Constructing Canonical Strategies For Parallel Implementation Of Isogeny Based Cryptography

Aaron Hutchinson and Koray Karabina
Florida Atlantic University

INDOCRYPT 2018

Acknowledgment: This research was supported by the Army Research Office Grant W911NF-17-1-0311

Outline

ECDH: Elliptic Curve Diffie-Hellman

$\langle P \rangle \subseteq E$

$$
\begin{tikzcd}
 & aP \arrow[dr, "{[b]}"] & \\
P \arrow[ur, "{[a]}"] \arrow[ddr, "{[b]}"'] & & baP \arrow[d, equal] \\
 & & abP \\
 & bP \arrow[ur, "{[a]}"'] &
\end{tikzcd}
$$

Elliptic curves and isogenies

Definition

Let (E_1, O_1) and (E_2, O_2) be elliptic curves. An **isogeny** from E_1 to E_2 is a rational map $\phi : E_1 \to E_2$ satisfying $\phi(O_1) = O_2$.

Theorem

Let E be an elliptic curve.

- *If H is a finite subgroup of E, then there exists an elliptic curve E' and an isogeny $\phi : E \to E'$ such that $\ker(\phi) = H$.*
- *If $\phi : E \to E_1$ and $\psi : E \to E_2$ are isogenies such that $\ker(\phi) = \ker(\psi)$, then there is an isomorphism $\alpha : E_1 \to E_2$ such that $\alpha\phi = \psi$.*

We write E/H for the curve E'.

SIDH: Supersingular Isogeny-based Diffie-Hellman

$\ker(\phi_A) = \langle m_A P_A + n_A Q_A \rangle$

$\ker(\phi'_B) = \langle m_B \phi_A(P_B) + n_B \phi_A(Q_B) \rangle$

$$
\begin{tikzcd}
 & E_A \arrow[dr, "\phi'_B"] \arrow[dd, dashed, shift right=2, "\substack{E_A \\ \phi_A(P_B) \\ \phi_A(Q_B)}"'] & \\
E \arrow[ur, "\phi_A"] \arrow[dr, "\phi_B"'] & & \substack{E_{BA} \\ \| \wr \\ E_{AB}} \\
 & E_B \arrow[ur, "\phi'_A"'] \arrow[uu, dashed, shift right=2, "\substack{E_B \\ \phi_B(P_A) \\ \phi_B(Q_A)}"'] &
\end{tikzcd}
$$

$\ker(\phi_B) = \langle m_B P_B + n_B Q_B \rangle$

$\ker(\phi'_A) = \langle m_A \phi_B(P_A) + n_A \phi_B(Q_A) \rangle$

Computational problems

- Given a curve $E/\mathbb{F}_q$ and a point $R \in E(\mathbb{F}_q)$ of order ℓ^n, compute a curve E_n, where $\phi : E \to E_n$ with kernel $\langle R \rangle$. Also, evaluate ϕ at some points.
- Velu's formulas are not very helpful when n is large.
- The decomposition strategy: Set $E_0 = E$, $R_0 = R$, and factor ϕ as a composition of n degree-ℓ isogenies ϕ_i, $i = 0, ..., n-1$:

$$\phi = \phi_{n-1} \circ \phi_{n-2} \circ \cdots \circ \phi_1 \circ \phi_0,\ \phi :\ E \to E_n,\ \text{Kernel}(\phi) = R,$$

with

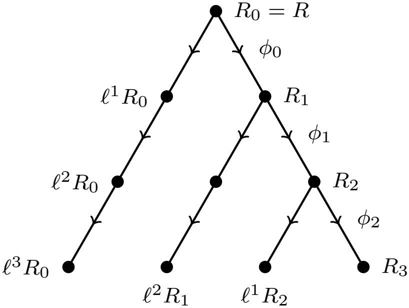
$$\phi_i :\ E_i \to E_{i+1},\ \text{Kernel}(\phi_i) = \ell^{n-i-1} R_i,\ R_{i+1} = \phi_i(R_i)$$

$$E = E_0 \xrightarrow{\phi_0} E_1 \xrightarrow{\phi_1} \cdots \xrightarrow{\phi_{n-2}} E_{n-1} \xrightarrow{\phi_{n-1}} E_n$$

