PQCrypto 2021
The Twelfth International Conference on Post-Quantum Cryptography

SimS: A Simplification of SiGamal[a]

Isogeny-Based Cryptography

Tako Boris Fouotsa
ROMA TRE UNIVERSITÀ DEGLI STUDI

Christophe Petit
UNIVERSITY OF BIRMINGHAM
ULB UNIVERSITÉ LIBRE DE BRUXELLES

July 2021

[a] See full paper on eprint: `https://eprint.iacr.org/2021/218`

Results

- An IND-CCA attack on a variant of SiGamal suggested by Moriya et al. for IND-CCA security.
- A new IND-CCA secure PKE: SimS.
- SimS is more efficient, and provides more compact keys and ciphertexts compared to SiGamal.

Outline

Elliptic curves and isogenies

CSIDH

Public Key Encryption schemes

SiGamal and C-SiGamal

On the IND-CCA security of SiGamal

SimS

Elliptic curves and isogenies

- Montgomery curves: $E : BY^2 = X^3 + AX^2 + X$.
- E has an abelian group structure.
- **Isogenies**: rational maps between elliptic curves that are morphism respect to the group structure. They are given by Velu formulas.
- Over finite field: E is either **ordinary** ($End(E)$ is an order in a quadratic imaginary field) or **supersingular**, ($End(E)$ is a maximal order in a quaternion algebra).
- **Seperable** isogeny: degree is equal to the size of its kernel. They are easy to compute when their kernel has smooth order.

CSIDH

Classic Diffie-Hellman

Let $G = \langle g \rangle$ be a cyclic group of prime order n.

Classic Diffie-Hellman

Let $G = \langle g \rangle$ be a cyclic group of prime order n.

$\boxed{g}$

Classic Diffie-Hellman

Let $G = \langle g \rangle$ be a cyclic group of prime order n.

$$\boxed{g} \xrightarrow{a} \boxed{A = g^{a}}$$

Classic Diffie-Hellman

Let $G = \langle g \rangle$ be a cyclic group of prime order n.

$$\begin{tikzcd}
g \arrow[r, "a"] \arrow[d, "b"'] & A = g^{a} \\
B = g^{b} &
\end{tikzcd}$$

Classic Diffie-Hellman

Let $G = \langle g \rangle$ be a cyclic group of prime order n.

$$\begin{tikzcd}
g \arrow[r, "a"] \arrow[d, "b"'] & A = g^{a} \arrow[d, "b"] \\
B = g^{b} \arrow[r, "a"'] & \boxed{A^{b} = B^{a} = g^{ab}}
\end{tikzcd}$$

Classic Diffie-Hellman

$$
\begin{tikzcd}
g \arrow[r, "a"] \arrow[d, "b"'] & A = g^{a} \arrow[d, "b"] \\
B = g^{b} \arrow[r, "a"'] & A^{b} = B^{a} = g^{ab}
\end{tikzcd}
$$

Hard problems

DLP: Given g and g^a, compute a.

CDH: Given g, g^a and g^b, compute g^{ab}.

DDH: Given g, g^a, g^b. Find a polynomial time algorithm that succeeds in distinguishing a random group element $h \in G$ from g^{ab} with a probability considerably greater than 1/2.

Classic Diffie-Hellman

Hard problems

DLP: Given g and g^a, compute a.

CDH: Given g, g^a and g^b, compute g^{ab}.

DDH: Given g, g^a, g^b. Find a polynomial time algorithm that succeeds in distinguishing a random group element $h \in G$ from g^{ab} with a probability considerably greater than 1/2.

Bad news: Quantum algorithm by Peter Shor (1994) can compute discrete logs in polynomial time using a large scale quantum computer.

CSIDH (sea-side): Commutative Supersingular Isogeny Diffie-Hellman

Let $\mathcal{S}_p$ be the set of supersingular elliptic curves defined over $\mathbb{F}_p$ where p is a well chosen prime. Let $E \in \mathcal{S}_p$, then

$$\begin{array}{rccc} \pi : & E & \to & E \\ & (x, y) & \mapsto & (x^p, y^p) \end{array}$$

is an endomorphism of E defined over $\mathbb{F}_p$ and $\mathbb{Z}[\pi] \subset End_{\mathbb{F}_p}(E)$. The class group $\mathsf{cl}(\mathbb{Z}[\pi])$ of $\mathbb{Z}[\pi]$ acts freely and transitively on $\mathcal{S}_p$. The action of an ideal class $[\mathfrak{a}]$ of smooth norm N on a curve E translates into an isogeny $\phi_{[\mathfrak{a}]} : E \to [\mathfrak{a}]E$ of smooth degree N.

