The wrong use of a FESTA trapdoor function leads to an adaptive attack

Tomoki Moriya[1] and Hiroshi Onuki[2]

[1] School of Computer Science, University of Birmingham, UK
t.moriya@bham.ac.uk
[2] Department of Mathematical Informatics, The University of Tokyo, Japan
onuki@mist.i.u-tokyo.ac.jp

Abstract. Isogeny-based cryptography is one of the candidates for post-quantum cryptography. In 2023, Kani's theorem breaks an isogeny-based scheme SIDH, which was considered a promising post-quantum scheme. Though Kani's theorem damaged isogeny-based cryptography, some researchers have been trying to dig into the applications of this theorem. A FESTA trapdoor function is an isogeny-based trapdoor function that is one trial to apply Kani's theorem to cryptography.
This paper claims that there is an adaptive attack for a FESTA-based scheme if this scheme does not check the correctness of the input matrix. Our attack cannot be adapted to IND-CCA PKE schemes named FESTA proposed in the FESTA original paper so far. In this paper, we provide an adaptive attack for a FESTA trapdoor function using a specific oracle, and it reveals the secret key of the function. This oracle may be constructed if the FESTA trapdoor function is used in the wrong way (*i.e.,* without the checking process of the input matrix). As an example, we explain that our attack can be adapted to a possible PKE scheme based on a FESTA trapdoor function in the wrong way.

Keywords: Isogeny-based cryptography · FESTA · Kani's theorem · adaptive attack

1 Introduction

Public key cryptography is an important technology for the security of our information society. For example, we use RSA [19] and Elliptic Curve Cryptography [14,12] to prevent the leakage of our crucial information. However, Shor showed that quantum computers may be able to break these cryptosystems in polynomial time [21]. Therefore, we need to construct novel cryptosystems to resist the attacks via 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 [17].

In 2022, some studies break SIDH and cryptosystems related to SIDH [3,13,20]. 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 have not been broken by these attacks, it caused some damages 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 a FESTA trapdoor function is used in the wrong way. There are several studies of adaptive attacks for SIDH (*e.g.,* [8,7]). We construct a similar attack for a FESTA trapdoor function and related schemes when the function is used in the wrong way.

As a part of the input of a FESTA trapdoor function, we use a 2×2 regular matrix belonging to a fixed set $\mathcal{M}_b$. We usually check whether this 2×2 matrix is taken from the set $\mathcal{M}_b$ when computing the inverse map of the trapdoor function. We show that we can construct a specific oracle O for a possible scheme not checking the inclusivity of this matrix, and an adversary can reveal the secret key of the FESTA trapdoor functions by using this oracle.

Note that FESTA is not threatened by our attack directly because we check the correctness of the ciphertexts in its description process.

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 [16] and [22] for more detail.

Let k be a field. We denote the characteristic of k by $\mathrm{ch}(k)$. Let A be an abelian variety over k with an isomorphism $\varphi\colon A \to \hat{A}$, where $\hat{A}$ is the dual abelian variety of A. We call the pair (A, φ) a principally polarised abelian variety over k. In this paper, we often omit the polarisation φ and represent (A, φ) by A. An elliptic curve is a principally polarised abelian variety of dimension 1. 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 a principally polarised abelian variety A satisfies, as an abelian variety, $A \cong \prod_{i=1}^{\dim A} E_i$ for supersingular elliptic curves $E_1, \ldots, E_{\dim A}$, we call A a superspecial principally polarised abelian variety.

Let A and B be principally polarised 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, the image variety B is unique up to isomorphism. We denote by A/G a representative of the 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.,* [24] and [23]). For an isogeny $\phi\colon A \to B$, there is an isogeny $\hat{\phi}$ satisfying $\hat{\phi} \circ \phi = \deg\phi$ and $\phi \circ \hat{\phi} = \deg\phi$. We call $\hat{\phi}$ the dual isogeny of ϕ.

