SPDH-Sign

Towards Efficient, Post-quantum, Group-based Signatures

Christopher Battarbee [1] Delaram Kahrobaei [2]
Ludovic Perret [3] Siamak F. Shahandashti [1]

[1] University of York

[2] City University of New York

[3] Sorbonne University

August 7, 2023

What is SPDH-Sign?

- Semidirect Product Diffie-Hellman Signatures

[1] Jean-Marc Couveignes. “Hard homogeneous spaces”. In: *Cryptology ePrint Archive* (2006), Alexander Rostovtsev and Anton Stolbunov. “Public-key cryptosystem based on isogenies”. In: *Cryptology ePrint Archive* (2006).

What is SPDH-Sign?

- Semidirect Product Diffie-Hellman Signatures
- A Couveignes-Rostostev-Stolbunov[1] style signature scheme based on group actions arising from group-based cryptography (CRS schemes)

[1] Jean-Marc Couveignes. "Hard homogeneous spaces". In: *Cryptology ePrint Archive* (2006), Alexander Rostovtsev and Anton Stolbunov. "Public-key cryptosystem based on isogenies". In: *Cryptology ePrint Archive* (2006).

What is SPDH-Sign?

- Semidirect Product Diffie-Hellman Signatures
- A Couveignes-Rostostev-Stolbunov[1] style signature scheme based on group actions arising from group-based cryptography (CRS schemes)
- Addresses a problem with efficient sampling found in similar schemes

[1] Jean-Marc Couveignes. “Hard homogeneous spaces”. In: *Cryptology ePrint Archive* (2006), Alexander Rostovtsev and Anton Stolbunov. “Public-key cryptosystem based on isogenies”. In: *Cryptology ePrint Archive* (2006).

Group Actions

Definition

A group action $(G, X, \circledast)$ consists of a finite abelian group G, a set X, and a function $\circledast : G \times X \to X$ such that for all $g, h \in G, x \in X$

- $(g + h) \circledast x = g \circledast (h \circledast x)$
- $0 \circledast x = x$

Group Actions

Definition

A group action $(G, X, \circledast)$ consists of a finite abelian group G, a set X, and a function $\circledast : G \times X \to X$ such that for all $g, h \in G, x \in X$

- $(g + h) \circledast x = g \circledast (h \circledast x)$
- $0 \circledast x = x$

We are interested in free, transitive group actions: *only* 0 fixes every element of x, every pair (x, y) has a (unique) element $g \in G$ such that $g \circledast x = y$.

A Sigma Protocol

$$X_0, X_1 \in X,\ s \circledast X_0 = X_1$$

P $\qquad\qquad$ V

$r \xleftarrow{\$} G$

$I \leftarrow r \circledast X_0$

$$\xrightarrow{\quad I \quad}$$

$c \xleftarrow{\$} \{0, 1\}$

$$\xleftarrow{\quad c \quad}$$

$p \leftarrow r - c \cdot s$

$$\xrightarrow{\quad p \quad}$$

Transcripts (I, c, p) are passing if

$$p \circledast X_c \stackrel{?}{=} I$$

A Sigma Protocol

$$X_0, X_1 \in X,\ s \circledast X_0 = X_1$$

$P \qquad\qquad\qquad\qquad V$

$r \xleftarrow{\$} G$

$I \leftarrow r \circledast X_0$

$$\xrightarrow{\quad I \quad}$$

$c \xleftarrow{\$} \{0, 1\}$

$$\xleftarrow{\quad c \quad}$$

$p \leftarrow r - c \cdot s$

$$\xrightarrow{\quad p \quad}$$

Transcripts (I, c, p) are passing if

$p \circledast X_c \stackrel{?}{=} I$

$$\begin{tikzcd}
X_0 \arrow[r, "s\circledast"] \arrow[d, "r\circledast"'] & X_1 \arrow[dl, dashed, "(r-s)\circledast"] \\
I &
\end{tikzcd}$$

Fiat-Shamir Transform

- Uses hash functions into the challenge space to generate sigma protocol transcripts non-interactively

[2] Ward Beullens et al. "CSI-FiSh: efficient isogeny based signatures through class group computations". In: *ASIACRYPT*. 2019.

