Adaptive attack for a possible PKE scheme using FESTA trapdoor functions in the wrong way

Tomoki Moriya

Computer Science, University of Birmingham, UK
t.moriya@bham.ac.uk

Abstract. Isogeny-based cryptography is one of the candidates for post-quantum cryptography. In 2023, Kani's theorem breaks some isogeny-based schemes including SIDH, which was considered as a promising post-quantum scheme. Though Kani's theorem damaged isogeny-based cryptography, some researchers try to dig into the applications of Kani's theorem. A FESTA trapdoor function is an isogeny-based trapdoor function that is one trial to apply Kani's theorem to cryptography.
The claim of this paper is that there is an adaptive attack if the FESTA trapdoor function is used in the wrong way. In this paper, we provide an adaptive attack for a possible PKE scheme based on FESTA trapdoor functions. Our attack reveals the secret key of the function. Our attack may be used if the recipient of the PKE scheme does not check whether the hidden matrix corresponding to the ciphertext is correct. In other words, the recipient can prevent our attack by checking the correctness of the matrix. Our attack cannot be adapted to IND-CCA PKE schemes named FESTA proposed in the FESTA original paper.

1 Introduction

Public key cryptography is an important technology for today's information society. In particular, we use RSA [18] and Elliptic Curve Cryptography [14,12] to keep our information secure. However, Shor showed that quantum computers with sufficient ability can break these cryptosystems [20]. Therefore, we need to construct new cryptosystems that resist attacks using quantum computers. We call such cryptography post-quantum cryptography (PQC).

Isogeny-based cryptography is one of the candidates for post-quantum cryptography. Isogeny-based cryptography attracts interest from some cryptographers due to its compactness and mathematical structures. Indeed, SIKE [1], which is an isogeny-based key encapsulation scheme based on SIDH [10], remained as an alternative candidate in the 4th round of the NIST PQC standardization process [16].

In 2023, some studies break SIDH and cryptosystems related to SIDH [3,13,19]. These studies use Kani's theorem [11] that describes the relationship between an isogeny diagram of elliptic curves and an isogeny of abelian varieties of dimension 2. Although CSIDH (an isogeny-based key exchange scheme) [4] and SQISign

(an isogeny-based digital signature scheme) [6] and some other schemes do not be broken by these attacks, it was a sore point for isogeny-based cryptography.

On the other hand, Kani's theorem leads to some novel isogeny-based schemes. In 2023, Dartois, Leroux, Robert, and Wesolowski proposed a novel isogeny-based digital signature SQISignHD [5]. This new signature is based on Kani's theorem and is more compact than SQISign. Moreover, Basso, Maino, and Pope proposed a novel isogeny-based trapdoor function and a public key encryption (PKE) scheme based on this trapdoor function FESTA (Fast Encryption from Supersingular Torsion Attacks) [2]. FESTA is also based on Kani's theorem and is expected to lead to a next-generation isogeny-based PKE scheme instead of SIDH. To dig the applications of these new schemes and to analyze their security are important tasks for isogeny-based cryptography.

1.1 Contribution

In this paper, we show that there is an adaptive attack if the FESTA trapdoor function is used in the wrong way. For SIDH, there are several studies of adaptive attacks (*e.g.,* [8,7]). We construct a similar attack to them for FESTA trapdoor functions.

Our attack is an attack for a possible PKE scheme based on a FESTA trapdoor function and reveals the secret key of the function. As a part of the input of a FESTA trapdoor function, there is a 2×2-regular matrix in a fixed set $\mathcal{M}_b$. Usually, we check whether this 2×2-matrix belongs to $\mathcal{M}_b$ in computing the inverse map of the trapdoor function. We show in this paper that our attack works if the recipient of the PKE scheme does not check the inclusivity of this matrix. In other words, one can prevent our attack by checking the inclusivity of the matrix. Because we need to check the ciphertexts in the IND-CCA secure PKE schemes named FESTA proposed in [2], these IND-CCA secure schemes are not threatened by our attack.

2 Preliminaries

In this section, we introduce some mathematical concepts and facts.

2.1 Abelian varieties and isogenies

This subsection provides some knowledge about abelian varieties and isogenies. Refer to [15] and [21] for more detail.

Let k be a field. We denote the characteristic of k by $\mathrm{ch}(k)$. Let A be an algebraic variety over k. If A has a group structure compatible with its structure as an algebraic variety, we call A an abelian variety. If the dimension of A is 1, we call A an elliptic curve. Let d be an integer. The d-torsion subgroup of A is a subgroup of A defined as $\{P \in A \mid dP = 0\}$. We denote this group by $A[d]$. If d is coprime to $\mathrm{ch}(k)$, then it holds that

$$A[d] \cong (\mathbb{Z}/d\mathbb{Z})^{2 \dim A}.$$

Suppose that $\mathrm{ch}(k) = p$ for a prime number p. Let E be an elliptic curve. If it holds that $E[p] = \{0\}$, we call E a supersingular elliptic curve. If an abelian variety A satisfies $A \cong \prod_{i=1}^{\dim A} E_i$ for supersingular elliptic curves $E_1, \ldots, E_{\dim A}$, we call A a superspecial abelian variety.

Let A and B be abelian varieties. An isogeny $\phi\colon A \to B$ is a morphism between A and B such that ϕ is surjective, ϕ is a group morphism, and the kernel of ϕ is a finite subgroup of A. Let G be a finite subgroup of A. There is a separable isogeny $\phi\colon A \to B$ with $\ker\phi = G$. Moreover, an image variety B is unique up to isomorphism. We denote by A/G a representative of an isomorphism class of B. If A is of dimension 1 or 2, there are well-known algorithms to compute an isogeny $A \to A/G$ from given A and G (*e.g.,* [23] and [22]). For an isogeny $\phi_1\colon A \to B$, there is an isogeny $\hat{\phi}_1$ satisfying $\hat{\phi}_1 \circ \phi_1 = \deg\phi_1$ and $\phi_1 \circ \hat{\phi}_1 = \deg\phi_1$. We call $\hat{\phi}_1$ the dual isogeny of ϕ_1.

2.2 Kani's theorem

In this subsection, we introduce Kani's theorem provided in [11]. Kani's theorem describes the relationship between an isogeny of products of two elliptic curves and an isogeny diamond of elliptic curves.