CSIDH (sea-side): Commutative Supersingular Isogeny Diffie-Hellman

Replace G by $\mathcal{S}_p$ and the exponentiation by the action of the class group $\mathsf{cl}(\mathbb{Z}[\pi])$ on $\mathcal{S}_p$.

CSIDH (sea-side): Commutative Supersingular Isogeny Diffie-Hellman

Replace G by $\mathcal{S}_p$ and the exponentiation by the action of the class group $\mathsf{cl}(\mathbb{Z}[\pi])$ on $\mathcal{S}_p$.

$\boxed{E_0}$

CSIDH (sea-side): Commutative Supersingular Isogeny Diffie-Hellman

Replace G by $\mathcal{S}_p$ and the exponentiation by the action of the class group $\mathsf{cl}(\mathbb{Z}[\pi])$ on $\mathcal{S}_p$.

$$\boxed{E_0} \xrightarrow{[\mathfrak{a}]} \boxed{E_1 = [\mathfrak{a}]E_0}$$

CSIDH (sea-side): Commutative Supersingular Isogeny Diffie-Hellman

Replace G by $\mathcal{S}_p$ and the exponentiation by the action of the class group $\mathsf{cl}(\mathbb{Z}[\pi])$ on $\mathcal{S}_p$.

$$\begin{tikzcd}
E_0 \arrow[r, "{[\mathfrak{a}]}"] \arrow[d, "{[\mathfrak{b}]}"'] & E_1 = [\mathfrak{a}]E_0 \\
E_3 = [\mathfrak{b}]E_0 &
\end{tikzcd}$$

CSIDH (sea-side): Commutative Supersingular Isogeny Diffie-Hellman

Replace G by $\mathcal{S}_p$ and the exponentiation by the action of the class group $\mathsf{cl}(\mathbb{Z}[\pi])$ on $\mathcal{S}_p$.

$$\begin{tikzcd}
E_0 \arrow[r, "{[\mathfrak{a}]}"] \arrow[d, "{[\mathfrak{b}]}"'] & E_1 = [\mathfrak{a}]E_0 \arrow[d, "{[\mathfrak{b}]}"] \\
E_3 = [\mathfrak{b}]E_0 \arrow[r, "{[\mathfrak{a}]}"'] & {[\mathfrak{a}][\mathfrak{b}]E_0 = [\mathfrak{b}][\mathfrak{a}]E_0}
\end{tikzcd}$$

CSIDH (sea-side): Commutative Supersingular Isogeny Diffie-Hellman

$$
\begin{tikzcd}
E_0 \arrow[r, "{[\mathfrak{a}]}"] \arrow[d, "{[\mathfrak{b}]}"'] & E_1 = [\mathfrak{a}]E_0 \arrow[d, "{[\mathfrak{b}]}"] \\
E_3 = [\mathfrak{b}]E_0 \arrow[r, "{[\mathfrak{a}]}"'] & {[\mathfrak{a}][\mathfrak{b}]E_0 = [\mathfrak{b}][\mathfrak{a}]E_0}
\end{tikzcd}
$$

CSSIP: Given E_0 and $[\mathfrak{a}]E_0$, compute $[\mathfrak{a}]$

CSSICDH: Given E_0, $[\mathfrak{a}]E_0$ and $[\mathfrak{b}]E_0$, compute $[\mathfrak{b}][\mathfrak{a}]E_0$

CSSIDDH: Given E_0, $[\mathfrak{a}]E_0$, $[\mathfrak{b}]E_0$. Find a polynomial time algorithm that succeeds in distinguishing a random curve E from $[\mathfrak{b}][\mathfrak{a}]E_0$ with a probability considerably greater than 1/2.