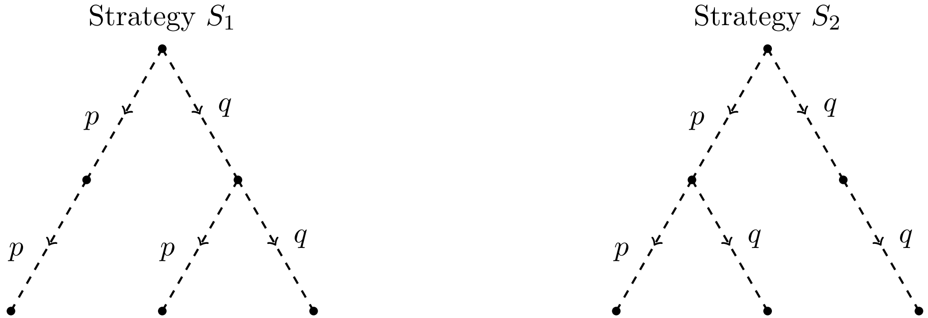
Traversing trees

$$E = E_0 \xrightarrow{\phi_0} E_1 \xrightarrow{\phi_1} E_2 \longrightarrow \cdots \xrightarrow{\phi_{n-1}} E_n$$

$$\ker(\phi_{n-1} \cdots \phi_2 \phi_1) = \langle R \rangle, \qquad \deg(\phi_i) = \ell$$



Two strategies: Serial vs. parallel



- Take $p = 1$, $q = 2$
- The cost of S_1 is $3p + 2q = 7$ and S_2 is $2p + 3q = 8$
- The parallelized cost of S_1 is $3p + 2q = 7$ and S_2 is $2p + 2q = 6$
- S_1 looses its optimality when parallelized

Parallelization of SIDH

- Evaluating a strategy S involves the following computations:
  (1) computation of elliptic curves E_i from a small subgroup H_i.
  (2) the evaluation of $[\ell]$ at varying points on varying curves.
  (3) the evaluation of isogenies at varying points on varying curves.

Theorem

Let S be a canonical strategy with $n \geq 3$ leaves and let a and b be distinct positive slope edges in S. Then a and b cannot be parallelized together.

Parallelization of SIDH

- $\mathcal{L}_i$: Positive slope diagonals indexed top-down
- $\mathcal{R}_i$: Negative slope diagonals indexed bottom-up
- P_i : Positive slope edges lying on $\mathcal{L}_{i+1}$
- Q_i : Negative slope edges lying between $\mathcal{L}_i$ and $\mathcal{L}_{i+1}$

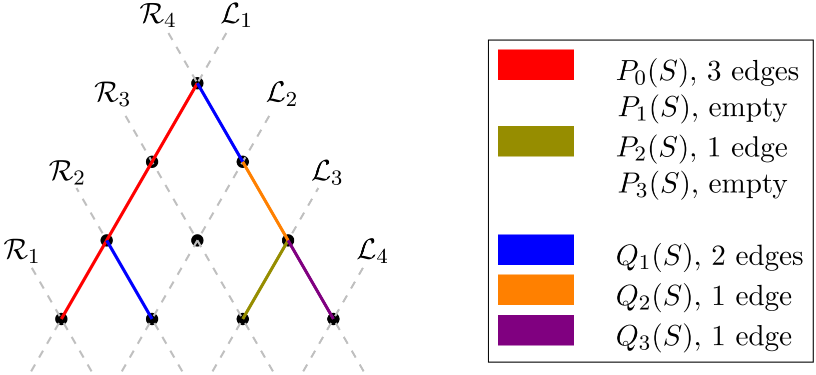


Figure: An example of the lines $\mathcal{L}_i$ and $\mathcal{R}_i$ and the bins $P_i(S)$ and $Q_i(S)$ on a strategy S with $n = 4$.

Parallelization of SIDH: PCP model

Parallelization Model (Per-Curve Parallel)

The only computations that we allow to be parallelized are isogeny evaluations which involve the same isogeny.

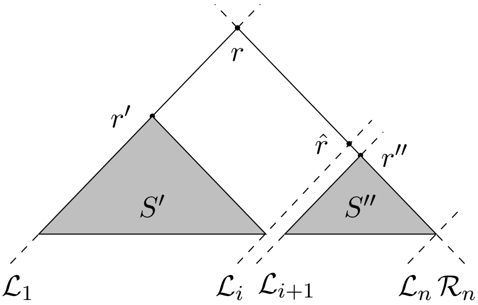
- Evaluate $P_0(S)$ in serial,
- Evaluate $Q_1(S)$ in parallel,
- Evaluate $P_1(S)$ in serial,
- Evaluate $Q_2(S)$ in parallel,
- $\vdots$ $\vdots$

Parallelization of SIDH: PCP model

Intuition:

