our proposed optimization can be applied not only to an attack strategy but also to an improved verification method. Our main contributions are summarized as follows:

- **Optimized Forgery Attack** : We propose a novel attack strategy that efficiently searches for a quantum banknote with a given serial number, outperforming direct brute-force approaches. Our attack method $O(\log^4 p)$ times faster than the brute-force attack and requires $12\lceil \log p \rceil^2$ qubits, which is $\log p$ times fewer than the brute-force method.
- **Concrete Security Estimation** : As our approach is significantly simpler than implementing the point-counting algorithm, we can provide a more precise estimation of the quantum resources required to forge quantum money. We offer a detailed analysis of the quantum resources. The result is shown in Table 6.3.

- **Improved Verification** : We demonstrate how our insights can improve the efficiency of the verification of a given serial number. The process of checking a serial number becomes $O(\log^4 p)$ times faster than the original approach.
- **Utilization of Rational Points** : We show that rational points can be used to verify the cardinality based on the properties of the quadratic twists of elliptic curves. We expect that our method inspires future research.

1.2 Organization of The Paper

In Section 2, we introduce the background on elliptic curves and the notions of quantum money and quantum lightning. We also briefly present the quantum lightning scheme proposed in [11]. In Section 3, we demonstrate that Kuperberg's algorithm is infeasible in our case and that our attack method provides the more optimized than the quantum random walk-based attack. Section 4 illustrates our quantum search algorithm and its oracle. In Section 5, we show that the lower and upper bound of class numbers that leads to the number of iterations in the quantum search algorithm. In Section 6, our concrete quantum oracle is described and we analyzed its space and time complexity. In Section 7, we apply our attack method to the verification algorithm. Finally, we conclude our work in Section 8.

2 Preliminaries

2.1 Class Group Actions on Elliptic Curves

Elliptic Curves For a large prime p, an elliptic curve $E/\mathbb{F}_p$ is defined as follows in Weierstrass form:

$$E : y^2 = x^3 + Ax + B$$

where $A, B \in \mathbb{F}_p$. The isomorphism class of elliptic curves can be represented as its j-invariant $j(E) = 1728\frac{4a^3}{4a^3+27b^2}$. Two elliptic curves E_1, E_2 are isomorphic over the algebraically closed field $\bar{\mathbb{F}}_p$ if and only if $j(E_1) = j(E_2)$, which means that j-invariant uniquely represents the $\bar{\mathbb{F}}_p$-isomorphism classes of elliptic curves. When we consider $\mathbb{F}_p$-isomorphism classes of elliptic curves, we need additional information to uniquely identify the classes. We introduce the representation used in [11]. The $\mathbb{F}_p$-isomorphism classes of elliptic curves can be represented by pairs $(j, b) \in \mathbb{F}_p \times \mathbb{Z}$ where $b \in \{0, 1\}$ except in the following cases:

- If $j \equiv 1728 \mod p$ and $p \equiv 1 \mod 4$, then $0 \leq b \leq 3$.
- If $j \equiv 0 \mod p$ and $p \equiv 1 \mod 3$, then $0 \leq b \leq 5$.

For a quadratic non-residue $\alpha_2 \in \mathbb{F}_p$, we can recover the Weierstrass pair $(A, B) \in \mathbb{F}_p$ from the given pair (j, b) as follows:
If $j \not\equiv 1728, 0 \mod p$,

$$y^2 = x^3 + \frac{3j\alpha_2^{2b}}{1728 - j}x + \frac{2j\alpha_2^{3b}}{1728 - j}$$

If $j \equiv 1728 \mod p$, the elliptic curve is given by

$$y^2 = x^3 + \alpha_4^b x$$

where α_4 is a quartic non-residue if $p \equiv 1 \mod 4$, a quadratic non-residue otherwise.
If $j \equiv 0 \mod p$,

$$y^2 = x^3 + \alpha_6^b$$

where α_6 is a sextic non-residue if $p \equiv 1 \mod 3$, a quadratic non-residue otherwise.

By [16, Cor. X.5.4.1], there is one-to-one correspondence between pairs (j, b) and $\mathbb{F}_p$-isomorphism classes of elliptic curves. While the (j, b) representation is used in the quantum money scheme [11], we mainly use the Weierstrass representation (A, B) in this paper. Unlike j-invariant form, Weierstrass pairs are directly adaptable to the quantum search algorithm. Throughout Sections 4 and 6, we change the j-invariant form into the Weierstrass form. Algorithm 1 demonstrates the corresponding quantum algorithm. Algorithm 1 requires just a few multiplications in $\mathbb{F}_p$.

Algorithm 1 Algorithm $\mathsf{GetWeierstrassPair}_p$

Input: A prime p and a quantum state $|j\rangle\,|b\rangle$ where $(j, b) \in \mathbb{F}_p \times \mathbb{Z}$ and $0 \le b \le 5$.
Output: A quantum state $|A\rangle\,|B\rangle$.

if $p \equiv 1 \mod 12$ **then**
  Set α_2, α_4 and α_6 as quadratic, quartic and sextic non-residue respectively.
else if $p \equiv 1 \mod 4$ and $p \not\equiv 1 \mod 3$ **then**
  Set α_2 and α_6 as quadratic non-residues and α_4 as a quartic non-residue.
else if $p \equiv 1 \mod 3$ and $p \not\equiv 1 \mod 4$ **then**
  Set α_2 and α_4 as quadratic non-residues and α_6 as a sextic non-residue.
else
  Set all α_2, α_4 and α_6 as quadratic non-residues.
end if
Compute $|A\rangle \leftarrow |(j \equiv 1728) \times \alpha_4^b + (j \not\equiv 0, 1728) \times \left(\frac{3j\alpha_2^{2b}}{1728-j}\right) \mod p\rangle$.
Compute $|B\rangle \leftarrow |(j \equiv 0) \times \alpha_6^b + (j \not\equiv 0, 1728) \times \left(\frac{2j\alpha_2^{3b}}{1728-j}\right) \mod p\rangle$.
return $|A\rangle\,|B\rangle$.

We also note that the group structure of an elliptic curve defined over $\mathbb{F}_p$ is $\mathbb{Z}/m\mathbb{Z} \times \mathbb{Z}/mk\mathbb{Z}$.

Theorem 1. *For a prime p, the cardinality of an elliptic curve defined over $\mathbb{F}_p$ is isomorphic to*

$$\mathbb{Z}/m\mathbb{Z} \times \mathbb{Z}/mk\mathbb{Z}$$

where $m, k \in \mathbb{Z}^+$ and $m|(p-1)$.

Proof. We refer to the proof in [14].

We note that the number of points of elliptic curves is bounded by well-known Theorem 2. We use this property in Section 6 to prove the correctness and the soundness of our algorithm.

Theorem 2 (Hasse's theorem). *Let E be an elliptic curve defined over a finite field $\mathbb{F}_p$. Then*

$$|\#E(\mathbb{F}_p) - p - 1| \leq 2\sqrt{p}$$

Proof. We refer to the proof in [16, Theorem 1.1].

Quadratic Twist of Elliptic Curves Given an elliptic curve $E : y^2 = x^3 + Ax + B$ defined over $\mathbb{F}_p$, a quadratic twist of E is defined as $E^t : y^2 = x^3 + \alpha^{-2}Ax + \alpha^{-3}B$ where α is a quadratic non-residue in $\mathbb{F}_p$. There is an isomorphism $\phi : E \to E^t$ defined over $\mathbb{F}_{p^2}$ such that $\phi(x, y) = (\alpha^{-1}x, \alpha^{-3/2}y)$. Note that E and E^t are not isomorphic over $\mathbb{F}_p$.