[3] Jelle Don et al. "Security of the Fiat-Shamir transformation in the quantum random-oracle model". In: *CRYPTO2019*.

Fiat-Shamir Transform

- Uses hash functions into the challenge space to generate sigma protocol transcripts non-interactively
- Gives ‘secure’ signature schemes provided hash functions are modelled as random oracles

[2] Ward Beullens et al. “CSI-FiSh: efficient isogeny based signatures through class group computations”. In: *ASIACRYPT*. 2019.

[3] Jelle Don et al. “Security of the Fiat-Shamir transformation in the quantum random-oracle model”. In: *CRYPTO2019*.

Fiat-Shamir Transform

- Uses hash functions into the challenge space to generate sigma protocol transcripts non-interactively
- Gives 'secure' signature schemes provided hash functions are modelled as random oracles
- As shown (roughly) in[2] results of[3] (QROM) go through given special soundness and honest verifier zero knowledge.

[2] Ward Beullens et al. "CSI-FiSh: efficient isogeny based signatures through class group computations". In: *ASIACRYPT*. 2019.

[3] Jelle Don et al. "Security of the Fiat-Shamir transformation in the quantum random-oracle model". In: *CRYPTO2019*.

Fiat-Shamir Transform

- Uses hash functions into the challenge space to generate sigma protocol transcripts non-interactively
- Gives ‘secure’ signature schemes provided hash functions are modelled as random oracles
- As shown (roughly) in[2] results of[3] (QROM) go through given special soundness and honest verifier zero knowledge.
- Provided large enough challenge space, security bounded by advantage against recovering s from X_0, X_1; so-called group action discrete logarithm problem admitting quantum subexponential algorithms

---

[2] Ward Beullens et al. “CSI-FiSh: efficient isogeny based signatures through class group computations”. In: *ASIACRYPT*. 2019.

[3] Jelle Don et al. “Security of the Fiat-Shamir transformation in the quantum random-oracle model”. In: *CRYPTO2019*.

Honest Verifier Zero Knowledge

Definition

Our sigma protocol has HVZK if given c we can efficiently produce a transcript $(\bar{l}, c, \bar{p})$ such that

- $\bar{p} \circledast X_c \stackrel{?}{=} \bar{p}$
- $(\bar{l}, c, \bar{p})$ has the same distribution as an honestly generated transcript

Honest Verifier Zero Knowledge

Definition

Our sigma protocol has HVZK if given c we can efficiently produce a transcript $(\bar{l}, c, \bar{p})$ such that

- $\bar{p} \circledast X_c \stackrel{?}{=} \bar{p}$
- $(\bar{l}, c, \bar{p})$ has the same distribution as an honestly generated transcript

---

- Given c choose $r \xleftarrow{\$} G$ and output $(r \circledast X_c, c, r)$.

Honest Verifier Zero Knowledge

Definition

Our sigma protocol has HVZK if given c we can efficiently produce a transcript $(\bar{I}, c, \bar{p})$ such that

- $\bar{p} \circledast X_c \stackrel{?}{=} \bar{p}$
- $(\bar{I}, c, \bar{p})$ has the same distribution as an honestly generated transcript

---

- Given c choose $r \xleftarrow{\$} G$ and output $(r \circledast X_c, c, r)$.
- Recall honest transcripts look like

$$(r \circledast X_0, 0, r) \text{ or } (r \circledast X_0, 1, r - s)$$

- In other words we have HVZK provided we can sample uniformly at random

Sampling

- Ability to sample uniformly gives zero knowledge property

[4] Ward Beullens et al. "CSI-FiSh: efficient isogeny based signatures through class group computations". In: *ASIACRYPT*. 2019.

[5] Luca De Feo and Steven D Galbraith. "SeaSign: compact isogeny signatures from class group actions". In: *EUROCRYPT*. 2019.

