Fault Attacks on Supersingular Isogeny Cryptosystems

Yan Bo Ti

Department of Mathematics,
University of Auckland

PQCrypto 2017, 26th of June
1 Preliminaries
   Introduction
   Supersingular isogenies
   SSI cryptosystems

2 Fault attack
   Fault injection
   Recovering secret isogeny

3 Application
Definition (Discrete Logarithm Problem)

Pick an abelian group \( G = \langle g \rangle \). Given \( g \) and \( X \), where \( X = g^s \), recover \( s \).

- Each scalar \( s \) determines the map \( g \mapsto g^s \).
- Fixing \( s \) is same as fixing endomorphism \( \phi_s : G \to G \).
Definition (Discrete Logarithm Problem)

*Pick an abelian group* $G = \langle g \rangle$. *Given* $g$ and $X$, where $X = g^s$, *recover* $s$.

- Each scalar $s$ determines the map $g \mapsto g^s$.
- Fixing $s$ is same as fixing endomorphism $\phi_s : G \to G$.

Let’s generalise this!
• Fix a finite field \( k = \mathbb{F}_p \) and a finite extension \( K = \mathbb{F}_q \) where \( q = p^k \).

• Let \( E_1 \) and \( E_2 \) be elliptic curves over \( K \).

**Definition**

An isogeny between \( E_1 \) and \( E_2 \) is a non-constant morphism defined over \( \mathbb{F}_q \) that sends \( O_1 \) to \( O_2 \). We say that \( E_1 \) and \( E_2 \) are isogenous.
Fun facts:

- Isogenies are group homomorphisms.
- For every finite subgroup $G \subset E_1$, there is a unique $E_2$ (up to isomorphism) and a separable $\phi : E_1 \to E_2$ such that $\ker \phi = G$. We write $E_2 = E_1/G$.
- The isogeny can be constructed by an algorithm by Vélu.
- For any $\phi : E \to E'$ of degree $n$, there exists a unique $\hat{\phi} : E' \to E$ such that $\phi \circ \hat{\phi} = [n] = \hat{\phi} \circ \phi$.
- For any $\phi : E \to E'$ of degree $nm$, we can decompose $\phi$ into isogenies of degrees $m$ and $n$. 
Supersingular Elliptic Curves

Definition

An elliptic curve $E/\mathbb{F}_{p^k}$ is said to be supersingular if $\#E(\mathbb{F}_{p^k}) \equiv 1 \pmod{p}$.

Fun facts:

- All supersingular elliptic curves can be defined over $\mathbb{F}_{p^2}$.
- There are approximately $p/12$ supersingular curves up to isomorphism.
Definition (Discrete logarithm problem)

Pick an abelian group $G = \langle g \rangle$. Given $g$ and $X$, where $X = g^s$, recover $s$.

- Each scalar $s$ determines the map $g \mapsto g^s$.
- Fixing $s$ is same as fixing endomorphism $\phi_s : G \to G$. 

Supersingular isogeny problem
Supersingular isogeny problem

Definition (Discrete logarithm problem)

Pick an abelian group \( G = \langle g \rangle \). Given \( g \) and \( X \), where \( X = g^s \), recover \( s \).

- Each scalar \( s \) determines the map \( g \mapsto g^s \).
- Fixing \( s \) is same as fixing endomorphism \( \phi_s : G \to G \).

Definition (Supersingular isogeny problem)

Given two supersingular elliptic curves \( E_1 \) and \( E_2 \), find an isogeny between them.
Set up:

- Choose $p = 2^n \cdot 3^m \cdot f \pm 1$, such that $2^n \approx 3^m$ and $f$ small.
- Choose supersingular elliptic curve $E$ over $\mathbb{F}_{p^2}$.
- Alice works over $E[2^n]$ with linearly independent points $P_A, Q_A$.
- Bob works over $E[3^m]$ with linearly independent points $P_B, Q_B$. 