Theorem 3. *Given an elliptic curve $E : y^2 = x^3 + Ax + B$ over $\mathbb{F}_p$ and its quadratic twist E^t of E, they satisfy the following equation.*

$$\#E(\mathbb{F}_p) + \#E^t(\mathbb{F}_p) = 2p + 2.$$

Proof. For an $x \in \mathbb{F}_p$, if x^3+Ax+B is a quadratic residue over $\mathbb{F}_p$, there exist $\pm y$ such that $y^2 = x^3+Ax+B$, which means $(x, \pm y) \in E(\mathbb{F}_p)$. Otherwise, there exist $\pm y$ such that $\alpha y^2 = x^3 + Ax + B$ for a quadratic non-residue α, which means $(\alpha^{-1}x, \pm\alpha^{-1}y) \in E^t(\mathbb{F}_p)$. When $y = 0$, $(x, 0) \in E(\mathbb{F}_p)$ and $(x, 0) \in E^t(\mathbb{F}_p)$. It says that the sum of the sizes of two groups $E(\mathbb{F}_p)$ and $E^t(\mathbb{F}_p)$ equals to $2(|\mathbb{F}_p| + 1) = 2p + 2$ considering the point at infinity 0_E and 0_{E^t}. □

Class Group Actions For an elliptic curve E, we say that E has complex multiplication by $\mathcal{O}$ if $\mathrm{End}(E) \cong \mathcal{O}$ where $\mathcal{O}$ is an order of an imaginary quadratic number field K.

According to Deuring correspondence, an element $\alpha \in \mathcal{O}$ corresponds to an endomorphism $\theta_\alpha \in \mathrm{End}(E)$ and an integral ideal $I \subset \mathcal{O}$ corresponds to an isogeny $\phi_I : E \to E_I := E/E[I]$ where $E[I] = \{\cap \ker(\theta_\alpha) : \alpha \in I\}$.

Theorem 4. *Let E be an elliptic curve over $\mathbb{F}_p$ with complex multiplication $\mathcal{O}$, and let $\mathfrak{l} \subset \mathcal{O}$ be a prime ideal of norm ℓ, where ℓ is a rational prime. Then there is a classical algorithm which computes the isogeny $\varphi_\mathfrak{l}$ in time complexity $O(\ell M(p) \log \ell \log\log \ell \log p)$ where $M(p)$ is the complexity of a multiplication in $\mathbb{F}_p$.*

We refer to [8] and [11] for the proof and the concrete implementation as quantum algorithm.

2.2 Quantum Money and Quantum Lightning

In this section, we define public key quantum money and quantum lightning. Both quantum money and quantum lightning consist of two functions Gen and Ver as follows:

- $\mathsf{Gen}(1^\lambda)$. Takes as input a security paramter λ and generate a quantum money state $|\psi\rangle$ and the associated serial number σ.
- $\mathsf{Ver}(\sigma, |\psi\rangle)$. Takes as input a pair of a serial number σ and a supposed quantum money $|\psi\rangle$, verify them.

Definition 1 (Quantum Money Unforgeability). *($\mathsf{Gen}, \mathsf{Ver}$) is secure quantum money if, for all quantum polynomial-time adversary A, it is negligible probability that A wins the following game :*

- *The challenger runs $(\sigma, |\psi\rangle) \leftarrow \mathsf{Gen}(1^\lambda)$ and give $\sigma, |\psi\rangle$ to A.*
- *A produces and sends to the challenger two supposed quantum money $|\psi_1\rangle$ and $|\psi_2\rangle$.*
- *The challenger runs $b_1 \leftarrow \mathsf{Ver}(\sigma, |\psi_1\rangle)$ and $b_2 \leftarrow \mathsf{Ver}(\sigma, |\psi_2\rangle)$. If $b_1 = b_2 = 1$, A wins.*

Definition 2 (Quantum Lightning Unforgeability). *($\mathsf{Gen}, \mathsf{Ver}$) is secure quantum lightning if, for all quantum polynomial-time adversary A, it is negligible probability that A wins the following game :*

- *A, on input 1^λ, produces and sends to the challenger a serial number σ and supposed quantum money $|\psi_1\rangle$ and $|\psi_2\rangle$.*
- *The challenger runs $b_1 \leftarrow \mathsf{Ver}(\sigma, |\psi_1\rangle)$ and $b_2 \leftarrow \mathsf{Ver}(\sigma, |\psi_2\rangle)$. If $b_1 = b_2 = 1$, A wins.*

2.3 Quantum Money from Class Group Actions

In this section, we briefly introduce the construction of quantum lightning using class group actions on elliptic curves, which is proposed in [11]. Gen uses $\mathsf{ECSupGen}$ as a subroutine which makes a uniform superposition of elliptic curves over $\mathbb{F}_p$. Ver uses $\mathsf{ECSupVer}$ as a subroutine which verify the given serial number and uniformity of the superposition of the supposed quantum money. The isogeny computation uses SEA isogeny algorithm in Theorem 4.

Algorithm 2 Algorithm $\mathsf{ECSupGen}$

Input: p a prime
Output: $|E\rangle$ a quantum state

Let $\mathcal{S}$ be a register that can store a pair (j, b), where $j \in \mathbb{F}_p$ and $0 \leq b \leq 5$.
Generate a uniform superposition $|\psi\rangle \in \mathcal{S}$ over all pairs (j, b), where
- If $j \not\equiv 0, 1728 \mod p$, then $b = 0$ or 1.
- If $j \equiv 1728$ and $p \equiv 1 \mod 4$, then $0 \leq b \leq 3$. If $p \equiv 3 \mod 4$, then $b = 0$ or 1.
- If $j \equiv 0$ and $p \equiv 1 \mod 3$, then $0 \leq b \leq 5$. If $p \equiv 2 \mod 3$, then $b = 0$ or 1.

In this scheme, a quantum banknote $|\psi\rangle$ is a uniform superposition of the set of elliptic curves such that all curves have the same cardinality σ. The cardinality σ is used as the serial number of a quantum banknote. In Algorithm 3, it counts the number of rational points of each elliptic curves in the quantum state. The point-counting algorithm is Schoof's algorithm in [15].

Algorithm 3 Algorithm Gen

Input: p a prime

Output: $|\psi\rangle$ a quantum state and an associated serial number $\sigma \in \mathbb{Z}$.

1: Compute a superposition $\sum |j, b\rangle$ over all elliptic curves over $\mathbb{F}_p$ using Algorithm 2.
2: Use Schoof's point-counting algorithm to compute the cardinality in $\sum |j, b\rangle$ in superposition and obtain $\sum |j, b\rangle |\#E_{j,b}\rangle$
3: Measure the last register and obtain σ. Compute $\Delta_{\mathrm{Fr}}(E) = 4p-(\sigma-p-1)^2$ and set a third register to be 1 if $\Delta_{\mathrm{Fr}}(E)$ is square-free and $\Delta_{\mathrm{Fr}}(E) > 3p$, and 0 otherwise. Measure the third register; if the result is 0, start over at step 1.
4: **return** The quantum state $\sum_{\#E_{j,b}=\sigma} |j, b\rangle$ and the associated serial number σ.

In Algorithm 4, it also uses Schoof's point-counting algorithm to compute the associated serial number. The uniformity of the quantum money is verified by applying all the possible isogenies and checking if the quantum state is still the same. In order to compute an isogeny with the superposition of elliptic curves, we need to use SEA isogeny algorithm, specifically Elkies steps (Algorithm 3 and 4) in [8, p. 12].

Algorithm 4 Algorithm ECSupVer

Input: a prime p, integers N and τ, and a quantum state $|\psi\rangle$ stored in a register $\mathcal{S}$

Output: a bit 0 or 1, then ECSupVer alters $|\psi\rangle$ to a state $|\psi'\rangle$ which it then outputs.