Definition 1 (Isogeny diamond). *Let E_0 be an elliptic curve. Let G_1 and G_2 be finite subgroups of E_0 such that $\gcd(\#G_1, \#G_2) = 1$. Then, there is the following diagram:*

$$\begin{array}{ccc} E_0 & \xrightarrow{\phi_1} & E_0/G_1 \\ {\scriptstyle\phi_2}\downarrow & & \downarrow{\scriptstyle[\phi_1]_*\phi_2} \\ E_0/G_2 & \xrightarrow{[\phi_2]_*\phi_1} & E_0/\langle G_1, G_2\rangle \end{array}$$

Here, an isogeny ϕ_1 (resp. an isogeny ϕ_2) is a separable isogeny with $\ker\phi_1 = G_1$ (resp. $\ker\phi_2 = G_2$), and an isogeny $[\phi_2]_*\phi_1$ (resp. an isogeny $[\phi_1]_*\phi_2$) is a separable isogeny with $\ker[\phi_2]_*\phi_1 = \phi_2(G_1)$ (resp. $\ker[\phi_1]_*\phi_2 = \phi_1(G_2)$). We call this diagram an isogeny diamond.

Theorem 1 (Kani's theorem [11]). *Suppose that there is an isogeny diamond:*

$$\begin{array}{ccc} E_0 & \xrightarrow{\phi_1} & E_1 \\ {\scriptstyle\phi_2}\downarrow & & \downarrow{\scriptstyle[\phi_1]_*\phi_2} \\ E_2 & \xrightarrow{[\phi_2]_*\phi_1} & E_3 \end{array}$$

Then, there is an isogeny $\Phi\colon E_2 \times E_1 \to E_0 \times E_3$ defined as

$$\Phi = \begin{pmatrix} \hat{\phi}_2 & -\hat{\phi}_1 \\ [\phi_2]_*\phi_1 & [\phi_1]_*\phi_2 \end{pmatrix}$$

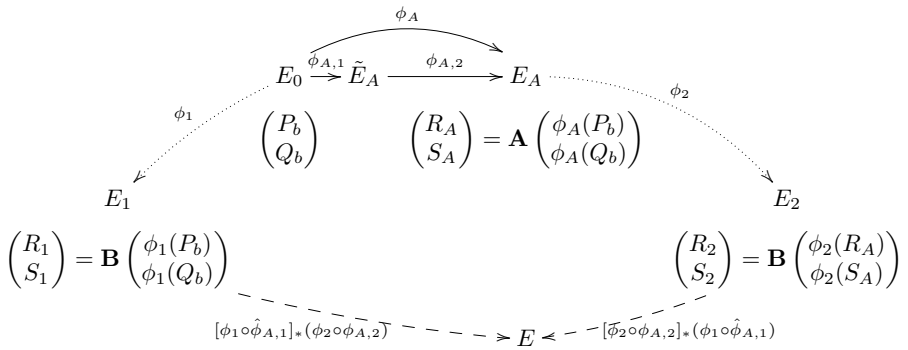
with $\ker\Phi = \langle(\deg\phi_1 P, \phi_1 \circ \hat{\phi}_2(P)) \mid P \in E_2[\deg\phi_1 + \deg\phi_2]\rangle$.

3 FESTA trapdoor function

This section introduces an overview of a FESTA trapdoor function [2].

3.1 Construction

The following diagram shows the overall picture of FESTA trapdoor functions.



We first provide a brief explanation. Notations are the same as in the above diagram. To set up the trapdoor function, we first compute $\tilde{E}_A$, E_A, and (R_A, S_A). Here, a matrix $\mathbf{A}$ belongs to a set $\mathcal{M}_b$ that is defined as a commutative subgroup of a 2×2-linear group over $\mathbb{Z}/2^b\mathbb{Z}$ (*e.g.,* the set of regular circulant matrices). Let E_A, R_A, S_A be published. We define a function f_{E_A,R_A,S_A} as

$$f_{E_A,R_A,S_A}(\mathbf{B}, \phi_1, \phi_2) = (E_1, (R_1, S_1), E_2, (R_2, S_2)).$$

Let $(\mathbf{A}, \phi_{A,1}, \phi_{A,2})$ be a secret key. We call f_{E_A,R_A,S_A} a FESTA trapdoor function. One who knows the secret key can compute the inverse map of the function as follows. Note that $\mathbf{AB} = \mathbf{BA}$. By using Kani's theorem and the matrix $\mathbf{A}^{-1}$, we can compute an isogeny $E_1 \times E_2 \to \tilde{E}_A \times E$, and we can get ϕ_1 and ϕ_2. Finally, using ϕ_1 and solving Discrete Logarithm Problem by Pohlig-Hellman algorithm [17], we can detect the matrix $\mathbf{B}$.

We explain more details of FESTA trapdoor functions:

Public parameter: Let $d_1, d_2, d_{A,1}, d_{A,2}$ be odd integers such that they are pairwise coprime. Let m_1, m_2, d be integers such that

$$m_1^2 d_{A,1} d_1 + m_2^2 d_{A,2} d_2 = 2^b.$$

Define a prime p as $p = 2^b d_1 d_2 (d_{A,1} d_{A,2})_{\mathsf{sf}} f - 1$, where f is a small positive integer and $(d_{A,1} d_{A,2})_{\mathsf{sf}}$ is the square-free part of $d_{A,1} d_{A,2}$. Let E_0 be a supersingular elliptic curve over $\mathbb{F}_{p^2}$ whose j-invariant is not 1728 or 0. Let $\{P_b, Q_b\}$ be a basis of $E_0[2^b]$. Define $\mathcal{M}_b$ as a commutative subgroup of a 2×2-linear group over $\mathbb{Z}/2^b\mathbb{Z}$.

Public key: We compute a $d_{A,1}$-isogeny $\phi_{A,1}\colon E_0 \to \tilde{E}_A$ and a $d_{A,2}$-isogeny $\phi_{A,2}\colon \tilde{E}_A \to E_A$. Denote $\phi_{A,2}\circ\phi_{A,1}$ by ϕ_A. Take a random matrix $\mathbf{A}$ in $\mathcal{M}_b$. We compute

$$\begin{pmatrix} R_A \\ S_A \end{pmatrix} = \mathbf{A}\begin{pmatrix} \phi_A(P_b) \\ \phi_A(Q_b). \end{pmatrix}$$

Finally, publish (E_A, R_A, S_A) as a public key, and keep $(\mathbf{A}, \phi_{A,1}, \phi_{A,2})$ as a secret.