Public Key Encryption schemes

Public Key Encryption schemes (PKE)

- Used for message **confidentiality.**
- Made-up of three PPT algorithms :
  - $\mathsf{KeyGeneration}$: which generates a pair of keys $(\mathsf{sk}, \mathsf{pk})$ for a user Alice.
  - $\mathsf{Encryption}$: which computes a ciphertext c when given a public key pk and a plaintext message m.
  - $\mathsf{Decryption}$: which recovers a plaintext m when given the secret key sk and a ciphertext c of m.
- Needs to fulfil:
  - **Correctness**: $\mathsf{Decryption}(\mathsf{Encryption}(\mathsf{m})) = \mathsf{m}$.
  - **OW-CPA secure**: no PPT adversary should be able to recover m from c and pk without the knowledge of sk.

- Needs to fulfil:
  - **Correctness**: $\mathsf{Decryption}(\mathsf{Encryption}(\mathsf{m})) = \mathsf{m}$.
  - **OW-CPA secure**: no PPT adversary should be able to recover m from c and pk without the knowledge of sk.

- Higher security requirements:
  - **IND-CPA secure**: no PPT adversary who chooses to plaintexts m_0 and m_1 should be able to distinguish if a ciphertext of a random m_b is that of m_0 or m_1.
  - **IND-CCA secure**: no PPT adversary having access to a decryption oracle who chooses to plaintexts m_0 and m_1 should be able to distinguish if a ciphertext c of a random m_b ($b = 0$ or $b = 1$) is that of m_0 or m_1.

A PKE from CSIDH

- $\mathsf{KeyGeneration}$: A starting curve E_0 is given. Choose a secret key $\mathsf{sk} = [\mathfrak{a}]$ and compute the public key $\mathsf{pk} = [\mathfrak{a}]E_0$.
- $\mathsf{Encryption}$: Given a plaintext m, choose a random ideal $[\mathfrak{b}]$, the ciphertext is $\mathsf{c} = ([\mathfrak{b}]E_0, \mathsf{c}_1)$ where $\mathsf{c}_1 = A_{[\mathfrak{a}][\mathfrak{b}]E_0} \oplus \mathsf{m}$.
- $\mathsf{Decryption}$: Given a ciphertext $\mathsf{c} = (E_3, \mathsf{c}_1)$ and the secret key $[\mathfrak{a}]$, compute $[\mathfrak{a}]E_3$ and recover $\mathsf{m} = A_{[\mathfrak{a}]E_3} \oplus \mathsf{c}_1$.

- OW-CPA secure? **Yes**.
- IND-CPA secure? No.
  Why? Supersingular curves are distinguishable from random strings. If a ciphertext c is that of m_0, then $\mathsf{c}_1 \oplus \mathsf{m}_1$ is unlikely to be a supersingular curve.
- Any repair? **Yes** : use hash functions and set $\mathsf{c}_1 = H(A_{[\mathfrak{a}][\mathfrak{b}]E_0}) \oplus \mathsf{m}$. (Or other generic transforms...).
- Any repair without using hash functions? **Yes: SiGamal and C-SiGamal**

SiGamal and C-SiGamal

SiGamal PKE

$\boxed{E_0,\ P_0}$

SiGamal PKE

$$\begin{tikzcd} \boxed{E_0,\ P_0} \arrow[r, "\mathfrak{a}"] & \boxed{\begin{array}{c} E_1 = [\mathfrak{a}]E_0, \\ P_1 = \mathfrak{a}P_0 \end{array}} \end{tikzcd}$$

$\mathsf{sk} = \mathfrak{a}, \quad \mathsf{pk} = (E_1, P_1),$

SiGamal PKE

$$\begin{tikzcd}
E_0,\ P_0 \arrow[r, "\mathfrak{a}"] \arrow[d, "\mathfrak{b}"] & \begin{array}{c} E_1 = [\mathfrak{a}]E_0, \\ P_1 = \mathfrak{a}P_0 \end{array} \arrow[d, "\mathfrak{b}"] \\
\begin{array}{c} E_3 = [\mathfrak{b}]E_0, \\ P_3 = \mathfrak{b}P_0 \end{array} & \begin{array}{c} E_4 = [\mathfrak{b}]E_1, \\ P_4 = [2m+1]\mathfrak{b}P_1 \end{array}
\end{tikzcd}$$

$$\mathsf{sk} = \mathfrak{a}, \quad \mathsf{pk} = (E_1, P_1), \quad \mathsf{c} = (E_3, P_3, E_4, P_4).$$