1: Check that $|\psi\rangle$ is properly formatted. If not output 0
2: Use Schoof's algorithm to compute the cardinality of the elliptic curve in a new register.
3: Measure the value in the new register. If it is not N, output 0. Otherwise, compute the list of group actions B_K and discard the new register.
4: Let $r = \#B_K$. Using a new register, create the state $|\varphi\rangle := |\mathbf{1}_{2r}\rangle \otimes |\psi\rangle$.
5: Repeat the following τ times:
   1. Apply the unitary U to $|\varphi\rangle$.
   2. Apply the projection-valued measurement corresponding to $|\mathbf{1}_{2r}\rangle \langle \mathbf{1}_{2r}|$ to the resulting state. If the measurement fails output 0
6: **return** 0

– $|\mathbf{1}_n\rangle := \frac{1}{\sqrt{n}} \sum_{i=0}^{n} |i\rangle$.

- $U := \sum_{i=1}^{r} |i\rangle\langle i| \otimes \sigma_i + \sum_{i=r+1}^{2r} |i\rangle\langle i| \otimes I_k$ where $r = \#B_K$, $k = \#\mathcal{I}_N$, σ_i is a group action in B_K and $\mathcal{I}_N$ is the set of elliptic curves with N points.
- The group action σ_i can be efficiently evaluated by Theorem 4.
- B_K is *Bach Generating Set* generated by [11, Algorithm 4.1], which is the set of ideal classes of unramified primes $\mathfrak{l}$ of a ring of integers $\mathcal{O}_K$ of an imaginary quadratic field K with $N(\mathfrak{l}) < 6(\log \mathrm{Disc}(K))^2$.

By [11, Proposition 8.3], Algorithm 5 runs in

$$\max(O(\log^8 p), O(\tau(\log^5 p)(\log\log^2 p)(\log\log\log^2 p)))$$

where $\tau = 33r^3\lambda$ for $r = \#B_K$. The second term corresponds to the running time of the SEA isogeny algorithm, which dominates or is at least comparable to the complexity of the point-counting algorithm when $\lambda = O(\log p)$ and $\#B_K = O(\log p)$.

Algorithm 5 Algorithm Ver

Input: a quantum state $|\psi\rangle$ and a serial number $\sigma \in \mathbb{F}_p$.
Output: 0, $\perp$ or 1, $|\psi'\rangle$.

1: Run Algorithm 4 and receive an output tuple $(|\psi'\rangle, b)$.
2: If $b = 0$ then return 0 and $\perp$ and discard $|\psi'\rangle$.
3: **return** 1 and $|\psi'\rangle$.

3 Quantum Random Walk-Based Forgery Attack

In this section, we introduce an attack method based on the quantum random walk. We demonstrate the time complexity of this attack and emphasize that this method is much slower than the brute-force attack using the point-counting algorithm.

3.1 Difficulty of Applying Kuperberg's Algorithm

Given a serial number σ, there is no known general polynomial-time algorithm to construct an elliptic curve with cardinality exactly σ. However, suppose that one manages to obtain such a curve E_σ, for example, by collapsing the given quantum money state. From E_σ as the starting curve, one can simulate a quantum random walk over the isogeny graph, resulting in an (almost) uniform superposition of the form $\sum_{\mathfrak{a}} |\mathfrak{a}\rangle |[\mathfrak{a}]E_\sigma\rangle$. To pass the verification process, we must discard the first register $|\mathfrak{a}\rangle$ and retain only the second. This task is known as the Index Erasure Problem.

It is well known that the discrete logarithm problem in group actions can be reduced to the HSP (Hidden Shift Problem) and it can be solved in sub-exponential time by using Kuperberg's algorithm [9]. Here we note that Kuperberg's algorithm is infeasible to erase the index of a superposition state, which underpins the exponential security of the quantum money scheme in [11].

In Kuperberg's algorithm, a weak Fourier measurement is applied to a quantum query of the hidden function f, which yields a qubit (known as a phase vector) whose phases encode information about the hidden shift s:

$$\sum_{0 \le j < \ell} \exp(2\pi i b_j s / 2^n) \, |j\rangle$$

This procedure assumes the ability to measure and isolate a specific instance of the HSP. However, we are dealing with a superposition over multiple HSP instances and Kuperberg's algorithm becomes infeasible to extract a useful set of phase vectors. As a result, the most viable strategy is to apply a quantum search algorithm to recover the index $\mathfrak{a}$, which still requires exponential time.

3.2 Time Complexity of Solving the Index Erasure Problem

By using the quantum search algorithm, we aim to find a group action $\mathfrak{a}$ such that $[\mathfrak{a}]E_\sigma = E$. The size of the search space is determined by the class number $h(d)$ of an imaginary quadratic number field. As we illustrate in Section 5, the lower bound of $h(d)$ is $\Theta(\frac{\sqrt{p}}{\log p})$.

Assuming that the search oracle requires time T_1, the total time complexity of the quantum search algorithm is $T_1 \times \sqrt{h(d)}$ and the lower bound is $\Theta(p^{1/4} \log p^{-1/2} T_1)$. On the other hand, if we use our method to create the desired quantum state, the size of the search space becomes $\frac{2p}{h(d)}$, resulting in a total time complexity of $T_2 \times \sqrt{2p/h(d)}$, with the upper bound of $\Theta(p^{1/4} \log p^{1/2} T_2)$, where T_2 is the time complexity of our custom oracle. While the quantum random walk method offers at most a $\log p$ improvement in the number of search iterations compared to our method (note that the actual factor is smaller on average), the time complexity T_1 is significantly greater than T_2, more than offsetting this advantage.

In [3, p.376], Bach showed that the set B_K of ideal classes of unramified primes $\mathfrak{l}$ of an imaginary quadratic field K, with $N(\mathfrak{l}) < 6 \log^2 d$, generates the class group $\mathrm{Cl}(\mathcal{O}_K)$. Consequently, the size of B_K is approximately $\log(6 \log^2 d)$. Since each ideal class appears as an index in the Index Erasure Problem, we need to compute all ideal classes in $\mathrm{Cl}(\mathcal{O}_K)$ using B_K. The number of isogeny computations using a prime ideal $\mathfrak{l} \in B_K$ is bounded by $h(d)^{1/\#B_K}$, which evaluates to $\Theta(p^{\frac{1}{4 \log(\log p)}})$ when $\log d \approx \log p$. This result makes the overall time complexity of this attack significantly large.

One can address this problem by using a larger factor base B_K to reduce the number of isogeny computations. For example, if we enlarge B_K to have size $\log p$, the maximum norm of prime ideals in B_K would be approximately p and the number of isogeny computations of each prime ideal can be reduced to $O(1)$. However, the cost of each isogeny computation becomes large.

Since we are dealing with a superposition of elliptic curves, we cannot efficiently compute group actions using Vélu's formula. In [11], the authors employ the SEA isogeny algorithm instead. According to [8, p. 12], the bottleneck in the

SEA algorithm is computing the $\mathbb{F}_p$-rational roots of the modular polynomial $\Phi_\ell(x, j_0)$ which is a degree $\ell + 1$ polynomial whose roots are the j-invariants of elliptic curves ℓ-isogenous to the curve with j-invariant j_0. To compute the roots, we compute $\gcd(\Phi_\ell, x^p - x)$ which incurs a time complexity of $O(\ell^2 \log p)$ multiplications in $\mathbb{F}_p$. As an ℓ-isogeny computation requires $O(\ell^2 \log p) = O(p^2 \log p)$, this approach also incurs substantial computational cost.

As the quantum random walk-based attack is significantly inefficient, we focus on optimizing the quantum search algorithm which uses the point-counting algorithm.

4 General Search Algorithm To Forge Quantum Money

In order to find a quantum banknote associated to the given serial number σ, we use Grover search algorithm. Grover search algorithm consists of the oracle $O_{p,\sigma}$, n-bit Walsh-Hadamard gate W_n and the phase rotation gate $O_0 = -2\left|0^{n+3}\right\rangle\left\langle 0^{n+3}\right| + I$. The search algorithm requires $O(1/\sqrt{r})$ queries to $O_{p,\sigma}$ to obtain a uniform superposition of the target set T when the ratio of the target among the given set X is $r = |T|/|X|$. Figure 1 represents the general Grover search algorithm.