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 commutative 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.* $\ker[\phi_1]_*\phi_2 = \phi_1(G_2)$*) satisfying $([\phi_2]_*\phi_1) \circ \phi_2 = ([\phi_1]_*\phi_2) \circ \phi_1$. 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*

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

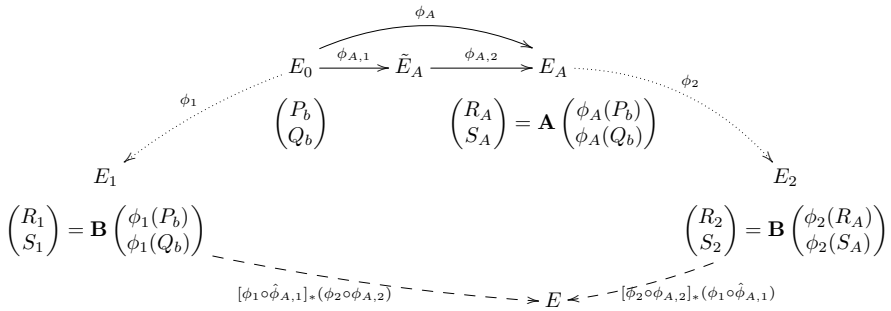
with $\ker \Phi = \langle (\deg \phi_1 P, \phi_1 \circ \hat{\phi}_2(P)) \mid P \in E_2[\deg \phi_1 + \deg \phi_2] \rangle$.

3 FESTA trapdoor function

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

3.1 Construction

The following diagram shows the outline of a FESTA trapdoor function. The public parameter is E_0 and (P_b, Q_b), where E_0 is a supersingular elliptic curve and (P_b, Q_b) is a basis of $E_0[2^b]$. The symbols $\mathbf{A}$ and $\mathbf{B}$ represent 2×2 matrices.



We first provide a brief explanation. Notations are the same as in the above diagram. To set up the trapdoor function, we 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 the general linear group $\mathrm{GL}_2(\mathbb{Z}/2^b\mathbb{Z})$. For example, we can use the set of regular circulant matrices for $\mathcal{M}_b$. 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 $\mathbf{A}$, 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 the Discrete Logarithm Problem via the Pohlig-Hellman algorithm [18], we can detect the matrix $\mathbf{B}$.

We explain more details of a FESTA trapdoor function:

Public parameter: Let λ be a security parameter. Let $d_1, d_2, d_{A,1}, d_{A,2}, m_1, m_2$ be odd integers such that they are pairwise coprime, $d_1, d_2, (d_{A,1}d_{A,2}) > 2^{2\lambda}$, and there is an integer b with $b > 3\lambda$ satisfying

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

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 $\mathbf{GL}_2(\mathbb{Z}/2^b\mathbb{Z})$.

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

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

Since, the integers $m_1 d_{A,1}$ and $m_2 d_{A,2}$ are coprime to d_1 and d_2 respectively, we have

$$\ker \phi_1 = \hat{\phi}_{A,1} \circ (m_1 \phi_{A,1} \circ \hat{\phi}_1)(E_1[d_1]), \ \ker \phi_2 = \phi_{A,2} \circ (-m_1 \hat{\phi}_{A,2} \circ \hat{\phi}_2)(E_2[d_2]).$$

Hence, we can get ϕ_1 and ϕ_2 by computing the images of $E_1[d_1] \times \{0\}$ and $\{0\} \times E_2[d_2]$ under 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 the 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 a FESTA trapdoor function

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 the 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 Attack model

In this section, 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 $\mathbf{A}$).

4.1 Oracles for the attack

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

Furthermore, we can make the oracle O'' simpler by the following proposition.

Proposition 1. *Let O'' be the oracle defined as above, and $\phi_1 : E_0 \to E_1$ and $\phi_2 : E_A \to E_2$ be isogenies of degree d_1 and d_2, respectively. Define $P_1 = \phi_1(P_b)$, $Q_1 = \phi_1(Q_b)$, $P_2 = \phi_2(P_A)$, and $Q_2 = \phi_2(Q_A)$. Then for any matrices $\mathbf{B}, \mathbf{B}' \in \mathrm{GL}_2(\mathbb{Z}/2^b\mathbb{Z})$,*

$$O''(E_1, (P_1, Q_1)^t\mathbf{B}', E_2, (P_2, Q_2)^t\mathbf{B}'') = 1$$

if and only if $\mathbf{A}\mathbf{B}' = \mathbf{B}''\mathbf{A}$.