SiGamal PKE

$$\begin{tikzcd}
E_0,\ P_0 \arrow[r, "\mathfrak{a}"] \arrow[dd, "\mathfrak{b}"'] & \begin{array}{c} E_1 = [\mathfrak{a}]E_0, \\ P_1 = \mathfrak{a}P_0 \end{array} \arrow[d, "\mathfrak{b}"] \\
 & \begin{array}{c} E_4 = [\mathfrak{b}]E_1, \\ P_4 = [2m+1]\mathfrak{b}P_1 \end{array} \\
\begin{array}{c} E_3 = [\mathfrak{b}]E_0, \\ P_3 = \mathfrak{b}P_0 \end{array} \arrow[r, "\mathfrak{a}"'] & \begin{array}{c} E_4 = [\mathfrak{a}]E_3, \\ m' = DLP(P_4, \mathfrak{a}P_3) \end{array}
\end{tikzcd}$$

$\mathsf{sk} = \mathfrak{a}, \quad \mathsf{pk} = (E_1, P_1), \qquad \mathsf{c} = (E_3, P_3, E_4, P_4).$

Too large ciphertexts?

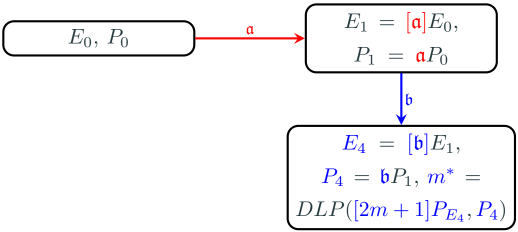
C-SiGamal PKE

For any given curve E, P_E is a canonical point of E of order 2^r.

$$\boxed{E_0,\ P_0} \xrightarrow{\mathfrak{a}} \boxed{\begin{array}{c} E_1 = [\mathfrak{a}]E_0, \\ P_1 = \mathfrak{a}P_0 \end{array}}$$

C-SiGamal PKE

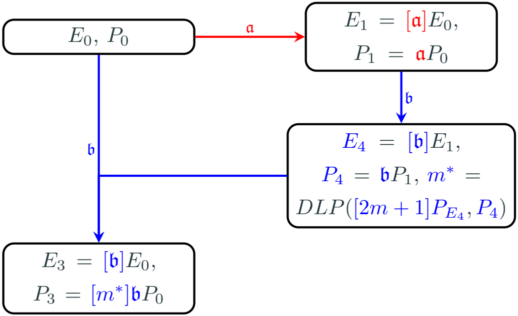
For any given curve E, P_E is a canonical point of E of order 2^r.



$\mathsf{sk} = \mathfrak{a}, \quad \mathsf{pk} = (E_1, P_1),$

C-SiGamal PKE

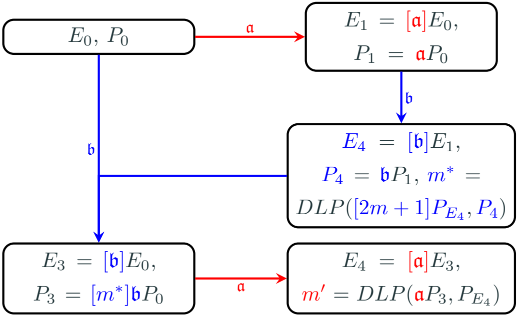
For any given curve E, P_E is a canonical point of E of order 2^r.



sk $= \mathfrak{a}$, pk $= (E_1, P_1)$, c $= (E_3, P_3)$.

C-SiGamal PKE

For any given curve E, P_E is a canonical point of E of order 2^r.



$\mathsf{sk} = \mathfrak{a}, \quad \mathsf{pk} = (E_1, P_1), \quad \mathsf{c} = (E_3, P_3).$

OW-CPA security : P-CSSICDH assumption

Given E_0, P_0, $[\mathfrak{a}]E_0$, $\mathfrak{a}P_0$, $[\mathfrak{b}]E_0$, $\mathfrak{b}P_0$ and $[\mathfrak{a}][\mathfrak{b}]E_0$, no PPT adversary can return $[\mathfrak{a}][\mathfrak{b}]P_0$ with non negligible probability.