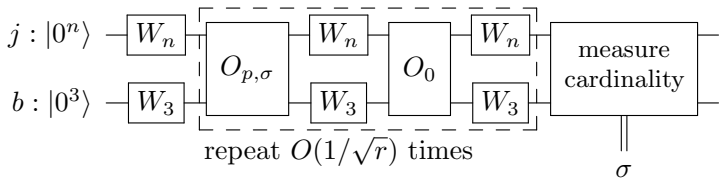

Fig. 1. Grover search algorithm to forge a quantum money

The general oracle $O_{p,\sigma}$ runs as follows:

- The oracle is classically initialized by a positive integer σ satisfying $0 < |\sigma - p - 1| \leq 2\sqrt{p}$ which represents the cardinality of the elliptic curves to be sampled.
- It takes as input a pair of elliptic curve coefficient $(j, b) \in \mathbb{F}_p \times \mathbb{Z}$ and the size of the pair of the quantum registers is $\lceil \log p \rceil + 3$.
- Oracle flips the phase of the quantum state if the Weierstrass curve $E_{j,b}$ has cardinality σ.

The direct approach to get the desired quantum banknote is to search with the minting algorithm illustrated in Algorithm 3. Since the minting algorithm uses Schoof's point-counting algorithm, search oracle $O_{p,\sigma}$ is to count the number of rational points and check if it is the same with the given σ. Because the point-counting algorithm requires to compute arithmetics on a polynomial ring of a

large degree over a finite field, it requires $O(\log^8 p)$ bit operations in total as mentioned in [15]. In Section 6, we show that we can construct a more optimal oracle than using the point-counting algorithm.

To predict the overall time complexity of the search algorithm, we need to compute the ratio of solutions that pass through the oracle $O_{p,\sigma}$. The more accurately we compute the ratio of the solutions, the more efficiently the search algorithm can be executed. In our case, the size of the target set T equals to the class number of a imaginary quadratic field K such that the endomorphism ring of the elliptic curves are isomorphic to the maximal order $\mathcal{O}_K$ of K since the quantum state generated by Algorithm 3 satisfies that $\Delta_{\mathrm{Fr}}(E)$ is square-free and $\mathbb{Z}[\mathrm{Fr}] = \mathrm{End}(E) \cong \mathcal{O}_K$.

Computing the class number of a given number field requires sub-exponential time in a classical setting [6, Section 5.4]. However, a polynomial-time quantum algorithm has been developed to compute the class number [4]. This result is based on a quantum reduction from the class group problem (CGP) to the continuous hidden subgroup problem (CHSP), which can be efficiently solved in a quantum setting. The cost of solving CHSP has been tightly estimated in [5].

Although we do not delve into the quantum algorithm for CHSP in this paper, the class number is crucial to determine the number of search iterations. We provide the lower and upper bound of class numbers, which leads to the bound of the number of search iterations.

5 Lower and Upper Bound of Class Numbers

For the elliptic curves generated by Algorithm 3, the number of $\mathbb{F}_p$-isomorphism classes of elliptic curves of cardinality σ is equal to the class number $h(d)$, where d is the discriminant of an imaginary quadratic order $\mathcal{O}_K$ isomorphic to $\mathrm{End}(E)$. In [11], the lower bound of the class number was provided. The lower bound is calculated from the Dirichlet class number formula and the bound of the Dirichlet series $L(1, \chi)$.

By Dirichlet class number formula, given an integer $d < -4$, the class number $h(d)$ of the imaginary quadratic field of discriminant d satisfies the following equation.

$$h(d) = \frac{\sqrt{|d|}}{\pi} L(1, \chi)$$

where $L(1, \chi)$ is Dirichlet L-function and $\chi(m) = \left(\frac{d}{m}\right)$ is Kronecker symbol.

Theorem 5. *For a negative integer d, let $0 < \epsilon < \frac{1}{2}$, $|d| \geq \max(e^{1/\epsilon}, e^{11.2})$ and $\chi(m) = \left(\frac{d}{m}\right)$. Then*

$$L(1, \chi) > 0.655 \frac{\epsilon}{|d|^{\epsilon}}$$

Proof. We refer to the proof in [17, Theorem 2].

For a large prime p, when $\epsilon = 1/\ln p$ and $|d| > \max(p, e^{11.2}) = p$, this theorem leads directly to the lower bound of the class number.

$$h(d) = \frac{\sqrt{|d|}}{\pi} L(1,\chi) > 0.11 \frac{\sqrt{p}}{\log p}.$$

Using research on number fields, we can also derive the upper bound of class numbers. Next, we show that the upper bound of class numbers can be approximated based on Pólya-Vinogradov inequality.

Theorem 6 (Pólya-Vinogradov Inequality). *Let d be a positive integer and $\chi(k)$ is a Dirichlet character modulus d. Then*

$$\forall m, n \in \mathbb{N}, \sum_{k=n}^{m} \chi(k) = O(\sqrt{d} \log d).$$

Pólya-Vinogradov Inequality is improved in [12]. They suggest a more explicit version of inequality for the sum of values of a Dirichlet character on an interval.

Theorem 7 (Theorem 1, [12]). *Let d be a positive integer and $\chi(k)$ is a primitive Dirichlet character modulus d. Then*

$$\forall m, n \in \mathbb{N}, \sum_{k=n}^{m} \chi(k) \leq \begin{cases} d^{1/2}\left(\frac{2}{\pi^2}\ln d + \frac{4}{\pi^2}\ln\ln d + \frac{3}{2}\right) & \text{if } \chi \text{ is even.} \\ d^{1/2}\left(\frac{1}{2\pi}\ln d + \frac{1}{\pi}\ln\ln d + 1\right) & \text{if } \chi \text{ is odd.} \end{cases}.$$

Proof. The proof is in [12, Theorem 1].

Theorem 8. *For a negative integer d and a Dirichlet character $\chi(m) = \left(\frac{d}{m}\right)$ modulus $|d|$,*

$$L(1,\chi) \leq \left(\frac{1}{2} + \frac{1}{2\pi}\right)\ln|d| + \frac{1}{\pi}\ln\ln|d| + 1$$

Proof. This can be proved as follows.

$$\begin{aligned} L(1,\chi) &= \sum_{n\geq 1} \chi(n) n^{-1} = \sum_{n\geq 1}\left(\chi(n)\int_n^{\infty} x^{-2}dx\right) \\ &= \int_1^{\infty}\left(\sum_{n\leq x}\chi(n)\right)x^{-2}dx \end{aligned}$$

Since d is negative, $\chi(-1) = -1$ which is odd Dirichlet character. By Theorem 7,

$$\begin{aligned} &\leq \int_1^{\sqrt{|d|}} x\cdot x^{-2}dx + \int_{\sqrt{|d|}}^{\infty}\left(\frac{1}{2\pi}|d|^{1/2}\ln|d| + \frac{1}{\pi}|d|^{1/2}\ln\ln|d| + |d|^{1/2}\right)\cdot x^{-2}dx \\ &= \left(\frac{1}{2} + \frac{1}{2\pi}\right)\ln|d| + \frac{1}{\pi}\ln\ln|d| + 1 \end{aligned}$$

□

Theorem 8 leads directly to the upper bound of the class number. As $|d| = |4p - t^2| \leq 4p$,

$$h(d) = \frac{\sqrt{|d|}}{\pi} L(1,\chi) \leq \left(\frac{1+\pi}{\pi^2}\right)\sqrt{p}\ln(4p) + \frac{2}{\pi^2}\sqrt{p}\ln\ln(4p) + \frac{2}{\pi}\sqrt{p}. \quad (1)$$

We can see that the result of the calculating class number $h(d)$ using [4] is bounded by the above inequality (1), which means the number of iterations in Grover search algorithm of Figure 1 is bounded as follows:

$$\frac{\sqrt{2}\pi p^{1/4}}{\sqrt{(\pi+1)\ln(4p) + 2\ln\ln(4p) + 2\pi}} \leq \sqrt{\frac{2p}{h(d)}} \leq 4.251 p^{1/4}\sqrt{\log p}.$$

The lower bound is approximately $2.622\frac{p^{1/4}}{\sqrt{\log p + \Theta(\log\log p)}}$.