FESTA trapdoor function: Let ϕ_1 be a d_1-isogeny mapping from E_0 to E_1, and ϕ_2 be a d_2-isogeny mapping from E_A to E_2. Let $\mathbf{B}$ be a matrix in $\mathcal{M}_b$. Compute (R_1, S_1) and (R_2, S_2) such that

$$\begin{pmatrix} R_1 \\ S_1 \end{pmatrix} = \mathbf{B}\begin{pmatrix} \phi_1(P_b) \\ \phi_2(Q_b) \end{pmatrix}, \quad \begin{pmatrix} R_2 \\ S_2 \end{pmatrix} = \mathbf{B}\begin{pmatrix} \phi_2(R_A) \\ \phi_2(S_A) \end{pmatrix}.$$

Output $(E_1, (R_1, S_1), E_2, (R_2, S_2))$.

Inverse map: We first compute ${}^t(R_2', S_2') = \mathbf{A}^{-1}\cdot {}^t(R_2, S_2)$. Since $\mathbf{AB} = \mathbf{BA}$, it holds that

$$\begin{pmatrix} R_2' \\ S_2' \end{pmatrix} = \mathbf{B}\begin{pmatrix} \phi_2(\phi_A(P_b)) \\ \phi_2(\phi_A(Q_b)) \end{pmatrix}.$$

Therefore, from Kani's theorem, the group

$$\langle (m_2 d_{A,2} d_2 R_1, m_1 d_1 R_2'), (m_2 d_{A,2} d_2 S_1, m_1 d_1 S_2') \rangle$$

is the kernel of the $(2^b, 2^b)$-isogeny $\Phi\colon E_1 \times E_2 \to \tilde{E}_A \times E$ defined as

$$\Phi = \begin{pmatrix} m_1\phi_{A,1}\circ\hat{\phi}_1 & -m_2\hat{\phi}_{A,2}\circ\hat{\phi}_2 \\ m_2[\phi_1\circ\hat{\phi}_{A,1}]_*(\phi_2\circ\phi_{A,2}) & m_1[\phi_2\circ\phi_{A,2}]_*(\phi_1\circ\hat{\phi}_{A,1}) \end{pmatrix}.$$

Hence, we can get ϕ_1 and ϕ_2 by computing Φ. If the image of Φ is not a product of two elliptic curves, we output $\perp$. Finally, we compute $(\hat{\phi}_1(R_1), \hat{\phi}_1(S_1))$ and find a matrix $\mathbf{B}$ such that

$$\begin{pmatrix} \hat{\phi}_1(R_1) \\ \hat{\phi}_1(S_1) \end{pmatrix} = d_1\mathbf{B}\begin{pmatrix} P_b \\ Q_b \end{pmatrix}$$

by Pohlig-Hellman algorithm. If $\mathbf{B} \notin \mathcal{M}_b$, we output $\perp$. If $\mathbf{B} \in \mathcal{M}_b$, we output $(\mathbf{B}, \phi_1, \phi_2)$.

3.2 Example for PKE based on FESTA trapdoor functions

This subsection introduces one easy public key encryption scheme based on a FESTA trapdoor function. This example relates to our attack model. See [2] for more secure and concrete PKE schemes based on the functions.

All notations are the same as in the previous subsection. Bob (sender) tries to send a message to Alice (recipient).

Public parameters: Take the same parameters as those of a FESTA trapdoor function. In addition, take one basis $\{P, Q\}$ of $E_0[d_1]$.

Public key: Alice computes ϕ_A and (R_A, S_A), and publishes (E_A, R_A, S_A). She keeps $(\mathbf{A}, \phi_{A,1}, \phi_{A,2})$ as a secret.

Encryption: Bob takes a plaintext μ from $\mathbb{Z}/d_1\mathbb{Z}$. He computes an isogeny ϕ_1 with $\ker \phi_1 = \langle P + \mu Q \rangle$. He takes ϕ_2 and $\mathbf{B}$ at random. He computes $f_{E_A,R_A,S_A}(\mathbf{B}, \phi_1, \phi_2)$ and sends it to Alice as a ciphertext.

Decryption: Alice detects ϕ_1 by computing the inverse map of f_{E_A,R_A,S_A}. It provides a plaintext μ.

4 Adapted attack for FESTA trapdoor functions

In this section, we explain the method to attack FESTA trapdoor functions under our attack model.

4.1 Attack model

In this subsection, we explain the attack model that we consider. We use the FESTA notation (the same notation in Section 3).

The goal of the adversary is to reveal the secret key of the FESTA trapdoor function f_{E_A,R_A,S_A} (*i.e.,* $\phi_{A,1}$, $\phi_{A,2}$, and $\mathbf{A}$).

Let $\{P_1, Q_1\}$ be a basis of $E_1[2^b]$, and $\{P_2, Q_2\}$ be a basis of $E_2[2^b]$. We assume that the adversary can access the following oracle O':

$$O'(E_1, (P_1, Q_1), E_2, (P_2, Q_2)) = \begin{cases} 1 & (\text{if } (E_1 \times E_2)/G \cong \tilde{E}_A \times E) \\ 0 & (\text{otherwise}) \end{cases},$$

where

$$G = \langle (m_2 d_{A,2} d_2 P_1, m_1 d_1 P_2'), (m_2 d_{A,2} d_2 Q_1, m_1 d_1 Q_2') \rangle$$

for ${}^t(P_2', Q_2') = \mathbf{A}^{-1} \cdot {}^t(P_2, Q_2)$.

There are some situations of attacks related to this assumption. For example, we can consider an attack for a public key encryption scheme in Section 3.2 under the following setting:

1. The recipient does not compute $\mathbf{B}$ in the decryption process.
2. The adversary has access to a decryption oracle.

The adversary takes a ciphertext corresponding to $(E_1, (P_1, Q_1), E_2, (P_2, Q_2))$ and sends it to the decryption oracle. If the decryption oracle returns a plaintext μ, the adversary knows $(E_1 \times E_2)/G \cong \tilde{E}_A \times E$, and if it fails to output the correct plaintext μ or refuses the encryption, it knows the ciphertext is incorrect. Therefore, we can construct the oracle O' from the decryption oracle.

From the Kani's theorem, the kernel of the $(2^b, 2^b)$-isogeny $E_1 \times E_2 \to \tilde{E}_A \times E$ is $\langle (m_2 d_{A,2} d_2 P, m_1 \phi_2 \circ \phi_A \circ \hat{\phi}_1(P)) \mid P \in E_1[2^b] \rangle$. Since the number of isomorphism classes of superspecial abelian varieties is $\approx p^3$ if $p \geq 7$ (see [9, Theorem 3.3]), we can define the oracle O that is heuristically equivalent to O' as follows:

$$O(E_1, (P_1, Q_1), E_2, (P_2, Q_2)) = \begin{cases} 1 & \left(\text{if } \begin{pmatrix} P_2' \\ Q_2' \end{pmatrix} = \frac{1}{d_1} \phi_2 \circ \phi_A \circ \hat{\phi}_1 \begin{pmatrix} P_1 \\ Q_1 \end{pmatrix}\right) \\ 0 & (\text{otherwise}) \end{cases},$$

where ${}^t(P_2', Q_2') = \mathbf{A}^{-1} \cdot {}^t(P_2, Q_2)$. Hence, we assume that the adversary can access the oracle O.