IND-CPA security: P-CSSIDDH assumption

Given E_0, P_0, $[\mathfrak{a}]E_0$, $\mathfrak{a}P_0$, $[\mathfrak{b}]E_0$, $\mathfrak{b}P_0$ and $[\mathfrak{a}][\mathfrak{b}]E_0$, no PPT adversary succeeds in distinguishing a random point $P \in [\mathfrak{a}][\mathfrak{b}]E_0(\mathbb{F}_p)[2^r]$ from $[\mathfrak{b}][\mathfrak{a}]P_0$ with a probability non negligibly greater than $1/2$.

On the IND-CCA security of SiGamal

On the IND-CCA security of SiGamal and C-SiGamal

SiGamal and C-SiGamal are not IND-CCA secure

Given a ciphertext $([\mathfrak{b}]E_0, \mathfrak{b}P_0, [\mathfrak{b}][\mathfrak{a}]E_0, [2\mathsf{m}+1]\mathfrak{b}\mathfrak{a}P_0)$ for m, $([\mathfrak{b}]E_0, \mathfrak{b}P_0, [\mathfrak{b}][\mathfrak{a}]E_0, [3][2\mu+1]\mathfrak{b}\mathfrak{a}P_0)$ is a ciphertext for $3\mathsf{m}+1$ since $3(2\mathsf{m}+1) = 2(3\mathsf{m}+1)+1$. A similar reason applies for C-SiGamal

A variant that could be IND-CCA secure?

Moriya et al. suggested removing the curve $[\mathfrak{b}][\mathfrak{a}]E_0$ from the ciphertext. Hence the ciphertext would become $([\mathfrak{b}]E_0, \mathfrak{b}P_0, [2\mathsf{m}+1]\mathfrak{b}\mathfrak{a}P_0)$

The variant is not IND-CCA secure

A simple IND-CCA attack

We prove that given a ciphertext $([\mathfrak{b}]E_0, \mathfrak{b}P_0, [2\mathsf{m}+1]\mathfrak{b}\mathfrak{a}P_0)$ for m, $([\mathfrak{b}]E_0, [3^{-1}]\mathfrak{b}P_0, [2\mathsf{m}+1]\mathfrak{b}\mathfrak{a}P_0)$ is a ciphertext for $3\mathsf{m}+1$

Why is this attack successful?

Because the ciphertext contains a curve and one of its points.

Can we avoid it?

May be by making sure that when a curve is part of the ciphertext, then none of its points is, and the other way around.

SimS

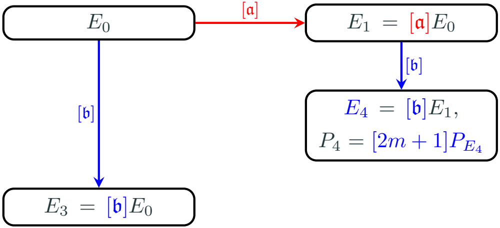
Replace $\mathfrak{a}\mathfrak{b}P_0$ in SiGamal by the canonical point $P_{E_4} \in E_4 = [\mathfrak{a}][\mathfrak{b}]E_0$. A ciphertext for m is $([\mathfrak{b}]E_0, P_4 = [2\mathsf{m} + 1]P_{[\mathfrak{a}][\mathfrak{b}]E_0})$. In order to recover m, Alice computes $[\mathfrak{a}][\mathfrak{b}]E_0$ and $P_{[\mathfrak{a}][\mathfrak{b}]E_0}$, solves a discrete logarithm instance between P_4 and $P_{[\mathfrak{a}][\mathfrak{b}]E_0}$.

An overview of SimS

Replace $\mathfrak{a}\mathfrak{b}P_0$ in SiGamal by the canonical point $P_{E_4} \in E_4 = [\mathfrak{a}][\mathfrak{b}]E_0$. A ciphertext for m is $([\mathfrak{b}]E_0, P_4 = [2\mathsf{m} + 1]P_{[\mathfrak{a}][\mathfrak{b}]E_0})$. In order to recover m, Alice computes $[\mathfrak{a}][\mathfrak{b}]E_0$ and $P_{[\mathfrak{a}][\mathfrak{b}]E_0}$, solves a discrete logarithm instance between P_4 and $P_{[\mathfrak{a}][\mathfrak{b}]E_0}$.

$$\begin{tikzcd} E_0 \arrow[r, "{[\mathfrak{a}]}"] & E_1 = [\mathfrak{a}]E_0 \end{tikzcd}$$