6 Implementation of Our Attack

6.1 Division Polynomials and Recurrence Relation

Given a prime p and a positive integer ℓ, one can calculate a division polynomial $\psi_\ell(x,y) \in \mathbb{F}_p[x,y]$ such that $\psi_\ell(x,y) = 0$ if and only if $(x,y) \in E[\ell]$. For a given Weierstrass form $E : y^2 = x^3 + Ax + B$ where $A, B \in \mathbb{F}_p$, we define the *division polynomials* as follows :

$$\begin{aligned}
\psi_{-1} &= -1,\\
\psi_0 &= 0,\\
\psi_1 &= 1,\\
\psi_2 &= 2y,\\
\psi_3 &= 3x^4 + 6Ax^2 + 12Bx - A^2\\
\psi_4 &= 2y(2x^6 + 10Ax^4 + 40Bx^3 - 10A^2x^2 - 8ABx - 2A^3 - 16B^2),\\
\psi_{2n+1} &= \psi_{n+2}\psi_n^3 - \psi_{n-1}\psi_{n+1}^3 = g_1(\psi_{n-1},\psi_n,\psi_{n+1},\psi_{n+2}) && \text{for } n \leq 2\\
\psi_{2n} &= \frac{\psi_{n-1}^2\psi_n\psi_{n+2} - \psi_{n-2}\psi_n\psi_{n+1}^2}{\psi_2} = g_2(\psi_{n-2},\psi_{n-1},\psi_n,\psi_{n+1},\psi_{n+2}) && \text{for } n \leq 3
\end{aligned}$$

To compute the division polynomial, we need the value of y, which involves calculating a square root in $\mathbb{F}_p$. When $p \equiv 1 \mod 4$, this can be done using the Tonelli-Shanks algorithm, which requires $O(\log^2 p)$ multiplications in $\mathbb{F}_p$, making it relatively expensive. However, if we compute the division polynomial in the polynomial ring $\mathbb{F}_p[y]$, using the defining relation $y^2 = x^3 + Ax + B \in \mathbb{F}_p$, we can avoid explicitly computing the square root. This approach remains valid even in cases where such a y does not exist in $\mathbb{F}_p$. Moreover, every division polynomial ψ_ℓ

is either of the form $y \cdot f(x)$ or $f(x)$ for some polynomial $f(x) \in \mathbb{F}_p[x]$, meaning that it only requires $\log p$ qubits to represent $\psi_\ell \in \mathbb{F}_p[y]$. In particular, when $\psi_\ell = 0$, it corresponds uniquely to the zero polynomial in $\mathbb{F}_p[y]$.

Given a tuple $(\psi_k, ..., \psi_{k+9}) \in \mathbb{F}_p[y]^{10}$, we obtain $(\psi_{2k+4}, ..., \psi_{2k+15})$ using the above recurrence relation. Defining $\Psi_k := (\psi_k, ..., \psi_{k+9}) \in \mathbb{F}_p[y]^{10}$, we can determine Ψ_{2k+4}, Ψ_{2k+5} and Ψ_{2k+6} given Ψ_k. For $m > 5$, Ψ_m can be computed as follows:

$$\Psi_m = \begin{cases} f_1(\Psi_{(m-4)/2}) & \text{if } m \text{ is even} \\ f_2(\Psi_{(m-5)/2}) & \text{if } m \text{ is odd} \end{cases}$$

where

$$f_1(\Psi_k)[i] = \begin{cases} g_2(\Psi_k[i/2], \ldots, \Psi_k[i/2+4]) & \text{if } i \text{ is even} \\ g_1(\Psi_k[(i-1)/2], \ldots, \Psi_k[(i-1)/2+4]) & \text{if } i \text{ is odd} \end{cases} \tag{2}$$

$$f_2(\Psi_k)[i] = \begin{cases} g_2(\Psi_k[i/2+1], \ldots, \Psi_k[i/2+5]) & \text{if } i \text{ is even} \\ g_1(\Psi_k[(i-1)/2+1], \ldots, \Psi_k[(i-1)/2+5]) & \text{if } i \text{ is odd} \end{cases}$$

for $0 \leq i < 10$ and $\Psi_k[i]$ denotes the i-th element ψ_{k+i} in the tuple Ψ_k. Algorithm 6 is the quantum algorithm that computes division polynomials.

Algorithm 6 Algorithm $\mathsf{DivisionPolynomial}_{\mathsf{p},\sigma}$

Input: A prime p, a cardinality $\sigma \in \mathbb{N}$ and a quantum state $|A\rangle |B\rangle |x\rangle$ where $A, B, x \in \mathbb{F}_p$

Output: A quantum state $|\psi_\sigma(A, B, x)\rangle$.

1: Compute $|\Psi_{\sigma_r}\rangle$ from $|A\rangle |B\rangle |x\rangle$.
2: **for** $0 \leq i < r$ **do**
3: Compute $|\Psi_{\sigma_{r-i-1}}\rangle \leftarrow |f_{b_i}(\Psi_{\sigma_{r-i}})\rangle$.
4: **end for**
5: **for** $r - 1 > i \geq 0$ **do**
6: Uncompute $|\Psi_{\sigma_{r-i-1}}\rangle \leftarrow |f_{b_i}^{-1}(\Psi_{\sigma_{r-i}})\rangle$ and discard $|\Psi_{\sigma_{r-i-1}}\rangle$.
7: **end for**
8: Uncompute $|\Psi_{\sigma_r}\rangle$ from $|A\rangle |B\rangle |x\rangle$.
9: **return** $|\psi_\sigma(A, B, x)\rangle$.

In Algorithm 6, f_1 and f_2 are the functions in Equation (2) and σ_i's and b_i's are precomputed as follows:

$$\sigma_0 = \sigma \text{ and } \sigma_r \leq 5$$

$$\sigma_{i+1} = \begin{cases} (\sigma_i - 4)/2 & \text{if } \sigma_i \text{ is even and } \sigma_i > 5 \\ (\sigma_i - 5)/2 & \text{if } \sigma_i \text{ is odd and } \sigma_i > 5 \\ \sigma_i & \text{if } \sigma_i \leq 5 \end{cases}$$

$$b_{r-i-1} = \begin{cases} 1 & \text{if } \sigma_i \text{ is even} \\ 2 & \text{if } \sigma_i \text{ is odd} \end{cases}$$

where r is the smallest positive integer such that $\sigma_r = \sigma_{r+1}$ and $0 \leq i < r$.

Through this approach, we can deduce that computing ψ_ℓ requires $O(\log \ell)$ multiplication in $\mathbb{F}_p$ which translates to $O(\log^2 p \log \ell)$ bit operations. This process could be further optimized in future research.

6.2 Search Oracle $O_{p,\sigma}$ With Rational Points

In this section, we describe our quantum search oracle used to forge a quantum banknote. The oracle $O_{p,\sigma}$ consists of the evaluation of division polynomials and the phase rotation via the n-controlled-Z gate Z_n. In order to filter out all but the target curves, we verify that all rational points are annihilated by σ, while all rational points on its quadratic twist are annihilated by $2p + 2 - \sigma$. Given $x \in \mathbb{F}_p$, we determine the appropriate case by checking whether $x^3 + Ax + B$ is a quadratic residue. If it is, we verify annihilation by σ; otherwise, we verify annihilation by $2p + 2 - \sigma$. We then define $G_{p,\sigma}$ as a function that calculates the division polynomial with respect to the original curve and its twist. Instead of applying scalar multiplications, we compute division polynomials to verify annihilation, as this approach is computationally more efficient.