Remark 1. One possible countermeasure of the use of the oracle O is to use Weil pairing. Note that we have

$$e_{2^b}(\mathbf{B} \cdot {}^t(\phi_1(P_b), \phi_1(Q_b))) = e_{2^b}(P_b, Q_b)^{\deg \phi_1 \det \mathbf{B}},$$
$$e_{2^b}(\mathbf{B} \cdot {}^t(\phi_2(R_A), \phi_2(S_A)) = e_{2^b}(P_b, Q_b)^{\deg \phi_2 \deg \phi_A \det \mathbf{A} \det \mathbf{B}}.$$

Therefore, a simple strategy of the countermeasure is to check if

$$e_{2^b}(P_1, Q_1)^{\deg \phi_2 \deg \phi_A \det \mathbf{A}} = e_{2^b}(P_2, Q_2)^{\deg \phi_1}.$$

This strategy, however, does not work to prevent the use of the oracle O. By direct computation, if it holds that

$$\mathbf{A}^{-1} \begin{pmatrix} P_2 \\ Q_2 \end{pmatrix} = \frac{1}{\deg \phi_1} \phi_2 \circ \phi_A \circ \hat{\phi}_1 \begin{pmatrix} P_1 \\ Q_1 \end{pmatrix},$$

then we have

$$e_{2^b}(P_1, Q_1)^{\deg \phi_2 \deg \phi_A \deg {\phi_1}^{-1}} = e_{2^b}(P_2, Q_2)^{\det \mathbf{A}^{-1}}.$$

Then, checking by Weil pairing does not work, and the oracle O outputs 1.

4.2 Settings

We use the same notation in Section 3.

Put $\mathbf{A}^{-1}$ as

$$\mathbf{A}^{-1} = \begin{pmatrix} \gamma & \delta \\ \delta & \gamma \end{pmatrix}.$$

Let $\gamma_0, \ldots, \gamma_{b-1}, \delta_0, \ldots, \delta_{b-1}$ are values in $\{0, 1\}$ such that

$$\gamma = \gamma_0 2^0 + \gamma_1 2^1 + \cdots + \gamma_{b-1} 2^{b-1},$$
$$\delta = \delta_0 2^0 + \delta_1 2^1 + \cdots + \delta_{b-1} 2^{b-1}.$$

By Robert's attack [19], detecting $\mathbf{A}^{-1}$ reveals the secret key of the FESTA trapdoor fucntion; therefore, it is suffice to detect values $\gamma_0, \ldots, \gamma_{b-1}, \delta_1, \ldots, \delta_{b-1}$ to attack FESTA trapdoor functions. Thus, we assume that the adversary tries to detect these values instead of the secret key.

We also assume that $\mathcal{M}_b$ is the group of regular circulant matrices over $\mathbb{Z}/2^b\mathbb{Z}$. I.e.,

$$\mathcal{M}_b = \left\{ \begin{pmatrix} \alpha & \beta \\ \beta & \alpha \end{pmatrix} \;\middle|\; \alpha, \beta \in \mathbb{Z}/2^b\mathbb{Z},\ \alpha^2 - \beta^2 \in (\mathbb{Z}/2^b\mathbb{Z})^\times \right\}.$$

Let the correct input be

$$\mathbf{B} \begin{pmatrix} \phi_1(P_b) \\ \phi_1(Q_b) \end{pmatrix}, \quad \mathbf{B} \begin{pmatrix} \phi_2(R_A) \\ \phi_2(S_A) \end{pmatrix}.$$

In our adaptive attack, the adversary sends incorrect inputs based on the above input to the oracle O. Here, the adversary should generate a new input in each case that it accesses to O (and it is possible in our attack strategy). However, for simplicity, we fix the correct input as above in the next subsection.

4.3 Strategy

In this subsection, we explain the strategy of our adaptive attack.

Step 0: The goal of this step is to reveal γ_0, δ_0 and $2^{b-1}\phi_A(P_b)$. Algorithm 1 shows the outline of this step.

The adversary takes a random point $P_0 \in E_A$ of order 2, and sends to the oracle O

$$\mathbf{B}\begin{pmatrix}\phi_1(P_b)\\ \phi_1(Q_b)\end{pmatrix}, \quad \mathbf{B}\begin{pmatrix}\phi_2(R_A)\\ \phi_2(S_A)\end{pmatrix} + \begin{pmatrix}\phi_2(P_0)\\ \phi_2(P_0)\end{pmatrix}$$

instead of the correct input. Then, it holds that

$$\mathbf{A}^{-1}\left(\mathbf{B}\begin{pmatrix}\phi_2(R_A)\\ \phi_2(S_A)\end{pmatrix} + \begin{pmatrix}\phi_2(P_0)\\ \phi_2(P_0)\end{pmatrix}\right) = \mathbf{B}\begin{pmatrix}\phi_2(\phi_A(P_b))\\ \phi_2(\phi_A(Q_b))\end{pmatrix} + (\gamma+\delta)\begin{pmatrix}\phi_2(P_0)\\ \phi_2(P_0)\end{pmatrix}.$$

Therefore, if $\gamma_0 + \delta_0 \equiv 0 \pmod 2$, then O outputs 1, and if $\gamma_0 + \delta_0 \equiv 1 \pmod 2$, then O outputs 0.

The adversary takes three points $P_0, P_1, P_2 \in E_A$ of order 2. The adversary tries to judge which of three points is $2^{b-1}\phi_A(P_b)$.

Case of $\gamma_0 + \delta_0 \equiv 0 \pmod 2$: From the definition of the matrix $\mathbf{A}$, we have $\gamma_0 = \delta_0 = 1$. The adversary sends to the oracle O

$$\mathbf{B}\begin{pmatrix}\phi_1(P_b)\\ \phi_1(Q_b)\end{pmatrix} + \begin{pmatrix}2^{b-1}\phi_1(P_b)\\ 2^{b-1}\phi_1(P_b)\end{pmatrix}, \quad \mathbf{B}\begin{pmatrix}\phi_2(R_A)\\ \phi_2(S_A)\end{pmatrix} + \begin{pmatrix}\phi_2(P_i)\\ 0\end{pmatrix}$$

for $i = 0, 1, 2$. Then we have

$$\mathbf{A}^{-1}\left(\mathbf{B}\begin{pmatrix}\phi_2(R_A)\\ \phi_2(S_A)\end{pmatrix} + \begin{pmatrix}\phi_2(P_i)\\ 0\end{pmatrix}\right) = \mathbf{B}\begin{pmatrix}\phi_2(\phi_A(P_b))\\ \phi_2(\phi_A(Q_b))\end{pmatrix} + \begin{pmatrix}\phi_2(P_i)\\ \phi_2(P_i)\end{pmatrix}$$

Therefore, if P_i is $2^{b-1}\phi_A(P_b)$, then O outputs 1, and if $P_i \neq 2^{b-1}\phi_A(P_b)$, then O outputs 0. Therefore, by sending three incorrect inputs to the oracle O, the adversary can guess $2^{b-1}\phi_A(P_b)$.