Secret Key: $\mathsf{sk} = [\mathfrak{a}]$, Public Key: $\mathsf{pk} = E_1$, Replace $\mathfrak{a}\mathfrak{b}P_0$ in SiGamal by the canonical point $P_{E_4} \in E_4 = [\mathfrak{a}][\mathfrak{b}]E_0$. A ciphertext for m is $([\mathfrak{b}]E_0, P_4 = [2\mathsf{m}+1]P_{[\mathfrak{a}][\mathfrak{b}]E_0})$. In order to recover m, Alice computes $[\mathfrak{a}][\mathfrak{b}]E_0$ and $P_{[\mathfrak{a}][\mathfrak{b}]E_0}$, solves a discrete logarithm instance between P_4 and $P_{[\mathfrak{a}][\mathfrak{b}]E_0}$.



Secret Key: $\mathsf{sk} = [\mathfrak{a}]$, Public Key: $\mathsf{pk} = E_1$, Ciphertext: $\mathsf{c} = (E_3, P_4)$.

An overview of SimS

Replace $\mathfrak{a}\mathfrak{b}P_0$ in SiGamal by the canonical point $P_{E_4} \in E_4 = [\mathfrak{a}][\mathfrak{b}]E_0$. A ciphertext for m is $([\mathfrak{b}]E_0, P_4 = [2\mathsf{m}+1]P_{[\mathfrak{a}][\mathfrak{b}]E_0})$. In order to recover m, Alice computes $[\mathfrak{a}][\mathfrak{b}]E_0$ and $P_{[\mathfrak{a}][\mathfrak{b}]E_0}$, solves a discrete logarithm instance between P_4 and $P_{[\mathfrak{a}][\mathfrak{b}]E_0}$.

$$\begin{tikzcd}
E_0 \arrow[r, "{[\mathfrak{a}]}"] \arrow[dd, "{[\mathfrak{b}]}"'] & E_1 = [\mathfrak{a}]E_0 \arrow[d, "{[\mathfrak{b}]}"] \\
 & \begin{array}{c} E_4 = [\mathfrak{b}]E_1, \\ P_4 = [2m+1]P_{E_4} \end{array} \\
E_3 = [\mathfrak{b}]E_0 \arrow[r, "{[\mathfrak{a}]}"'] & \begin{array}{c} E_4 = [\mathfrak{a}]E_3,\ m' = \\ DLP(P_4, P_{E_4}) \end{array}
\end{tikzcd}$$

Secret Key: $\mathsf{sk} = [\mathfrak{a}]$, Public Key: $\mathsf{pk} = E_1$, Ciphertext: $\mathsf{c} = (E_3, P_4)$.

Could $[2\mathsf{m}+1]P_{[\mathfrak{a}][\mathfrak{b}]E_0}$ or its x-coordinate leak too much about $[\mathfrak{a}][\mathfrak{b}]E_0$?

We make use of a randomizing function $f_E : \mathbb{F}_p \to \mathbb{F}_p$ satisfying the following conditions:

- f_E is bijective, f_E and $g_E = f_E^{-1}$ can be efficiently computed when E is given;
- an adversary can not distinguish $f_E(x)$ from a random element of $\mathbb{F}_p$;
- an adversary can not compute $f_E(R(x))$ from $f_E(x)$ where $R(x)$ is a non identical rational function.

Could $[2\mathsf{m}+1]P_{[\mathfrak{a}][\mathfrak{b}]E_0}$ or its x-coordinate leak too much about $[\mathfrak{a}][\mathfrak{b}]E_0$?

We make use of a randomizing function $f_E : \mathbb{F}_p \to \mathbb{F}_p$ satisfying the following conditions:

- f_E is bijective, f_E and $g_E = f_E^{-1}$ can be efficiently computed when E is given;
- an adversary can not distinguish $f_E(x)$ from a random element of $\mathbb{F}_p$;
- an adversary can not compute $f_E(R(x))$ from $f_E(x)$ where $R(x)$ is a non identical rational function.