Remark 1. Using the Montgomery ladder algorithm, scalar multiplication may be faster than with division polynomials. Since the algorithm operates solely on x-coordinates, it also avoids square-root computations. However, in order to use the Montgomery ladder, Montgomery coefficient must be derived from a given j-invariant. The relation between them is given by $j = \frac{256(A^2-3)^3}{A^2-4}$. This implies that obtaining the corresponding Montgomery coefficient requires solving a cubic equation and computing a square root.

For elliptic curves $E_{A,B}$ such that $\#E_{A,B} \neq \sigma$, some rational points may still be annihilated by σ. The set of such points forms a subgroup $S \subset E_{A,B}(\mathbb{F}_p)$. If S is a proper subgroup, its complement $E_{A,B} \setminus S$ consists of the union of all other cosets of S, ensuring that $|E_{A,B} \setminus S| \geq |S|$. Consequently, if at least one rational point is not annihilated by σ, then at least half of the rational points remain unannihilated. We exploit this fact to distinguish the target curve from the others by computing $F_{p,\sigma,\tau}$.

For a positive integer τ, given a rational point $x \in \mathbb{F}_p$, $F_{p,\sigma,\tau}(A,B,x)$ is defined as follows:

$$G_{p,\sigma}(A,B,x) = \begin{cases} \psi_\sigma(A,B,x) & \text{if } (x^3+Ax+B)^{\frac{p-1}{2}} \equiv 1 \mod p \\ \psi_{2p+2-\sigma}(A,B,x) & \text{otherwise} \end{cases}$$

$$F_{p,\sigma,\tau}(A,B,x) = \sum_{i=0}^{\tau-1} G_{p,\sigma}(A,B,x+i)$$

The function $F_{p,\sigma,\tau}$ corresponds to Algorithm 7. $\mathsf{DivisionPolynomial}_{\mathsf{p},\sigma}$ refers to Algorithm 6.

Algorithm 7 Algorithm $F_{p,\sigma,\tau}$

Input: A prime p, a cardinality $\sigma \in \mathbb{N}$ and a quantum state $|A\rangle|B\rangle|x\rangle$ where $A,B,x \in \mathbb{F}_p$ and $0 < |\sigma - p - 1| \leq 2\sqrt{p}$.

Output: A quantum state $|F_{p,\sigma,\tau}(A,B,x)\rangle$.

1: **for** $0 \leq i < \tau$ **do**
2: Compute Euler's criteria $|t\rangle \leftarrow |(x^3+Ax+B)^{\frac{p-1}{2}} \mod p\rangle$.
3: Compute the division polynomials $|r_{i,1}\rangle \leftarrow \mathsf{DivisionPolynomial}_{p,\sigma}(|A\rangle|B\rangle|x\rangle)$ and $|r_{i,2}\rangle \leftarrow \mathsf{DivisionPolynomial}_{p,2p+2-\sigma}(|A\rangle|B\rangle|x\rangle)$.
4: Compute $|r\rangle \leftarrow |r + (t \equiv 1) \times r_{i,1} + (t \not\equiv 1) \times r_{i,2}\rangle$.
5: Uncompute $|t\rangle$ and discard it.
6: $|x\rangle \leftarrow |x+1\rangle$.
7: **end for**
8: **for** $\tau > i \geq 0$ **do**
9: $|x\rangle \leftarrow |x-1\rangle$.
10: Compute Euler's criteria $|t\rangle \leftarrow |(x^3+Ax+B)^{\frac{p-1}{2}} \mod p\rangle$.
11: Uncompute the division polynomials $|r_{i,1}\rangle \leftarrow \mathsf{DivisionPolynomial}^{-1}_{p,\sigma}(|A\rangle|B\rangle|x\rangle)$ and $|r_{i,2}\rangle \leftarrow \mathsf{DivisionPolynomial}^{-1}_{p,2p+2-\sigma}(|A\rangle|B\rangle|x\rangle)$ and discard $|r_{i,1}\rangle$ and $|r_{i,2}\rangle$.
12: Uncompute $|t\rangle$ and discard it.
13: **end for**
14: **return** $|r\rangle$

Here we show that Algorithm 7 runs correctly. The proof is based on Theorem 9 which is also known as Mestre's theorem.

Theorem 9. *Given a large prime $p > 2^{20}$ and an positive integer σ such that $0 < |\sigma - p - 1| \leq 2\sqrt{p}$, there is no elliptic curve $E_{A,B}$ defined over $\mathbb{F}_p$ such that $[\sigma]P = 0_E$ for all rational points $P \in E_{A,B}(\mathbb{F}_p)$, $[2p+2-\sigma]Q = 0_{E^t}$ for all rational points $Q \in E^t_{A,B}(\mathbb{F}_p)$ and the cardinality of $E_{A,B}$ differs from σ, where E^t is a quadratic twist of E.*

Proof. Suppose that a curve $E_{A,B}$ satisfies the condition. According to Theorem 1, let's say that m_1, k_1, m_2, k_2 are positive integers such that

$$E_{A,B} \cong \mathbb{Z}/m_1\mathbb{Z} \times \mathbb{Z}/m_1k_1\mathbb{Z}$$
$$E^t_{A,B} \cong \mathbb{Z}/m_2\mathbb{Z} \times \mathbb{Z}/m_2k_2\mathbb{Z}$$

where $E^t_{A,B}$ is a quadratic twist of $E_{A,B}$ and $m_1, m_2|(p-1)$. Let's denote the cardinality $\#E_{A,B}$ as σ'. Since $m_1^2 k_1 = \sigma'$ and $m_2^2 k_2 = 2p + 2 - \sigma'$ by Theorem 3, we get

$$\begin{aligned} \gcd(m_1^2 k_1, m_2^2 k_2) &= \gcd(\sigma', 2p + 2 - \sigma') \\ &= \gcd(\sigma', 2p + 2) \end{aligned} \tag{3}$$

According to the condition, all elements in $\mathbb{Z}/m_1 k_1 \mathbb{Z}$ are annihilated by both σ and σ'. Thus $m_1 k_1 | (\sigma - \sigma')$ and likewise, we deduce that $m_2 k_2 | (2p + 2 - \sigma - 2p - 2 + \sigma') = (\sigma' - \sigma)$. By Hasse's theorem, we obtain

$$\frac{m_1 k_1 \times m_2 k_2}{\gcd(m_1 k_1, m_2 k_2)} \Big| |\sigma - \sigma'| \leq 4\sqrt{p} \tag{4}$$

Since $m_1^2 k_1^2 \geq m_1^2 k_1 = \#E_{A,B}(\mathbb{F}_p) \geq p + 1 - 2\sqrt{p}$, we get $m_1 k_1 \geq \sqrt{p} - 1$. Likewise, $m_2 k_2 \geq \sqrt{p} - 1$ on the twist. As $\sigma - \sigma' \neq 0$,

$$\gcd(m_1 k_1, m_2 k_2) \geq \frac{m_1 k_1 \times m_2 k_2}{4\sqrt{p}} \geq \frac{(\sqrt{p} - 1)^2}{4\sqrt{p}} > \frac{\sqrt{p}}{4} - 1$$

As $m_1, m_2 | (p-1)$, we get $\gcd(p + 1, m_i) \leq 2$ and by (3),

$$\begin{aligned} &\gcd(m_1 k_1, m_2 k_2) = 2 \gcd(k_1, k_2) \text{ or } \gcd(k_1, k_2) \\ \Rightarrow &k_1, k_2 > \frac{\sqrt{p}}{8} - \frac{1}{2} \end{aligned}$$

Again by Hasse's theorem, $m_1^2 k_1 = \#E_{A,B}(\mathbb{F}_p) \leq p + 1 + 2\sqrt{p}$ and

$$\begin{aligned} m_1 &\leq \frac{\sqrt{p} + 1}{\sqrt{k_1}} < 2\sqrt{2} \frac{\sqrt{p} + 1}{\sqrt{\sqrt{p} - 4}} < 3p^{1/4} \\ \Rightarrow m_1 k_1 &= \frac{\#E_{A,B}(\mathbb{F}_p)}{m_1} \geq \frac{p + 1 - 2\sqrt{p}}{m_1} > \frac{p^{3/4}}{\sqrt{10}} \end{aligned}$$