Proof. By the definition, $O''(E_1, (P_1, Q_1)^t\mathbf{B}', E_2, (P_2, Q_2)^t\mathbf{B}'') = 1$ if and only if

$$\mathbf{A}^{-1}\mathbf{B}'' \begin{pmatrix} P_2 \\ Q_2 \end{pmatrix} = \frac{1}{d_1} \phi_2 \circ \phi_A \circ \hat{\phi}_1 \mathbf{B}' \begin{pmatrix} P_1 \\ Q_1 \end{pmatrix}.$$

The latter equation is equivalent to

$$\mathbf{A}^{-1}\mathbf{B}''\mathbf{A}\phi_2 \circ \phi_A \begin{pmatrix} P_b \\ Q_b \end{pmatrix} = \mathbf{B}'\phi_2 \circ \phi_A \begin{pmatrix} P_b \\ Q_b \end{pmatrix}.$$

Since $(\phi_2 \circ \phi_A(P_b), \phi_2 \circ \phi_A(Q_b))$ is a basis of $E_2[2^b]$, the last equation is equivalent to $\mathbf{A}\mathbf{B}' = \mathbf{B}''\mathbf{A}$.

By the above proposition, we obtain an oracle that returns whether $\mathbf{A}\mathbf{B}' = \mathbf{B}''\mathbf{A}$ or not. We denote this by O. We assume that the adversary can access the oracle O.

Remark 1. One may think that we can counter our attack model by using the Weil pairing. Note that we have

$$e_{2^b}(\mathbf{B}' \cdot {}^t(\phi_1(P_b), \phi_1(Q_b))) = e_{2^b}(P_b, Q_b)^{\deg \phi_1 \det \mathbf{B}'},$$
$$e_{2^b}(\mathbf{B}'' \cdot {}^t(\phi_2(R_A), \phi_2(S_A)) = e_{2^b}(P_b, Q_b)^{\deg \phi_2 \deg \phi_A \det \mathbf{A} \det \mathbf{B}''}$$

for the Weil pairing e_{2^b}. Therefore, a simple strategy of the countermeasure is to check whether

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

and stop the process if the above equation does not hold. This strategy, however, does not work to counter our attack model. It is because if $\mathbf{AB}' = \mathbf{B}''\mathbf{A}$, then it holds that $\det \mathbf{B}' = \det \mathbf{B}''$. Therefore, the process always proceeds if the adversary takes $(\mathbf{B}', \mathbf{B}'')$ satisfying $\mathbf{AB}' = \mathbf{B}''\mathbf{A}$, and the adversary can know $\mathbf{AB}' \neq \mathbf{B}''\mathbf{A}$ if the process stops.

4.2 Settings

We use the same notation as in Section 3. We assume that $b > 3$.

We consider the two cases for $\mathcal{M}_b$. The first one is that $\mathcal{M}_b$ is the group of regular circulant matrices over $\mathbb{Z}/2^b\mathbb{Z}$. I.e.,

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

The second one is that $\mathcal{M}_b$ is the group of regular diagonal matrices over $\mathbb{Z}/2^b\mathbb{Z}$. I.e.,

$$\mathcal{M}_b = \left\{ \begin{pmatrix} \alpha & 0 \\ 0 & \beta \end{pmatrix} \;\middle|\; \alpha, \beta \in (\mathbb{Z}/2^b\mathbb{Z})^\times \right\}.$$

Put $\mathbf{A}$ as

$$\mathbf{A} = \begin{pmatrix} \gamma & \delta \\ \delta & \gamma \end{pmatrix} \quad \text{or} \quad \mathbf{A} = \begin{pmatrix} \gamma & 0 \\ 0 & \delta \end{pmatrix}.$$

By using the Weil pairing for (P_b, Q_b) and (P_A, Q_A), we can detect $\det \mathbf{A}$. If $\mathbf{C}$ is a matrix in $\mathcal{M}_b$ with $\det \mathbf{C} = \det \mathbf{A}^{-1}$, then $O(\mathbf{CB}', \mathbf{B}''\mathbf{C}) = 1$ if and only if $(\mathbf{AC})\mathbf{B}' = \mathbf{B}''(\mathbf{AC})$. Therefore, we can assume that $\det \mathbf{A} = 1$ since $\det \mathbf{AC} = 1$.

Let $\gamma_0, \ldots, \gamma_{b-1}, \delta_0, \ldots, \delta_{b-1}$ be 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 [20], detecting $\mathbf{A}$ or $-\mathbf{A}$ reveals the secret key of the FESTA trapdoor function; therefore, it suffices to detect values

$$\gamma_0, \ldots, \gamma_{b-1}, \delta_0, \ldots, \delta_{b-1}$$

to attack the FESTA trapdoor functions. Thus, we assume that the adversary tries to detect these values instead of the secret key.