Example: $f_E : x \mapsto x \oplus A_E$.

$$
\begin{tikzcd}
E_0 \arrow[r, "{[\mathfrak{a}]}"] \arrow[dd, "{[\mathfrak{b}]}"'] & E_1 = [\mathfrak{a}]E_0 \arrow[d, "{[\mathfrak{b}]}"] \\
 & \begin{array}{c} E_4 = [\mathfrak{b}]E_1, \\ P_4 = [2m+1]P_{E_4}, \\ x' = f_{E_4}(x_{P_4}) \end{array} \\
E_3 = [\mathfrak{b}]E_0 \arrow[r, "{[\mathfrak{a}]}"'] & \begin{array}{c} E_4 = [\mathfrak{a}]E_3, \\ x_{P_4} = g_{E_4}(x'), \\ m' = DLP(P_4, P_{E_4}) \end{array}
\end{tikzcd}
$$

Secret Key: $\mathsf{sk} = [\mathfrak{a}]$, Public Key: $\mathsf{pk} = E_1$, Ciphertext: $\mathsf{c} = (E_3, x')$.

IND-CPA security

Theorem: If CSSIDDH holds, then SimS is IND-CPA secure.

IND-CCA security

A knowledge of Exponent assumption: For every PPT adversary $\mathcal{A}$ which when given a ciphertext (E_3, x') outputs a valid ciphertext $(F, y') \neq (E_3, x')$, there exists PPT adversary $\mathcal{A}'$ which when given a ciphertext (E_3, x') outputs $([\mathfrak{b}'], F, y')$ where $(F, y') \neq (E_3, x')$ is a valid ciphertext and $[\mathfrak{b}']E_0 = F$.

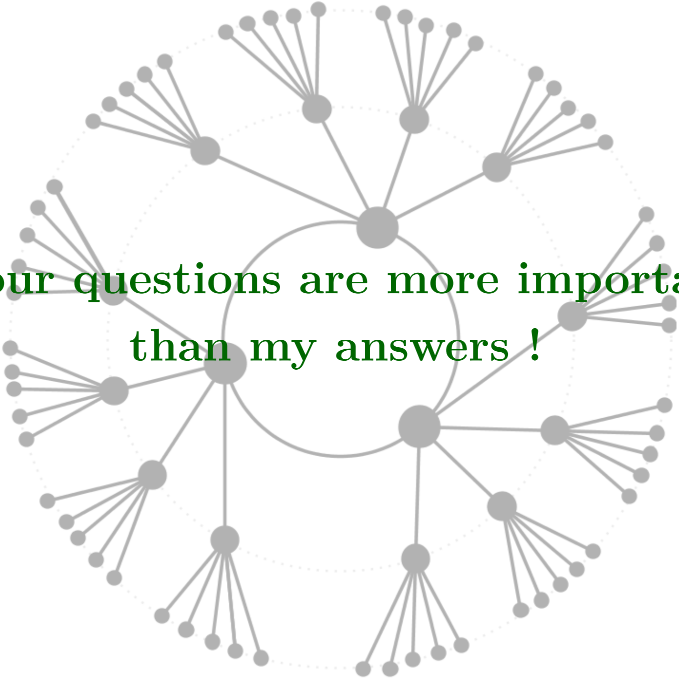
Theorem: If the previous assumption holds and SimS is IND-CPA secure, then SimS is IND-CCA secure.

Summary

$p_{128} = 2^{130} \cdot \ell_1 \cdots \ell_{60} - 1$ where ℓ_1 through ℓ_{59} are the smallest distinct odd primes, and ℓ_{60} is 569.

$p_{256} = 2^{258} \cdot \ell_1 \cdots \ell_{43} - 1$ where ℓ_1 through ℓ_{42} are the smallest distinct odd primes, and ℓ_{43} is 307.

	CSIDHpke	SimS	SiGamal	C-SiGamal
Private key	$[\mathfrak{a}]$	$[\mathfrak{a}]$	$\mathfrak{a}$	$\mathfrak{a}$
Size of plaintext	$\log_2 p$	$r-2$	$r-2$	$r-2$
Size of public key	$\log_2 p$	$\log_2 p$	$2\log_2 p$	$2\log_2 p$
Size of ciphertexts	$2\log_2 p$	$2\log_2 p$	$4\log_2 p$	$2\log_2 p$
Class group cost for p_{128}	x1.00	x1.30	x1.50	x1.50
Class group cost for p_{256}	x1.00	x2.31	x2.57	x2.57
Enc + Dec cost for p_{128}	x1.00	x1.38	x1.57	x1.65
Enc + Dec cost for p_{256}	x1.00	x2.62	x2.82	x3.17
Security	OW-CPA	IND-CCA	IND-CPA	IND-CPA



Your questions are more important than my answers !