Likewise, we can deduce that $m_2 k_2 > \frac{p^{3/4}}{\sqrt{10}}$. This again leads to (4) and we get

$$\begin{aligned} \gcd(m_1 k_1, m_2 k_2) &> \frac{m_1 k_1 \times m_2 k_2}{4\sqrt{p}} > \frac{p}{40} \\ \Rightarrow \frac{p}{40} &< \gcd(m_1^2 k_1, m_2^2 k_2) \leq 2 \gcd(p + 1, \sigma') \end{aligned}$$

If $\sigma' \neq p + 1$, then $\gcd(p + 1, \sigma') \leq |\sigma' - p - 1| \leq 2\sqrt{p}$ which leads to a contradiction. Otherwise, we have $\sigma' = p + 1$. It follows that $m_1 = 2$ or 1, implying that $\frac{p+1}{2} | \sigma$. This is impossible since $\sigma \neq p + 1$. □

Based on Theorem 9, $F_{p,\sigma,\tau}$ can be used to filter out all but the target curve of cardinality σ. We conclude by Corollary 1.

Corollary 1. *If* $\#E_{A,B}(\mathbb{F}_p) = \sigma$*, then* $F_{p,\sigma,\tau}(A, B, x) = 0$ *for all* $x \in \mathbb{F}_p$*, otherwise, the probability that* $F_{p,\sigma,\tau}(A, B, x) = 0$ *for* $x \in \mathbb{F}_p$ *is at most* $\left(\frac{3}{4} + \Theta\left(\frac{1}{\sqrt{p}}\right)\right)^{\tau}$.

Proof. It is trivial when the target curve is given. Let $E_{A,B}$ has different cardinality from σ. By Theorem 9, there is at least one rational point which is not annihilated among all rational points of $E_{A,B}$ and its twist $E^t_{A,B}$. Without loss of generality, let's assume that we can find one in $E_{A,B}$. The set of all rational points in $E_{A,B}$ annihilated by σ is a proper subgroup S of $E_{A,B}(\mathbb{F}_p)$. Then $|E_{A,B}(\mathbb{F}_p) \setminus S| > |S|$, since $E_{A,B}(\mathbb{F}_p) \setminus S$ contains at least one coset of S. When the all rational points on the quadratic twist satisfy the condition, the ratio of S among $E_{A,B}(\mathbb{F}_p) \cup E^t_{A,B}(\mathbb{F}_p)$ is at most $\frac{p+1+2\sqrt{p}}{4p+4} = \frac{1}{4} + \Theta\left(\frac{1}{\sqrt{p}}\right)$.

Since there is no known algebraic relation between x and $x + i$ from the perspective of the elliptic curve group, we assume that adding the integer i acts like a random sampling of rational points on the elliptic curve. From the set of $\{x, x + 1, ..., x + \tau - 1\}$, the probability that $G_{p,\sigma}(A, B, x + i) = 0$ for all $i \in [0, \tau)$ is less than $\left(1 - \frac{1}{4} + \Theta\left(\frac{1}{\sqrt{p}}\right)\right)^{\tau}$, which leads to that the probability $F_{p,\sigma,\tau}(A, B, x) = 0$ is less than $\left(\frac{3}{4} + \Theta\left(\frac{1}{\sqrt{p}}\right)\right)^{\tau}$. □

Using Algorithm 7 as a building block, we replace the search oracle $O_{p,\sigma}$ in Figure 1 by Algorithm 8. Z_n is the n-controlled Pauli-Z gate combined with n-bit NOT gates, which flips the phase only if it takes as input $|0^n\rangle$.

Algorithm 8 Algorithm OurOracle $O_{p,\sigma}$

Input: A prime p, a cardinality $\sigma \in \mathbb{N}$ and a quantum state $|j\rangle |b\rangle$ where $(j, b) \in \mathbb{F}_p \times \mathbb{Z}$ and $0 \leq b \leq 5$.

Output: A quantum state $-|j\rangle |b\rangle$ if the cardinality of $E_{j,b}$ equals to σ, outputs $|j\rangle |b\rangle$ otherwise.

1: Set $n \leftarrow \lceil \log p \rceil$ and $\tau \leftarrow 3\lceil \log p \rceil$.
2: Compute $|A\rangle |B\rangle \leftarrow \mathsf{GetWeierstrassPair}_p(|j\rangle |b\rangle)$
3: Set $|x\rangle \leftarrow |0^n\rangle$.
4: Compute $|r\rangle \leftarrow F_{p,\sigma,\tau}(|A\rangle |B\rangle |x\rangle)$.
5: Compute $|r\rangle \leftarrow Z_n(|r\rangle)$. ▷ Flip the phase only if $r = 0^n$
6: Uncompute $|r\rangle \leftarrow F^{-1}_{p,\sigma,\tau}(|A\rangle |B\rangle |x\rangle)$ and discard $|x\rangle$ and $|r\rangle$.
7: Uncompute $|A\rangle |B\rangle \leftarrow \mathsf{GetWeierstrassPair}_p^{-1}(|j\rangle |b\rangle)$ and discard $|A\rangle |B\rangle$.
8: **return** $|j\rangle |b\rangle$.

Theorem 10. *Algorithm 8 runs correctly for a large prime* $p > 2^{20}$ *and a positive integer* σ *such that* $0 < |\sigma - p - 1| \leq 2\sqrt{p}$*, i.e. it flips only the phase of target elliptic curves of cardinality* σ*.*

Proof. The phase is flipped if and only if $F_{p,\sigma,\tau}(A, B, 0^n) = 0$. Given a curve $E_{A,B}$ of cardinality different from σ, by Corollary 1, the probability that $x \in \mathbb{F}_p$

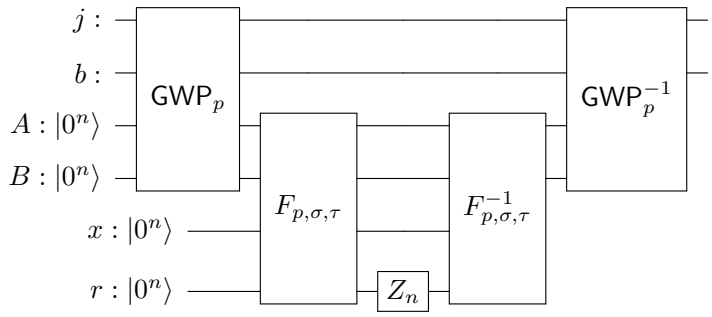

Fig. 2. The quantum circuit of $\mathsf{OurOracle}$ $O_{p,\sigma}$. GWP_p gate represents the $\mathsf{GetWeierstrassPair}_p$ algorithm and Z_n flips the phase only if $r = |0^n\rangle$.

satisfies $F_{p,\sigma,\tau}(A, B, x) = 0$ is at most

$$\left(\frac{3}{4} + \Theta\left(\frac{1}{\sqrt{p}}\right)\right)^{3\lceil \log p \rceil} < \frac{1}{p^2}$$

Since there are less than $3p$ pairs of $(j, b) \in \mathbb{F}_p \times \mathbb{Z}$, the expected number of curves $E_{A,B}$ such that $F_{p,\sigma,\tau}(A, B, 0^n) = 0$ is significantly less than 1. □

6.3 Quantum Resource Estimation

Given a serial number σ such that $0 < |\sigma - p - 1| \leq 2\sqrt{p}$, Algorithm 6 requires fewer than $80 \log p$ field multiplications in $\mathbb{F}_p$, assuming that g_1 and g_2 are computed naively with at most 4 multiplications in $\mathbb{F}_p$. Algorithm 7 invokes running Algorithm 6 4τ times and calculates Euler's criteria 4τ times. When $\tau = 3 \log p$, this results in fewer than $972 \log^2 p$ multiplications in $\mathbb{F}_p$. Consequently, Algorithm 8, which executes Algorithm 7 twice, requires fewer than $1944 \log^2 p$ multiplications in $\mathbb{F}_p$.

attack method	oracle $O_{p,\sigma}$	num. of iterations	num. of qubits
brute-force	$O(\log^6 p)$ $\mathbb{F}_p$-Mul	$\sqrt{2p/h(d)}$	$O(\log^3 p)$
our method	$< 1944 \log^2 p$ $\mathbb{F}_p$-Mul	$\sqrt{2p/h(d)}$	$< 12\lceil \log p \rceil^2$

Table 1. The comparison between the brute-force attack and our method. $h(d)$ is the class number corresponding to the given serial number σ. The boundary of the number of iterations $\sqrt{2p/h(d)}$ is calculated in Section 5.