Remark 2. If we know $\mathbf{A}$, we obtain the secret key by generating a random ciphertext and decrypting it. Therefore, in the FESTA setting, we do not need Robert's attack. The reason why we adopt Robert's attack is that we may need it in more general cases (*e.g.,* IS-CUBE [15]).

5 Strategies

In this section, we provide our adaptive attack for a FESTA trapdoor function.

5.1 Circulant matrices

We first explain the case that we use circulant matrices. I.e.,

$$\mathbf{A} \in \left\{ \begin{pmatrix} \alpha & \beta \\ \beta & \alpha \end{pmatrix} \;\middle|\; \alpha, \beta \in \mathbb{Z}/2^b\mathbb{Z},\ \alpha^2 - \beta^2 = 1 \right\}.$$

From the definition of $\mathbf{A}$, it suffices to find γ and δ for detecting $\mathbf{A}$.

Let $\varepsilon_1, \varepsilon_2$ be elements in $\mathbb{Z}/2^b\mathbb{Z}$ such that $\mathbf{B} + \begin{pmatrix} \varepsilon_1 & 0 \\ \varepsilon_2 & 0 \end{pmatrix}$ is regular for some $\mathbf{B} \in \mathcal{M}_b$ (*i.e.,* at least one of ε_1 and ε_2 is even). Then we have

$$\mathbf{A}\left(\mathbf{B} + \begin{pmatrix} \varepsilon_1 & 0 \\ \varepsilon_2 & 0 \end{pmatrix}\right) = \left(\mathbf{B} + \begin{pmatrix} 0 & 0 \\ \varepsilon_2 & \varepsilon_1 \end{pmatrix}\right)\mathbf{A} \quad \text{if and only if } \varepsilon_1\gamma + \varepsilon_2\delta = 0.$$

Therefore, by our oracle O, we can determine whether $\varepsilon_1\gamma + \varepsilon_2\delta = 0$ for any $\varepsilon_1, \varepsilon_2 \in \mathbb{Z}/2^b\mathbb{Z}$ such that at least one of ε_1 and ε_2 is even. We denote an oracle checking $\varepsilon_1\gamma + \varepsilon_2\delta = 0$ by $O_{\mathrm{coeff}}(\varepsilon_1, \varepsilon_2)$.

At first, the adversary determines (γ_0, δ_0). Note that $\gamma_0 \neq \delta_0$ since $\gamma^2 - \delta^2 = 1$. Since $2^{b-1}\gamma = 2^{b-1}\gamma_0$, the adversary obtains γ_0 by checking $O_{\mathrm{coeff}}(2^{b-1}, 0)$. Without loss of generality, we can assume $\gamma_0 = 1$ because we can swap γ and δ by multiplying $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ to $\mathbf{B}'$ and $\mathbf{B}''$ from left and right, respectively.

We next discuss how to determine (γ_1, δ_1). In fact, we do not need to find the "correct" value of γ_1. Robert's attack also works if the adversary detects $(-\gamma, -\delta)$ instead of (γ, δ). Therefore, we can assume $\gamma_1 = 0$ since $\gamma_0 = 1$. Note that we assume that $b > 3$. We have $(\gamma^{(1)})^2 - (\delta^{(1)})^2 \equiv 1 \pmod{2^3}$ since $\gamma^2 - \delta^2 = 1$. Therefore, it holds that $\delta_1 = 0$.

We denote $\sum_{i=0}^{k} \gamma_i 2^i$ by $\gamma^{(k)}$ and use the same notation for δ. For detecting $\mathbf{A}$, we define the following two procedures:

1. $\mathsf{GetDelta}_k$: Require $\gamma^{(k-2)}$ and $\delta^{(k-1)}$ with $k \in [2, b-1]$, and ensure δ_k,
2. $\mathsf{GetGamma}_k$: Require $\gamma^{(k-1)}$ and $\delta^{(k-1)}$ with $k \in [2, b-2]$, and ensure γ_k.