Case of $\gamma_0 + \delta_0 \equiv 1 \pmod 2$: The adversary sends to the oracle

$$\mathbf{B}\begin{pmatrix}\phi_1(P_b)\\ \phi_1(Q_b)\end{pmatrix} + \begin{pmatrix}2^{b-1}\phi_1(P_b)\\ 0\end{pmatrix}, \quad \mathbf{B}\begin{pmatrix}\phi_2(R_A)\\ \phi_2(S_A)\end{pmatrix} + \begin{pmatrix}\phi_2(P_i)\\ 0\end{pmatrix}$$

for $i = 0, 1, 2$. Then if $\gamma_0 = 1$, we have

$$\mathbf{A}^{-1}\left(\mathbf{B}\begin{pmatrix}\phi_2(R_A)\\ \phi_2(S_A)\end{pmatrix} + \begin{pmatrix}\phi_2(P_i)\\ 0\end{pmatrix}\right) = \mathbf{B}\begin{pmatrix}\phi_2(\phi_A(P_b))\\ \phi_2(\phi_A(Q_b))\end{pmatrix} + \begin{pmatrix}\phi_2(P_i)\\ 0\end{pmatrix},$$

and if $\delta_0 = 1$, we have

$$\mathbf{A}^{-1}\left(\mathbf{B}\begin{pmatrix}\phi_2(R_A)\\ \phi_2(S_A)\end{pmatrix}+\begin{pmatrix}\phi_2(P_i)\\ 0\end{pmatrix}\right)=\mathbf{B}\begin{pmatrix}\phi_2(\phi_A(P_b))\\ \phi_2(\phi_A(Q_b))\end{pmatrix}+\begin{pmatrix}0\\ \phi_2(P_i)\end{pmatrix}.$$

Therefore, if $\gamma_0 = 1$ and $P_i = 2^{b-1}\phi_A(P_b)$, then O outputs 1, and if not, it outputs 0. Therefore, the adversary gets (γ_0, δ_0) and $2^{b-1}\phi_A(P_b)$ if $\gamma_0 = 1$ by sending three incorrect inputs to O. If the oracle O outputs 0 for every three inputs, it holds that $\gamma_0 = 0$. In this case, the adversary additionally sends to the oracle O

$$\mathbf{B}\begin{pmatrix}\phi_1(P_b)\\ \phi_1(Q_b)\end{pmatrix}+\begin{pmatrix}0\\ 2^{b-1}\phi_1(P_b)\end{pmatrix},\quad \mathbf{B}\begin{pmatrix}\phi_2(R_A)\\ \phi_2(S_A)\end{pmatrix}+\begin{pmatrix}\phi_2(P_i)\\ 0\end{pmatrix}$$

for $i = 0, 1, 2$. Then it holds that

$$\mathbf{A}^{-1}\left(\mathbf{B}\begin{pmatrix}\phi_2(R_A)\\ \phi_2(S_A)\end{pmatrix}+\begin{pmatrix}\phi_2(P_i)\\ 0\end{pmatrix}\right)=\mathbf{B}\begin{pmatrix}\phi_2(\phi_A(P_b))\\ \phi_2(\phi_A(Q_b))\end{pmatrix}+\begin{pmatrix}0\\ \phi_2(P_i)\end{pmatrix}.$$

Therefore, the adversary can guess $2^{b-1}\phi_1(P_b)$ by sending three incorrect inputs. Consequently, the adversary can get (γ_0, δ_0) and $2^{b-1}\phi_1(P_b)$ by sending six incorrect inputs.

From the above processes, the adversary reveals γ_0, δ_0 and $2^{b-1}\phi_A(P_b)$ by sending at most seven incorrect inputs to O.

Step k ($1 \leq k \leq b-1$): The goal of this step is to reveal γ_k, δ_k and $2^{b-k-1}\phi_A(P_b)$. We assume that the adversary already knows $\gamma_0, \ldots, \gamma_{k-1}, \delta_0, \ldots, \delta_{k-1}$ and $2^{b-k-2}\phi_A(P_b)$. The outline of this step is in Algorithm 2.

The adversary defines a 2×2-matrix $\mathbf{C}_k$ over $\mathbb{Z}/2^b\mathbb{Z}$ as

$$\mathbf{C}_k = \sum_{i=0}^{k-1} 2^i \begin{pmatrix}\gamma_i & \delta_i\\ \delta_i & \gamma_i\end{pmatrix}.$$

It holds that

$$\mathbf{A}^{-1} = \mathbf{C}_k + 2^k \begin{pmatrix}\gamma_k & \delta_k\\ \delta_k & \gamma_k\end{pmatrix} + 2^{k+1}\mathbf{D},$$

where $\mathbf{D}$ is a 2×2-matrix over $\mathbb{Z}/2^b\mathbb{Z}$. Therefore, we have

$$2^{b-k-1}\mathbf{I}_2 = 2^{b-k-1}\mathbf{C}_k\mathbf{A} + 2^{b-1}\begin{pmatrix}\gamma_k & \delta_k\\ \delta_k & \gamma_k\end{pmatrix}\mathbf{A},$$

where $\mathbf{I}_2$ is the identity matrix. Hence, it holds that

$$2^{b-k-1}\begin{pmatrix}\phi_A(P_b)\\ \phi_A(Q_b)\end{pmatrix} = 2^{b-k-1}\mathbf{C}_k\begin{pmatrix}R_A\\ S_A\end{pmatrix} + 2^{b-1}\begin{pmatrix}\gamma_k & \delta_k\\ \delta_k & \gamma_k\end{pmatrix}\begin{pmatrix}R_A\\ S_A\end{pmatrix}.$$

From this equality, the adversary can get $2^{b-k-1}\phi_A(P_b)$ if it knows γ_k and δ_k.