From the perspective of the number of qubits, our algorithm also consumes less number of qubits. Each Ψ_{σ_i} accounts for $10\lceil \log p \rceil$ qubits. Thus, $10\lceil \log p \rceil^2$

qubits are needed in total. Algorithm 7 stores $r_{i,1}$ and $r_{i,2}$ every τ iterations, which requires $2\lceil \log p \rceil^2$ additional qubits. On the other hand, in Schoof's point counting algorithm [15], the ring element in $\mathbb{F}_p[X,Y]/(\psi_\ell(X), Y^2 - X^3 - AX - B)$ has size $\lceil \log p \rceil^3$.

In total, by combining the upper and lower bound on the class numbers from Section 5, the process of forging a quantum banknote using our method requires at least $5097 \frac{p^{1/4} \log^4 p}{\sqrt{\log p + \Theta(\log \log p)}} = O(p^{1/4} \log^{7/2} p)$ and at most $8264 p^{1/4} \log^{9/2} p = O(p^{1/4} \log^{9/2} p)$ bit operations. The exact time complexity varies depending on the specific class number.

7 Faster Verification of Quantum Money

In the quantum money scheme [11], the verification process is inherently related to the forgery attack. The verification consists of two processes : 1. checking the serial number and 2. checking the uniformity of the quantum state. Each of the two phases are related to two different types of forgery attacks : 1. searching by the serial number and 2. making a superposition by a isogeny walk. Since our attack is the prior case, our method can be applied to checking the validity of the serial number. We can perform a faster verification using rational points.

Algorithm 9 Algorithm CheckSerialNumber

Input: A prime p, a serial number $\sigma \in \mathbb{N}$ and a quantum state $|j\rangle |b\rangle$ whre $(j, b) \in \mathbb{F}_p \times \mathbb{Z}$

Output: Outputs 1 if the serial number is valid, outputs 0 otherwise.

1: Set $n \leftarrow \lceil \log p \rceil$ and $\tau \leftarrow 3\lceil \log p \rceil$.
2: Compute $|A\rangle |B\rangle \leftarrow \mathsf{GetWeierstrassPair}_p(|j\rangle |b\rangle)$.
3: Set $|x\rangle \leftarrow |0^n\rangle$.
4: Compute $|r\rangle \leftarrow F_{p,\sigma,\tau}(|A\rangle |B\rangle |x\rangle)$.
5: Uncompute $|A\rangle |B\rangle \leftarrow \mathsf{GetWeierstrassPair}_p^{-1}(|j\rangle |b\rangle)$
6: Measure $|r\rangle$. If it is 0^n, output 1. Otherwise, output 0.

Our method only applied to checking the validity of the serial number σ and verifying the uniformity of the given quantum money is same as the previous method in [11]. Our verification algorithm is directly derived from Algorithm 7. The algorithm $\mathsf{GetWeierstrassPair}_p$ converts the j-invariant form (j, b) into the Weierstrass pair (A, B) using the method illustrated in Algorithm 1. Our verification algorithm is $O(\log^4 p)$ times faster than using the point-counting algorithm. According to Theorem 10, the probability that Algorithm 9 outputs false positive is negligible when p is a large prime.

8 Conclusion

In this work, we propose an attack method to forge quantum money in the isogeny-based quantum money scheme presented in [11] and introduce a more efficient verification algorithm. We employ the fact that checking the exponent of an elliptic curve group with rational points is more efficient than directly computing its cardinality. Compared to the brute-force attack using the point-counting algorithm, our method achieves a speedup of $O(\log^4 p)$. More concretely, we estimate that forging quantum money using our approach requires fewer than $5097 \log^2 p$ multiplications in $\mathbb{F}_p$ for each search iteration and requires approximately $12\lceil \log p \rceil^2$ qubits.

Our key insight is to utilize rational points on elliptic curves $E_{A,B}$ for the efficient computation of division polynomials. Specifically, our method exploits the property of the group structure of quadratic twists of elliptic curves. As our approach leverages the properties of quadratic twists to utilize rational points, we expect it to contribute to future research on quantum algorithms for elliptic-curve-based quantum cryptography.

References

1. Aaronson, S.: Quantum copy-protection and quantum money. In: 2009 24th Annual IEEE Conference on Computational Complexity. pp. 229–242. IEEE (2009)
2. Aaronson, S., Christiano, P.: Quantum money from hidden subspaces. In: Proceedings of the forty-fourth annual ACM symposium on Theory of computing. pp. 41–60 (2012)
3. Bach, E.: Explicit bounds for primality testing and related problems. Mathematics of Computation **55**(191), 355–380 (1990)
4. Biasse, J.F., Song, F.: Efficient quantum algorithms for computing class groups and solving the principal ideal problem in arbitrary degree number fields. In: Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms. pp. 893–902. SIAM (2016)
5. de Boer, K., Ducas, L., Fehr, S.: On the quantum complexity of the continuous hidden subgroup problem. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 341–370. Springer (2020)
6. Cohen, H.: A course in computational algebraic number theory, vol. 138. Springer Science & Business Media (2013)
7. Conde Pena, M., Durán Díaz, R., Faugère, J.C., Hernández Encinas, L., Perret, L.: Non-quantum cryptanalysis of the noisy version of aaronson–christiano's quantum money scheme. IET Information Security **13**(4), 362–366 (2019)
8. De Feo, L., Kieffer, J., Smith, B.: Towards practical key exchange from ordinary isogeny graphs. In: Advances in Cryptology–ASIACRYPT 2018: 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2–6, 2018, Proceedings, Part III 24. pp. 365–394. Springer (2018)
9. Kuperberg, G.: Another subexponential-time quantum algorithm for the dihedral hidden subgroup problem. arXiv preprint arXiv:1112.3333 (2011)

10. Liu, J., Montgomery, H., Zhandry, M.: Another round of breaking and making quantum money: How to not build it from lattices, and more. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 611–638. Springer (2023)
11. Montgomery, H., Sharif, S.: Quantum money from class group actions on elliptic curves. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 33–64. Springer (2025)
12. Pomerance, C.: Remarks on the pólya–vinogradov inequality (2011)
13. Roberts, B.: Security analysis of quantum lightning. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 562–567. Springer (2021)
14. Rück, H.G.: A note on elliptic curves over finite fields. Mathematics of Computation **49**(179), 301–304 (1987)
15. Schoof, R.: Counting points on elliptic curves over finite fields. Journal de théorie des nombres de Bordeaux **7**(1), 219–254 (1995)
16. Silverman, J.H.: The Arithmetic of Elliptic Curves. Graduate texts in mathematics, Springer, Dordrecht (2009). https://doi.org/10.1007/978-0-387-09494-6, https://cds.cern.ch/record/1338326
17. Tatuzawa, T.: On a theorem of siegel. In: Japanese journal of mathematics: transactions and abstracts. vol. 21, pp. 163–178. The Mathematical Society of Japan (1952)
18. Wiesner, S.: Conjugate coding. ACM Sigact News **15**(1), 78–88 (1983)
19. Zhandry, M.: Quantum lightning never strikes the same state twice. or: quantum money from cryptographic assumptions. Journal of Cryptology **34**, 1–56 (2021)