The details of these procedures are as follows:

GetDelta$_k$: Let γ' be an integer such that $\gamma'\gamma^{(k-2)} \equiv 1 \pmod{2^{k-1}}$. Note that $\delta^{(k-1)} \equiv 0 \pmod{2^2}$. Therefore, it holds that

$$\begin{aligned}
&\delta^{(k-1)}\gamma' 2^{b-k-1}\gamma - 2^{b-k-1}\delta \\
&= 2^{b-k-1}\delta^{(k-1)}\gamma'\gamma^{(k-2)} + 2^{b-2}\delta^{(k-1)}\gamma'(\gamma_{k-1} + 2\gamma_k) - 2^{b-k-1}\delta^{(k-1)} - 2^{b-1}\delta_k \\
&= -2^{b-1}\delta_k.
\end{aligned}$$

Hence, $O_{\mathrm{coeff}}(\delta^{(k-1)}\gamma' 2^{b-k-1}, -2^{b-k-1})$ returns TRUE if and only if $\delta_k = 0$.

GetGamma$_k$: Note that $\delta + \delta^{(k-1)} \equiv 0 \pmod{2^2}$ and $k \leq b-2$. We have

$$\begin{aligned}
\gamma^2 - (\delta^{(k-1)})^2 &= 1 + \delta^2 - (\delta^{(k-1)})^2 \\
&= 1 + (\delta + \delta^{(k-1)}) \sum_{i=k}^{b-1} \delta_i 2^i \equiv 1 \pmod{2^{k+2}}.
\end{aligned}$$

Therefore, the adversary knows $\gamma^2 \bmod 2^{k+2}$ by computing $1 + (\delta^{(k-1)})^2 \bmod 2^{k+2}$. This value is determined by $\gamma^{(k-1)}$ and γ_k. In particular, it holds that

$$\gamma^2 = \left(\gamma^{(k-1)} + 2^k \sum_{i=k}^{b-1} \gamma_i 2^{i-k-1}\right)^2 \equiv (\gamma^{(k-1)})^2 + \gamma_k 2^{k+1} \pmod{2^{k+2}}$$

since $k \geq 2$. Hence, the adversary can find γ_k.

By using the above two procedures, the adversary can detect $\mathbf{A}$. The following is the outline of our strategy for determining $\mathbf{A}$.

1. Find γ_0 from checking $O_{\mathrm{coeff}}(2^{b-1}, 0)$. We can assume $\gamma_0 = 1$ by swapping γ and δ.
2. Set $\gamma_1 = \delta_1 = 0$.
3. For $k = 2, \ldots, b-2$:
   (a) Determine δ_k by GetDelta$_k$.
   (b) Determine γ_k by GetGamma$_k$.
4. Determine δ_{b-1} by GetDelta$_{b-1}$.
5. For $\gamma_{b-1} \in \{0, 1\}$, check the correctness of $\mathbf{A}$ by Robert's attack. If the attack succeeded then output $\mathbf{A}$.

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

Remark 3. As noted in Section 4, we need to assume that the recipient does not compute $\mathbf{B}$ in the decryption process. Indeed, the incorrect input

$$\mathbf{B} + 2^{b-k-1} \begin{pmatrix} \delta^{(k-1)}\gamma' & 0 \\ -1 & 0 \end{pmatrix},$$

which appears in GetDelta$_k$, does not belong to $\mathcal{M}_b$.

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

5.2 Diagonal matrices

We explain the case that we use diagonal matrices. I.e.,

$$\mathbf{A} \in \left\{ \begin{pmatrix} \alpha & 0 \\ 0 & \beta \end{pmatrix} \;\middle|\; \alpha, \beta \in \mathbb{Z}/2^b\mathbb{Z},\ \alpha\beta = 1 \right\}.$$

Let $\varepsilon_1, \varepsilon_2$ be elements in $\mathbb{Z}/2^b\mathbb{Z}$. Then, for any diagonal matrix $\mathbf{B}$, we have

$$\mathbf{A}\left(\mathbf{B} + \begin{pmatrix} 0 & 0 \\ \varepsilon_2 & 0 \end{pmatrix}\right) = \left(\mathbf{B} + \begin{pmatrix} 0 & 0 \\ \varepsilon_1 & 0 \end{pmatrix}\right)\mathbf{A} \quad \text{if and only if } \varepsilon_1\gamma = \varepsilon_2\delta.$$

Therefore, by our oracle O, we can determine whether $\varepsilon_1\gamma^2 - \varepsilon_2 = 0$ for any $\varepsilon_1, \varepsilon_2 \in \mathbb{Z}/2^b\mathbb{Z}$. We denote an oracle checking $\varepsilon_1\gamma^2 - \varepsilon_2 = 0$ by $O_{\mathrm{coeff}}(\varepsilon_1, \varepsilon_2)$.