Recall that $E[N] = \mathbb{Z}/N\mathbb{Z} \times \mathbb{Z}/N\mathbb{Z}$ if $N$ is co-prime to the characteristic of the field.
Set up:

- Choose $p = 2^n \cdot 3^m \cdot f \pm 1$, such that $2^n \approx 3^m$ and $f$ small.
- Choose supersingular elliptic curve $E$ over $\mathbb{F}_{p^2}$.
- Alice works over $E[2^n]$ with linearly independent points $P_A, Q_A$.
- Bob works over $E[3^m]$ with linearly independent points $P_B, Q_B$.

Recall that

$$E[N] = \mathbb{Z}/N\mathbb{Z} \times \mathbb{Z}/N\mathbb{Z}$$

if $N$ is co-prime to the characteristic of the field.
- Picks secret $1 \leq a_1, a_2 \leq 2^n$, not both divisible by 2, which determines $G_A = \langle [a_1]P_A + [a_2]Q_A \rangle$.
- Computes $\phi_A$ with $\ker \phi_A = G_A$ via Vélu.
- Sends $E/G_A$, $\phi_A(P_B)$, $\phi_A(Q_B)$. 

\[ E \xrightarrow{\phi_A} E/G_A \]
\[ \downarrow \phi_B \]
\[ E/G_B \]
Key exchange

- Receives $E / G_B$, $\phi_B(P_A)$, $\phi_B(Q_A)$.
- Computes
  
  \[
  G'_A = \langle [a_1]\phi_B(P_A) + [a_2]\phi_B(Q_A) \rangle \\
  = \langle \phi_B([a_1]P_A + [a_2]Q_A) \rangle \\
  = \phi_B(G_A).
  \]

- Uses $j(E_{AB})$ as secret key.
One can try to find mathematical algorithms to break the cryptosystem. Or, one can use side-channel attacks.

Fault attacks are physical attacks aimed at physical devices and may be induced by:

- EM probe
- Clock/volt glitching
- Temperature disturbances
One can try to find mathematical algorithms to break the cryptosystem. Or, one can use side-channel attacks.

Fault attacks are physical attacks aimed at physical devices and may be induced by:

- EM probe
- Clock/volt glitching
- Temperature disturbances
- and more!
Fault attacks

One can try to find mathematical algorithms to break the cryptosystem. Or, one can use side-channel attacks.

Fault attacks are physical attacks aimed at physical devices and may be induced by:

- EM probe
- Clock/volt glitching
- Temperature disturbances
- and more!

Fault attacks cause computation of unintended values which may leak sensitive data.
Given elliptic curve $E$, base point $P$, compute $[\lambda]P$.

- Introduce fault to base point $P \in E$ to become $P' \in E'$.
  - Change in curves occurs because operation does not use $a_6$.
- This changes the elliptic curve from $E$ to $E'$ and potentially makes solving ECDLP easier.
- Solving the ECDLP on $[\lambda]P'$ on $E'$, we learn information about $\lambda$. 
Given elliptic curve $E$, base point $P$, compute $[\lambda]P$.

- Introduce fault to base point $P \in E$ to become $P' \in E'$.
  - Change in curves occurs because operation does not use $a_6$.
- This changes the elliptic curve from $E$ to $E'$ and potentially makes solving ECDLP easier.
- Solving the ECDLP on $[\lambda]P'$ on $E'$, we learn information about $\lambda$.

\[ P \text{ becomes } P' \]

\[ P \xrightarrow{\text{fetch}} \text{Compute } [\lambda](\cdot) \xrightarrow{\text{output}} [\lambda]P' \]
Given elliptic curve $E$, base point $P$, compute $[\lambda]P$.

- Introduce fault to base point $P \in E$ to become $P' \in E'$.
- This changes the elliptic curve from $E$ to $E'$ and potentially makes solving ECDLP easier.
- Solving the ECDLP on $[\lambda]P'$ on $E'$, we learn information about $\lambda$. 
Given a point $P$ and an isogeny $\phi$, compute $\phi(P)$.

- Introduce fault to base point $P \in E$ to become $P' \in E'$.
- This changes the elliptic curve from $E$ to $E'$ and potentially makes solving ECDLP easier.
- Solving the ECDLP on $[\lambda]P'$ on $E'$, we learn information about $\lambda$. 
Given a point $P$ and an isogeny $\phi$, compute $\phi(P)$.