Sampling

- Ability to sample uniformly gives zero knowledge property
- Standard / original group action comes from isogenies; group hard to compute

[4] Ward Beullens et al. "CSI-FiSh: efficient isogeny based signatures through class group computations". In: *ASIACRYPT*. 2019.

[5] Luca De Feo and Steven D Galbraith. "SeaSign: compact isogeny signatures from class group actions". In: *EUROCRYPT*. 2019.

Sampling

- Ability to sample uniformly gives zero knowledge property
- Standard / original group action comes from isogenies; group hard to compute
- Workarounds include a one-time expensive calculation[4] and 'Fiat-Shamir with aborts'[5]

[4] Ward Beullens et al. "CSI-FiSh: efficient isogeny based signatures through class group computations". In: *ASIACRYPT*. 2019.

[5] Luca De Feo and Steven D Galbraith. "SeaSign: compact isogeny signatures from class group actions". In: *EUROCRYPT*. 2019.

Recap

- Certain types of group actions give short post-quantum signatures

Recap

- Certain types of group actions give short post-quantum signatures
- Important to be able to compute the group of the group action or the security proofs fall apart

Recap

- Certain types of group actions give short post-quantum signatures
- Important to be able to compute the group of the group action or the security proofs fall apart
- Mainstream example of cryptographic group actions are such that it is difficult to compute the group

Recap

- Certain types of group actions give short post-quantum signatures
- Important to be able to compute the group of the group action or the security proofs fall apart
- Mainstream example of cryptographic group actions are such that it is difficult to compute the group
- Current solutions are slow to compute signatures or to generate parameters

Recap

- Certain types of group actions give short post-quantum signatures
- Important to be able to compute the group of the group action or the security proofs fall apart
- Mainstream example of cryptographic group actions are such that it is difficult to compute the group
- Current solutions are slow to compute signatures or to generate parameters
- Would be nice to have a group action where we could compute the group easily...

The Semidirect Product

Definition

Let G be a finite group and $Aut(G)$ be its group of automorphisms. The semidirect product of G by $Aut(G)$ (written $G \rtimes Aut(G)$) is the group $G \times Aut(G)$ equipped with multiplication

$$(g, \phi)(h, \psi) = (\psi(g)h, \psi\phi)$$

The Semidirect Product

Definition

Let G be a finite group and $Aut(G)$ be its group of automorphisms. The semidirect product of G by $Aut(G)$ (written $G \rtimes Aut(G)$) is the group $G \times Aut(G)$ equipped with multiplication

$$(g, \phi)(h, \psi) = (\psi(g)h, \psi\phi)$$

Definition

Let G be a finite group. Each pair $(g, \phi) \in G \rtimes Aut(G)$ defines a function $s_{g,\phi} : \mathbb{Z} \to G$ by

$$(g, \phi)^x = (s_{g,\phi}(x), \phi^x)$$

Acting by Integers

Notice[6]

$$
\begin{aligned}
(s_{g,\phi}(x+y), \phi^{x+y}) &= (g,\phi)^{x+y} \\
&= (g,\phi)^{x}(g,\phi)^{y} \\
&= (s_{g,\phi}(x), \phi^{x})(s_{g,\phi}(y), \phi^{y}) \\
&= (\phi^{y}\left(s_{g,\phi}(x)\right) s_{g,\phi}(y), \phi^{x+y})
\end{aligned}
$$

[6] Maggie Habeeb et al. "Public key exchange using semidirect product of (semi) groups". In: *ACNS*. 2013.

Acting by Integers

Notice[6]

$$\begin{aligned}(s_{g,\phi}(x+y),\phi^{x+y}) &= (g,\phi)^{x+y}\\ &= (g,\phi)^{x}(g,\phi)^{y}\\ &= (s_{g,\phi}(x),\phi^{x})(s_{g,\phi}(y),\phi^{y})\\ &= (\phi^{y}\left(s_{g,\phi}(x)\right)s_{g,\phi}(y),\phi^{x+y})\end{aligned}$$

There is a function $*:\mathbb{Z}\times G\to G$ such that

[6] Maggie Habeeb et al. "Public key exchange using semidirect product of (semi) groups". In: *ACNS*. 2013.