From now on, we assume that the adversary tries to reveal γ_k and δ_k. The adversary takes four points P_0, P_1, P_2, P_3 of order 2^{k+1} such that $2P_i = 2^{b-k-2}\phi_A(P_b)$ for $i = 0, 1, 2, 3$. The adversary sends to the oracle O four incorrect inputs

$$\mathbf{B}\begin{pmatrix}\phi_1(P_b)\\ \phi_1(Q_b)\end{pmatrix} + \mathbf{C}_k 2^{b-k-1}\begin{pmatrix}\phi_1(P_b)\\ \phi_1(P_b)\end{pmatrix}, \quad \mathbf{B}\begin{pmatrix}\phi_2(R_A)\\ \phi_2(S_A)\end{pmatrix} + \begin{pmatrix}\phi_2(P_i)\\ \phi_2(P_i)\end{pmatrix}$$

for $i = 0, 1, 2, 3$. Then, it holds that

$$\begin{aligned}&\mathbf{A}^{-1}\left(\mathbf{B}\begin{pmatrix}\phi_2(R_A)\\ \phi_2(S_A)\end{pmatrix} + \begin{pmatrix}\phi_2(P_i)\\ \phi_2(P_i)\end{pmatrix}\right)\\ &= \mathbf{B}\begin{pmatrix}\phi_2(\phi_A(P_b))\\ \phi_2(\phi_A(Q_b))\end{pmatrix} + (\gamma_k + \delta_k)2^k\begin{pmatrix}\phi_2(P_i)\\ \phi_2(P_i)\end{pmatrix} + \mathbf{C}_k\begin{pmatrix}\phi_2(P_i)\\ \phi_2(P_i)\end{pmatrix}.\end{aligned}$$

If $\gamma_k + \delta_k \equiv 0 \pmod 2$ and $P_i = 2^{b-k-1}\phi_A(P_b)$, then O outputs 1, and if not, it outputs 0. Hence, if O outputs 1 for one of four incorrect inputs, it holds that $\gamma_k + \delta_k \equiv 0 \pmod 2$, and if it outputs 0 for all these inputs, then $\gamma_k + \delta_k \equiv 1 \pmod 2$. Consequently, the adversary can detect $\gamma_k + \delta_k \mod 2$ by sending the above four incorrect inputs to O.

Case of $\gamma_k + \delta_k \equiv 0 \pmod 2$: The adversary sends to the oracle O

$$\mathbf{B}\begin{pmatrix}\phi_1(P_b)\\ \phi_1(Q_b)\end{pmatrix} + \mathbf{C}_k 2^{b-k-1}\begin{pmatrix}\phi_1(P_b)\\ 0\end{pmatrix}, \quad \mathbf{B}\begin{pmatrix}\phi_2(R_A)\\ \phi_2(S_A)\end{pmatrix} + \begin{pmatrix}\phi_2(P_i)\\ 0\end{pmatrix}$$

for $i = 0, 1, 2, 3$. Note that $\gamma_k = \delta_k$. Then, we have

$$\begin{aligned}&\mathbf{A}^{-1}\left(\mathbf{B}\begin{pmatrix}\phi_2(R_A)\\ \phi_2(S_A)\end{pmatrix} + \begin{pmatrix}\phi_2(P_i)\\ 0\end{pmatrix}\right)\\ &= \mathbf{B}\begin{pmatrix}\phi_2(\phi_A(P_b))\\ \phi_2(\phi_A(Q_b))\end{pmatrix} + \gamma_k 2^k\begin{pmatrix}\phi_2(P_i)\\ \phi_2(P_i)\end{pmatrix} + \mathbf{C}_k\begin{pmatrix}\phi_2(P_i)\\ 0\end{pmatrix}.\end{aligned}$$

If $\gamma_k = 0$ and $P_i = 2^{b-k-1}\phi_A(P_b)$, then O outputs 1, and if not, it outputs 0. Therefore, if O outputs 0 for all these incorrect inputs, then $\gamma_k = \delta_k = 1$, and if it outputs 1 for one of them, it holds that $\gamma_k = \delta_k = 0$. Hence, the adversary can reveal γ_k and δ_k by sending four incorrect inputs to O.

Case of $\gamma_k + \delta_k \equiv 1 \pmod 2$: The adversary sends to the oracle O

$$\begin{aligned}&\mathbf{B}\begin{pmatrix}\phi_1(P_b)\\ \phi_1(Q_b)\end{pmatrix} + 2^{b-1}\begin{pmatrix}\phi_1(P_b)\\ 0\end{pmatrix} + \mathbf{C}_k 2^{b-k-1}\begin{pmatrix}\phi_1(P_b)\\ 0\end{pmatrix},\\ &\mathbf{B}\begin{pmatrix}\phi_2(R_A)\\ \phi_2(S_A)\end{pmatrix} + \begin{pmatrix}\phi_2(P_i)\\ 0\end{pmatrix}\end{aligned}$$

for $i = 0, 1, 2, 3$. Then it holds that

$$\begin{aligned}&\mathbf{A}^{-1}\left(\mathbf{B}\begin{pmatrix}\phi_2(R_A)\\ \phi_2(S_A)\end{pmatrix} + \begin{pmatrix}\phi_2(P_i)\\ 0\end{pmatrix}\right)\\ &= \mathbf{B}\begin{pmatrix}\phi_2(\phi_A(P_b))\\ \phi_2(\phi_A(Q_b))\end{pmatrix} + 2^k\begin{pmatrix}\gamma_k\phi_1(P_i)\\ \delta_k\phi_1(P_i)\end{pmatrix} + \mathbf{C}_k\begin{pmatrix}\phi_2(P_i)\\ 0\end{pmatrix}.\end{aligned}$$

Therefore, the oracle O ouputs 1 if and only if $\gamma_k = 1$ and $P_i = 2^{b-k-1}\phi_A(P_b)$. Hence, if O outputs 0 for all four incorrect inputs, then $\gamma_k = 0$ and $\delta_k = 1$, and if it outputs 1 for one of them, then $\gamma_k = 1$ and $\delta_k = 0$. Thus, the adversary can reveal γ_k and δ_k by sending four incorrect inputs to O.

In summary, the adversary can get γ_k and δ_k (and $2^{b-k-1}\phi_A(P_b)$) by sending eight incorrect inputs to O.

From the above steps, the adversary can know the matrix $\mathbf{A}$ in at most $8b-1$ queries to the oracle O.