From the definition of $\mathbf{A}$, it suffices to find γ to recover $\mathbf{A}$. It is clear that $\gamma_0 = 1$. Since Robert's attack also works if the adversary detects $-\gamma$ instead of γ, we can assume $\gamma_1 = 0$ without loss of generality.

We denote $\sum_{i=0}^{k} \gamma_i 2^i$ by $\gamma^{(k)}$. For detecting $\mathbf{A}$, we define the following procedure:

- $\mathsf{GetGamma}_k$: Require $\gamma^{(k-1)}$ with $k \in [2, b-2]$, and ensure γ_k.

$\mathsf{GetGamma}_k$: Since $\gamma + \gamma^{(k-1)} \equiv 2 \pmod{2^2}$, we have

$$2^{b-k-2}\gamma^2 - 2^{b-k-2}(\gamma^{(k-1)})^2 = 2^{b-2}(\gamma_k + \gamma_{k+1}2)(\gamma + \gamma^{(k-1)}) = \gamma_k 2^{b-1}.$$

Therefore, the output of $O_{\mathrm{coeff}}(2^{b-k-2}, 2^{b-k-2}(\gamma^{(k-1)})^2)$ is TRUE if and only if $\gamma_k = 0$. Hence, the adversary can obtain γ_k.

By using the above procedure, the adversary can obtain the target matrix $\mathbf{A}$. The following is the whole outline of our attack.

1. Set $\gamma_0 = 1$ and $\gamma_1 = 0$.
2. For $k = 2, \ldots, b-2$, determine γ_k by $\mathsf{GetGamma}_k$.
3. For $\gamma_{b-1} \in \{0, 1\}$, check the correctness of $\mathbf{A}$ by Robert's attack. If the attack succeeded then output $\mathbf{A}$.

From the above steps, the adversary can know the matrix $\mathbf{A}$ in $b-3$ queries to the oracle O.

6 Conclusion

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

For our attack, we need an oracle that judges whether a correct $(2^b, 2^b)$-isogeny can be calculated in the process of computing the inverse map of the FESTA trapdoor function from a given input. As an example, we showed that this oracle could be constructed under a specific PKE scheme that used a FESTA trapdoor function in the wrong way (*i.e.,* the recipient does not check in the decryption process whether the sender's matrix $\mathbf{B}$ belongs to the fixed group $\mathcal{M}_b$).

The IND-CCA secure PKE scheme named FESTA proposed in [2] is not attacked by our adaptive attack so far.

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 and in part conducted under a contract of "Research and development on new generation cryptography for secure wireless communication services" among "Research and Development for Expansion of Radio Wave Resources (JPJ000254)", which was supported by the Ministry of Internal Affairs and Communications, Japan.

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. In *Advances in Cryptology – ASIACRYPT 2023*, pages 98–126. Springer, 2023.
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. Tomoki Moriya. IS-CUBE: An isogeny-based compact KEM using a boxed SIDH diagram, 2023. https://eprint.iacr.org/2023/1516.
16. David Mumford, Chidambaran Padmanabhan Ramanujam, and Jurij Ivanovič Manin. *Abelian varieties*, volume 5. Oxford university press Oxford, 1974.
17. National Institute of Standards and Technology. Post–quantum cryptography standardization. https://csrc.nist.gov/Projects/post-quantum-cryptography/round-4-submissions.
18. 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.
19. 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.
20. Damien Robert. Breaking SIDH in polynomial time. In *Advances in Cryptology – EUROCRYPT 2023*, pages 472–503. Springer, 2023.
21. 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.
22. Joseph H Silverman. *The arithmetic of elliptic curves*, volume 106. Springer Science & Business Media, 2009.
23. Benjamin Smith. *Explicit endomorphisms and correspondences*. PhD thesis, University of Sydney, 2005.
24. Jacques Vélu. Isogénies entre courbes elliptiques. *CR Acad. Sci. Paris Sér. A*, 273(5):238–241, 1971.