Acting by Integers

Notice[6]

$$\begin{aligned}(s_{g,\phi}(x+y), \phi^{x+y}) &= (g,\phi)^{x+y} \\ &= (g,\phi)^x (g,\phi)^y \\ &= (s_{g,\phi}(x), \phi^x)(s_{g,\phi}(y), \phi^y) \\ &= (\phi^y\left(s_{g,\phi}(x)\right) s_{g,\phi}(y), \phi^{x+y})\end{aligned}$$

There is a function $* : \mathbb{Z} \times G \to G$ such that

$$y * s_{g,\phi}(x) = s_{g,\phi}(x+y)$$

[6] Maggie Habeeb et al. "Public key exchange using semidirect product of (semi) groups". In: *ACNS*. 2013.

Looping

- Some $N \in \mathbb{N}$ is such that $(g, \phi)^N = (1, id)$, so $s_{g,\phi}(N) = 1$

Looping

- Some $N \in \mathbb{N}$ is such that $(g, \phi)^N = (1, id)$, so $s_{g,\phi}(N) = 1$
- Let n be the smallest such integer; it follows that the set $\mathscr{C}_{g,\phi} = \{s_{g,\phi}(i) : i \in \mathbb{Z}\}$ has size n

Looping

- Some $N \in \mathbb{N}$ is such that $(g,\phi)^N = (1, id)$, so $s_{g,\phi}(N) = 1$
- Let n be the smallest such integer; it follows that the set $\mathscr{C}_{g,\phi} = \{s_{g,\phi}(i) : i \in \mathbb{Z}\}$ has size n

One has

$$\begin{aligned}(n+y) * s_{g,\phi}(x) &= y * s_{g,\phi}(x+n)\\ &= y * \phi^x(s_{g,\phi}(n))s_{g,\phi}(x)\\ &= y * s_{g,\phi}(x)\end{aligned}$$

Looping

- Some $N \in \mathbb{N}$ is such that $(g, \phi)^N = (1, id)$, so $s_{g,\phi}(N) = 1$
- Let n be the smallest such integer; it follows that the set $\mathscr{C}_{g,\phi} = \{s_{g,\phi}(i) : i \in \mathbb{Z}\}$ has size n

One has

$$\begin{aligned}(n+y) * s_{g,\phi}(x) &= y * s_{g,\phi}(x+n) \\ &= y * \phi^x(s_{g,\phi}(n)) s_{g,\phi}(x) \\ &= y * s_{g,\phi}(x)\end{aligned}$$

Theorem

$(\mathbb{Z}_n, \mathscr{C}_{g,\phi}, \circledast)$ *is a free, transitive group action.*

Looping

- Some $N \in \mathbb{N}$ is such that $(g, \phi)^N = (1, id)$, so $s_{g,\phi}(N) = 1$
- Let n be the smallest such integer; it follows that the set $\mathscr{C}_{g,\phi} = \{s_{g,\phi}(i) : i \in \mathbb{Z}\}$ has size n

One has

$$
\begin{aligned}
(n+y) * s_{g,\phi}(x) &= y * s_{g,\phi}(x+n) \\
&= y * \phi^x(s_{g,\phi}(n)) s_{g,\phi}(x) \\
&= y * s_{g,\phi}(x)
\end{aligned}
$$

Theorem
$(\mathbb{Z}_n, \mathscr{C}_{g,\phi}, \circledast)$ *is a free, transitive group action.*

For efficient sampling: how do we compute n?

Main Theorem

Theorem

$s_{g,\phi}(x) = \phi^{x-1}(g)...\phi(g)g$

Proof.