Remark 2. As noted in Section 4.1, we need to assume that the recipient does not compute $\mathbf{B}$ in the decryption process. Indeed, the incorrect input

$$\mathbf{B}\begin{pmatrix}\phi_1(P_b)\\ \phi_1(Q_b)\end{pmatrix} + \mathbf{C}_k 2^{b-k-1}\begin{pmatrix}\phi_1(P_b)\\ \phi_1(P_b)\end{pmatrix}, \quad \mathbf{B}\begin{pmatrix}\phi_2(R_A)\\ \phi_2(S_A)\end{pmatrix} + \begin{pmatrix}\phi_2(P_i)\\ \phi_2(P_i)\end{pmatrix}$$

leads the matrix

$$\mathbf{B} + 2^{b-k-1}\mathbf{C}_k\begin{pmatrix}1 & 0\\ 1 & 0\end{pmatrix},$$

and this matrix does not belong to $\mathcal{M}_b$.

It is future work to research the existence of an adaptive attack even if Alice checks whether $\mathbf{B} \in \mathcal{M}_b$.

5 Conclusion

We showed that an adaptive attack might be considered if the FESTA trapdoor function was used in the wrong way. This attack reveals its secret key.

For our attack, we need an oracle that decides whether one who knows the secret key of FESTA trapdoor functions can compute the correct $(2^b, 2^b)$-isogeny from the given input. This oracle relates to the situation that a FESTA trapdoor function is used as a specific PKE scheme, an adversary has access to the decryption oracle, and a recipient does not check if $\mathbf{B} \in \mathcal{M}_b$ in the decryption process. We can prevent this attack by checking if $\mathbf{B} \in \mathcal{M}_b$ in the decryption process. Therefore, the IND-CCA secure PKE schemes named FESTA proposed in [2] is not attacked by our adaptive attack.

Acknowledgements.

The authors would like to thank Andrea Basso, Luciano Maino, and Giacomo Pope for an important comment on this research. This work was supported by EPSRC through grant EP/V011324/1.

References

1. Reza Azarderakhsh, Matthew Campagna, Craig Costello, LD Feo, Basil Hess, A Jalali, D Jao, B Koziel, B LaMacchia, P Longa, et al. Supersingular isogeny key encapsulation. *Submission to the NIST Post-Quantum Standardization project*, 2017.
2. Andrea Basso, Luciano Maino, and Giacomo Pope. FESTA: Fast encryption from supersingular torsion attacks, 2023. `https://ia.cr/2023/660`.
3. Wouter Castryck and Thomas Decru. An efficient key recovery attack on SIDH. In *Advances in Cryptology – EUROCRYPT 2023*, pages 423–447. Springer, 2023.
4. Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: an efficient post-quantum commutative group action. In *Advances in Cryptology – ASIACRYPT 2018*, pages 395–427. Springer, 2018.
5. Pierrick Dartois, Antonin Leroux, Damien Robert, and Benjamin Wesolowski. SQISignHD: New dimensions in cryptography, 2023. `https://ia.cr/2023/436`.
6. Luca De Feo, David Kohel, Antonin Leroux, Christophe Petit, and Benjamin Wesolowski. SQISign: compact post-quantum signatures from quaternions and isogenies. In *Advances in Cryptology – ASIACRYPT 2020*, pages 64–93. Springer, 2020.
7. Tako Boris Fouotsa and Christophe Petit. A new adaptive attack on SIDH. In *Topics in Cryptology – CT-RSA 2022*, pages 322–344. Springer, 2022.
8. Steven D Galbraith, Christophe Petit, Barak Shani, and Yan Bo Ti. On the security of supersingular isogeny cryptosystems. In *Advances in Cryptology – ASIACRYPT 2016*, pages 63–91. Springer, 2016.
9. Tomoyoshi Ibukiyama, Toshiyuki Katsura, and Frans Oort. Supersingular curves of genus two and class numbers. *Compositio Mathematica*, 57(2):127–152, 1986.
10. David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In *Post-Quantum Cryptography – PQCrypto 2011*, pages 19–34. Springer, 2011.
11. Ernst Kani. The number of curves of genus two with elliptic differentials. 1997.
12. Neal Koblitz. Elliptic curve cryptosystems. *Mathematics of computation*, pages 203–209, 1987.
13. Luciano Maino, Chloe Martindale, Lorenz Panny, Giacomo Pope, and Benjamin Wesolowski. A direct key recovery attack on SIDH. In *Advances in Cryptology – EUROCRYPT 2023*, pages 448–471. Springer, 2023.
14. Victor S Miller. Use of elliptic curves in cryptography. In *Advances in Cryptology – CRYPTO '85*, pages 417–426. Springer, 1985.
15. David Mumford, Chidambaran Padmanabhan Ramanujam, and Jurij Ivanovič Manin. *Abelian varieties*, volume 5. Oxford university press Oxford, 1974.
16. National Institute of Standards and Technology. Post–quantum cryptography standardization. `https://csrc.nist.gov/Projects/post-quantum-cryptography/round-4-submissions`.
17. Stephen Pohlig and Martin Hellman. An improved algorithm for computing logarithms over $GF(p)$ and its cryptographic significance. *IEEE Transactions on information Theory*, 24(1):106–110, 1978.
18. Ronald L Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, pages 120–126, 1978.
19. Damien Robert. Breaking SIDH in polynomial time. In *Advances in Cryptology – EUROCRYPT 2023*, pages 472–503. Springer, 2023.

20. Peter W Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science – FOCS '94*, pages 124–134. IEEE, 1994.
21. Joseph H Silverman. *The arithmetic of elliptic curves*, volume 106. Springer Science & Business Media, 2009.
22. Benjamin Smith. *Explicit endomorphisms and correspondences*. PhD thesis, University of Sydney, 2005.
23. Jacques Vélu. Isogénies entre courbes elliptiques. *CR Acad. Sci. Paris Sér. A*, 273(5):238–241, 1971.