- Introduce fault to base point $P \in E$ to become $P' \in E$.
- This changes the elliptic curve from $E$ to $E'$ and potentially makes solving ECDLP easier.
- Solving the ECDLP on $[\lambda]P'$ on $E'$, we learn information about $\lambda$. 
Given a point $P$ and an isogeny $\phi$, compute $\phi(P)$.

- Introduce fault to base point $P \in E$ to become $P' \in E$.
- Compute $[3^m][f]\phi(P')$ to get $Z$ which will have order $2^n$ with high probability.
- Solving the ECDLP on $[\lambda]P'$ on $E'$, we learn information about $\lambda$. 
Given a point $P$ and an isogeny $\phi$, compute $\phi(P)$.

- Introduce fault to base point $P \in E$ to become $P' \in E$.
- Compute $[3^m][f] \phi(P')$ to get $Z$ which will have order $2^n$ with high probability.
- Use $Z$ to compute $\hat{\phi}$. 

Fault attacks in Isogenies
Given a point $P$ and an isogeny $\phi$, compute $\phi(P)$.

- Introduce fault to base point $P \in E$ to become $P' \in E$.
- Compute $[3^m][f]\phi(P')$ to get $Z$ which will have order $2^n$ with high probability.
- Use $Z$ to compute $\hat{\phi}$.

\[
P \xrightarrow{\text{fetch}} \text{Compute } \phi_A(\cdot) \xrightarrow{\text{output}} \phi_A(P')
\]
Given a point $P$ and an isogeny $\phi$, compute $\phi(P)$.

- Introduce fault to base point $P \in E$ to become $P' \in E$.
- Compute $[3^m][f]\phi(P')$ to get $Z$ which will have order $2^n$ with high probability.
- Use $Z$ to compute $\hat{\phi}$. 

Fault attacks in Isogenies
Faulted point still on curve

- Introduce a fault to the $x$-coordinate of $P$.
- Recover $P'$ by solving for $y$-coordinate. Then $P'$ will lie in $E$ or its quadratic twist $E'$.
- Some implementations do not distinguish between the two.
- If not, there is a 50% chance of $P'$ landing in $E$. 
Given a point $P$ and an isogeny $\phi$, compute $\phi(P)$.

- Introduce fault to base point $P \in E$ to become $P' \in E$.
- Compute $[3^m][f]\phi(P')$ to get $Z$ which will have order $2^n$ with high probability.
- Use $Z$ to compute $\hat{\phi}$. 

Fault attacks in Isogenies
Lemma

Let $E_1$ be a supersingular elliptic curve over $\mathbb{F}_{p^2}$, where $p = 2^n 3^m f \pm 1$. Suppose $\phi : E_1 \to E_2$ is a separable isogeny of degree $2^n$. If $\phi(P') \in E_2$ has order $2^n$, then the kernel of $\hat{\phi}$ will be generated by $\phi(P')$.

N.B. $\phi(P')$ does not have to have order $2^n$. If order is close to $2^n$, we can brute force.
Aim: Recover secret $\phi_A$. 

Key Exchange
Aim: Recover secret $\phi_A$.

- Need to evaluate image of random point under $\phi_A$.
- Fault injection before computation of $\phi_A(P_B)$ or $\phi_A(Q_B)$.
- Alice outputs $\phi_A(P')$, hence attacker may recover $\phi_A$. 
• Image of random points on secret isogeny gives away secret.
  • Recover point of order equal to degree of isogeny.
  • Use point as kernel to construct dual isogeny.
• Important to use countermeasures and checks in implementations!
  • Check point order
  • Able to use point compression in signatures
• Image of random points on secret isogeny gives away secret.
  • Recover point of order equal to degree of isogeny.
  • Use point as kernel to construct dual isogeny.
• Important to use countermeasures and checks in implementations!
  • Check point order
  • Able to use point compression in signatures

THANK YOU!
• Image of random points on secret isogeny gives away secret.
  • Recover point of order equal to degree of isogeny.
  • Use point as kernel to construct dual isogeny.
• Important to use countermeasures and checks in implementations!
  • Check point order
  • Able to use point compression in signatures

THANK YOU!
Also, thanks to NZMS!