Induction - notice $s_{g,\phi}(x+1) = 1 * s_{g,\phi}(x) = \phi(s_{g,\phi}(x))g$ $\square$

Main Theorem

Theorem

$s_{g,\phi}(x) = \phi^{x-1}(g)...\phi(g)g$

Proof.

Induction - notice $s_{g,\phi}(x+1) = 1 * s_{g,\phi}(x) = \phi(s_{g,\phi}(x))g$ $\square$

Theorem

Let N be the order of (g,ϕ) as a $G \rtimes Aut(G)$ element, and n be the smallest integer such that $s_{g,\phi}(n) = 1$. Then $n|N$.

Main Theorem

Theorem

$s_{g,\phi}(x) = \phi^{x-1}(g)...\phi(g)g$

Proof.

Induction - notice $s_{g,\phi}(x+1) = 1 * s_{g,\phi}(x) = \phi(s_{g,\phi}(x))g$ $\square$

Theorem

Let N be the order of (g, ϕ) as a $G \rtimes Aut(G)$ element, and n be the smallest integer such that $s_{g,\phi}(n) = 1$. Then $n|N$.

Proof.

Certainly $N = kn + l$ for $k, l \in \mathbb{N}$, and $s_{g,\phi}(N) = 1$. We know

$$1 = \phi^{kn+l-1}(g)...\phi(g)g$$

$\square$

Main Theorem

Proof.

We can write the set $\{0, ..., kn + l - 1\}$ as

$$\begin{aligned}
0 + \{0, ..., n-1\} \\
n + \{0, ..., n-1\} \\
\cdots \\
(k-1)n + \{0, ..., n-1\} \\
kn + \{0, ..., l-1\}
\end{aligned}$$

Main Theorem

Proof.

We can write the set $\{0,...,kn+l-1\}$ as

$$\begin{aligned} 0+\{0,...,n-1\} \\ n+\{0,...,n-1\} \\ \cdots \\ (k-1)n+\{0,...,n-1\} \\ kn+\{0,...,l-1\} \end{aligned}$$

In other words

$$\phi^{kn}\left(\phi^{l-1}(g)...g\right)\prod_{i=0}^{k-1}\phi^{(k-(i+1))n}\left(\phi^{n-1}(g)...g\right)=1$$

Main Theorem

Proof.

We can write the set $\{0, ..., kn + l - 1\}$ as

$$
\begin{aligned}
0 + \{0, ..., n-1\} \\
n + \{0, ..., n-1\} \\
\cdots \\
(k-1)n + \{0, ..., n-1\} \\
kn + \{0, ..., l-1\}
\end{aligned}
$$

In other words

$$\phi^{kn}\left(\phi^{l-1}(g)...g\right)\prod_{i=0}^{k-1}\phi^{(k-(i+1))n}\left(\phi^{n-1}(g)...g\right) = 1$$

We therefore have $s_{g,\phi}(l) = 1$ with $l < n$, so $l = 0$. $\square$

Example

Let p be an odd prime, and define

$$G_p = \left\{ \begin{pmatrix} a & b \\ 0 & 1 \end{pmatrix} : a, b \in \mathbb{Z}_{p^2}, a \equiv 1 \mod p \right\}$$

Example

Let p be an odd prime, and define

$$G_p = \left\{ \begin{pmatrix} a & b \\ 0 & 1 \end{pmatrix} : a, b \in \mathbb{Z}_{p^2}, a \equiv 1 \mod p \right\}$$

- G_p has order p^3, turns out $|Aut(G_p)| = p^3(p-1)$

Example

Let p be an odd prime, and define

$$G_p = \left\{ \begin{pmatrix} a & b \\ 0 & 1 \end{pmatrix} : a, b \in \mathbb{Z}_{p^2}, a \equiv 1 \mod p \right\}$$

- G_p has order p^3, turns out $|Aut(G_p)| = p^3(p-1)$
- Follows that for any $(g, \phi) \in G_p \rtimes Aut(G_p)$, associated $n | p^6(p-1)$