Algorithm 1 Step 0 of our adaptive attack for FESTA trapdoor functions

Require: The public parameter and public key of FESTA $(p, E_0, P_b, Q_b, E_A, R_A, S_A)$
Ensure: $\gamma_0, \delta_0 \in \{0,1\}$ and $2^{b-1}\phi_A(P_b)$

$P_0, P_1, P_2 \leftarrow$ points of order 2 in E_A
$\mathbf{B} \leftarrow$ a random matrix in $\mathcal{M}_b$
$\phi_1 \leftarrow$ a random d_1-isogeny mapping from E_0 to E_1
$\phi_2 \leftarrow$ a random d_2-isogeny mapping from E_A to E_2
$k \leftarrow O\left(E_1, \mathbf{B}\begin{pmatrix}\phi_1(P_b)\\ \phi_1(Q_b)\end{pmatrix}, E_2, \mathbf{B}\begin{pmatrix}\phi_2(R_A)\\ \phi_2(S_A)\end{pmatrix} + \begin{pmatrix}\phi_2(P_0)\\ \phi_2(P_0)\end{pmatrix}\right)$
if $k = 1$ **then**
  $\gamma_0, \delta_0 \leftarrow 1, 1$
  for $i = 0, 1, 2$ **do**
    $l \leftarrow O\left(E_1, \mathbf{B}\begin{pmatrix}\phi_1(P_b)\\ \phi_1(Q_b)\end{pmatrix} + \begin{pmatrix}2^{b-1}\phi_1(P_b)\\ 2^{b-1}\phi_1(P_b)\end{pmatrix}, E_2, \mathbf{B}\begin{pmatrix}\phi_2(R_A)\\ \phi_2(S_A)\end{pmatrix} + \begin{pmatrix}\phi_2(P_i)\\ 0\end{pmatrix}\right)$
    if $l = 1$ **then**
      $P \leftarrow P_i$
    end if
  end for
else
  $\gamma_0 \leftarrow 0$
  for $i = 0, 1, 2$ **do**
    $l \leftarrow O\left(E_1, \mathbf{B}\begin{pmatrix}\phi_1(P_b)\\ \phi_1(Q_b)\end{pmatrix} + \begin{pmatrix}2^{b-1}\phi_1(P_b)\\ 0\end{pmatrix}, E_2, \mathbf{B}\begin{pmatrix}\phi_2(R_A)\\ \phi_2(S_A)\end{pmatrix} + \begin{pmatrix}\phi_2(P_i)\\ 0\end{pmatrix}\right)$
    if $l = 1$ **then**
      $P \leftarrow P_i$
      $\gamma_0, \delta_0 \leftarrow 1, 0$
    end if
  end for
  if $\gamma_0 = 0$ **then**
    $\gamma_0, \delta_0 \leftarrow 0, 1$
    for $i = 0, 1, 2$ **do**
      $l \leftarrow O\left(E_1, \mathbf{B}\begin{pmatrix}\phi_1(P_b)\\ \phi_1(Q_b)\end{pmatrix} + \begin{pmatrix}0\\ 2^{b-1}\phi_1(P_b)\end{pmatrix}, E_2, \mathbf{B}\begin{pmatrix}\phi_2(R_A)\\ \phi_2(S_A)\end{pmatrix} + \begin{pmatrix}\phi_2(P_i)\\ 0\end{pmatrix}\right)$
      if $l = 1$ **then**
        $P \leftarrow P_i$
      end if
    end for
  end if
end if
return γ_0, δ_0, P

Algorithm 2 Step k of our adaptive attack for FESTA

Require: The public parameter and public key of FESTA $(p, E_0, P_b, Q_b, E_A, R_A, S_A)$ and $\gamma_0, \ldots, \gamma_{k-1}$ and $\delta_0, \ldots, \delta_{k-1}$ and $2^{b-k-2}\phi_A(P_b)$

Ensure: $\gamma_k, \delta_k \in \{0, 1\}$ and $2^{b-k-1}\phi_A(P_b)$

$P_0, P_1, P_2, P_3 \leftarrow$ points in E_A such that $2P_i = 2^{b-k-2}\phi_A(P_b)$

$\mathbf{C}_k \leftarrow \sum_{i=0}^{k-1} 2^i \begin{pmatrix} \gamma_i & \delta_i \\ \delta_i & \gamma_i \end{pmatrix}$

$\mathbf{B} \leftarrow$ a random matrix in $\mathcal{M}_b$

$\phi_1 \leftarrow$ a random d_1-isogeny mapping from E_0 to E_1

$\phi_2 \leftarrow$ a random d_2-isogeny mapping from E_A to E_2

$\epsilon \leftarrow 1$

for $i = 0, 1, 2, 3$ **do**

  $k \leftarrow O\left(E_1, \mathbf{B}\begin{pmatrix}\phi_1(P_b)\\ \phi_1(Q_b)\end{pmatrix} + \mathbf{C}_k 2^{b-k-1}\begin{pmatrix}\phi_1(P_b)\\ \phi_1(P_b)\end{pmatrix}, E_2, \mathbf{B}\begin{pmatrix}\phi_2(R_A)\\ \phi_2(S_A)\end{pmatrix} + \begin{pmatrix}\phi_2(P_i)\\ \phi_2(P_i)\end{pmatrix}\right)$

  if $k = 1$ **then**

    $\epsilon \leftarrow 0$

  end if

end for

if $\epsilon = 0$ **then**

  $\gamma_k, \delta_k \leftarrow 1, 1$

  for $i = 0, 1, 2, 3$ **do**

    $k \leftarrow O\left(E_1, \mathbf{B}\begin{pmatrix}\phi_1(P_b)\\ \phi_1(Q_b)\end{pmatrix} + \mathbf{C}_k 2^{b-k-1}\begin{pmatrix}\phi_1(P_b)\\ 0\end{pmatrix}, E_2, \mathbf{B}\begin{pmatrix}\phi_2(R_A)\\ \phi_2(S_A)\end{pmatrix} + \begin{pmatrix}\phi_2(P_i)\\ 0\end{pmatrix}\right)$

    if $k = 1$ **then**

      $\gamma_k, \delta_k \leftarrow 0, 0$

    end if

  end for

else

  $\gamma_k, \delta_k \leftarrow 0, 1$

  for $i = 0, 1, 2, 3$ **do**

    $k \leftarrow O\left(E_1, \mathbf{B}\begin{pmatrix}\phi_1(P_b)\\ \phi_1(Q_b)\end{pmatrix} + 2^{b-1}\mathbf{C}_k 2^{b-k-1}\begin{pmatrix}\phi_1(P_b)\\ 0\end{pmatrix} + \mathbf{C}_k 2^{b-k-1}\begin{pmatrix}\phi_1(P_b)\\ 0\end{pmatrix}, E_2, \mathbf{B}\begin{pmatrix}\phi_2(R_A)\\ \phi_2(S_A)\end{pmatrix} + \begin{pmatrix}\phi_2(P_i)\\ 0\end{pmatrix}\right)$

    if $k = 1$ **then**

      $\gamma_k, \delta_k \leftarrow 1, 0$

    end if

  end for

end if

$\begin{pmatrix} P \\ Q \end{pmatrix} \leftarrow 2^{b-k-1}\mathbf{C}_k \begin{pmatrix} R_A \\ S_A \end{pmatrix} + 2^{b-1}\begin{pmatrix} \gamma_k & \delta_k \\ \delta_k & \gamma_k \end{pmatrix}\begin{pmatrix} R_A \\ S_A \end{pmatrix}$

return γ_k, δ_k, P