Example

Let p be an odd prime, and define

$$G_p = \left\{ \begin{pmatrix} a & b \\ 0 & 1 \end{pmatrix} : a, b \in \mathbb{Z}_{p^2}, a \equiv 1 \mod p \right\}$$

- G_p has order p^3, turns out $|Aut(G_p)| = p^3(p-1)$
- Follows that for any $(g, \phi) \in G_p \rtimes Aut(G_p)$, associated $n | p^6(p-1)$
- 5 such values with additional restriction $n \leq p^3$

Example

Let p be an odd prime, and define

$$G_p = \left\{ \begin{pmatrix} a & b \\ 0 & 1 \end{pmatrix} : a, b \in \mathbb{Z}_{p^2}, a \equiv 1 \mod p \right\}$$

- G_p has order p^3, turns out $|Aut(G_p)| = p^3(p-1)$
- Follows that for any $(g, \phi) \in G_p \rtimes Aut(G_p)$, associated $n | p^6(p-1)$
- 5 such values with additional restriction $n \leq p^3$
- Each such check requires $\mathscr{O}(\log p)$ semidirect product group operations

Parameter Estimation

- Complexity of quantum attacks as a function of size of group in group action

[7] Wouter Castryck et al. "CSIDH: an efficient post-quantum commutative group action". In: *ASIACRYPT*. 2018, Xavier Bonnetain and André Schrottenloher. "Quantum security analysis of CSIDH". In: *EUROCRYPT*. 2020.

Parameter Estimation

- Complexity of quantum attacks as a function of size of group in group action
- Naive signature implementation impractical (several megabytes) - turns out we can boost challenge space (say to size S) at expense of public keys

[7] Wouter Castryck et al. “CSIDH: an efficient post-quantum commutative group action”. In: *ASIACRYPT*. 2018, Xavier Bonnetain and André Schrottenloher. “Quantum security analysis of CSIDH”. In: *EUROCRYPT*. 2020.

Parameter Estimation

- Complexity of quantum attacks as a function of size of group in group action
- Naive signature implementation impractical (several megabytes) - turns out we can boost challenge space (say to size S) at expense of public keys
- Borrowing estimates from isogeny group action[7] suggesting group size $\log n = 512$

[7] Wouter Castryck et al. "CSIDH: an efficient post-quantum commutative group action". In: *ASIACRYPT*. 2018, Xavier Bonnetain and André Schrottenloher. "Quantum security analysis of CSIDH". In: *EUROCRYPT*. 2020.

Parameter Estimation

- Complexity of quantum attacks as a function of size of group in group action
- Naive signature implementation impractical (several megabytes) - turns out we can boost challenge space (say to size S) at expense of public keys
- Borrowing estimates from isogeny group action[7] suggesting group size $\log n = 512$
- Tradeoffs available - at the short signature / long public key end signatures look like 8 pairs of $(\text{proof}, \text{challenge})$ with challenge space size 2^{16}
- Signatures of $538B$ at NIST 1 parameters

[7] Wouter Castryck et al. "CSIDH: an efficient post-quantum commutative group action". In: *ASIACRYPT*. 2018, Xavier Bonnetain and André Schrottenloher. "Quantum security analysis of CSIDH". In: *EUROCRYPT*. 2020.

Conclusions and Further Work

- Promising short signatures providing efficient sampling

Conclusions and Further Work

- Promising short signatures providing efficient sampling
- Checking of constants in asymptotic security estimates, implementation, optimisations, survey of appropriate groups etc required before concrete parameters can be given

Conclusions and Further Work

- Promising short signatures providing efficient sampling
- Checking of constants in asymptotic security estimates, implementation, optimisations, survey of appropriate groups etc required before concrete parameters can be given
- Shows the value of communication between fields! We learned from isogenists and used our results to augment the field

Conclusions and Further Work

- Promising short signatures providing efficient sampling
- Checking of constants in asymptotic security estimates, implementation, optimisations, survey of appropriate groups etc required before concrete parameters can be given
- Shows the value of communication between fields! We learned from isogenists and used our results to augment the field

$\sim$ The End $\sim$