- Cost of a strategy is the sum of the cost of the four pieces: $S' \cup r\hat{r}$, S'', rr', and $\hat{r}r''$
- rr' and $\hat{r}r''$ cannot be parallelized, and they cost $(n-i)p$ and q
- We write

$$
\begin{aligned}
C^K(S) &= C^K(S' \cup r\hat{r}) + C^K(S'') + C^K(rr') + C^K(\hat{r}r'') \\
&= C^K_{p,q}(S' \cup r\hat{r}) + C^K_{p,q}(S'') + (n-i)p + q.
\end{aligned}
$$



Parallelization of SIDH: PCP model

$$
\begin{aligned}
C^{k/K}(S) &= C^{k/K}(S' \cup r\hat{r}) + C^{k/K}(S'') + C^{k/K}(rr') + C^{k/K}(\hat{r}r'') \\
&= C^{k/K}_{p,q}(S' \cup r\hat{r}) + C^{k/K}_{p,q}(S'') + (n-i)p + q. \\
&= \begin{cases} C^{k-1/K}_{p,q}(S') + C^{k/K}_{p,q}(S'') + (n-i)p + q & \text{if } k > 1 \\ C^{K/K}_{p,q}(S') + C^{k/K}_{p,q}(S'') + (n-i)p + iq & \text{if } k = 1 \end{cases}
\end{aligned}
$$

Corollary

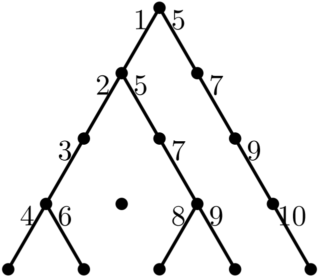
Minimizing $C^{k/K}(S'')$ *and*

$$
\begin{cases} C^{k-1/K}_{p,q}(S') & \textit{if } k > 1 \\ C^{K/K}_{p,q}(S') & \textit{if } k = 1 \end{cases}
$$

will minimize $C^{k/K}(S)$ *among strategies with partition* $(i, n-i)$.

A Toy example

$K = 2:$



(a) PCP Model

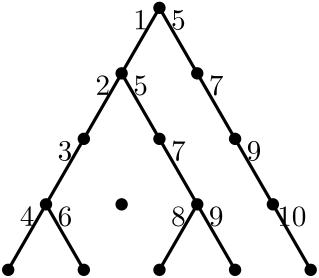
CCP: A Generalized model

- PCP suffers from idle processors

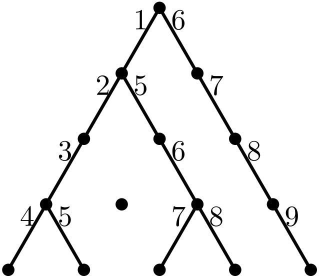
Parallelization Model (Consecutive-Curve Parallel)

Apply parallelization among:

- $Q_i(S) \cup Q_{i-1}(S)$ *for* $i = 2, 3, \ldots, n-1$,
- $P_i(S) \cup Q_i(S)$ *for* $i = 1, 2, \ldots, n-1$.



(a) PCP Model



(b) CCP Model

Parallelization of SIDH

- Algorithm computes $C^K_{p,q}(S)$ for a given S.
- Compared 3 sets for parameters $n = 186, p = 25.8, q = 22.8$:
  - Serially Optimal strategies (1,623,160)
  - PCP Optimal strategies (randomly sampled 5,000,000)
  - Canonical strategies (randomly sampled 5,000,000)

Results and remarks

- Introduced two models of parallelization
- Models are constructive with some optimality results

	K	2	3	4	5	6	7	8
PCP	Cost	25942.2	22521.6	20373.0	19197.0	17941.2	16978.8	16617.0
	% speedup	**24.27**	34.26	40.53	43.96	47.63	50.44	**51.49**
CCP S.O.	Cost	24247.2	21784.8	20941.2	20781.6	20781.6	20781.6	20781.6
	% speedup	**29.22**	36.41	38.87	39.34	39.34	39.34	**39.34**
CCP A.C.	Cost	25440.6	22200.6	20880.6	19825.2	19606.2	19218.6	18739.2
	% speedup	**25.73**	35.19	39.05	42.13	42.77	43.90	**45.30**
CCP P.O.	Cost	23890.2	20515.2	18252.6	17555.4	16482.0	16021.2	15294.6
	% speedup	**30.26**	40.11	46.72	48.75	51.89	53.23	**55.35**

Table: Data for parameters $n = 186, p = 25.8, q = 22.8$. Row PCP: optimal PCP costs over all canonical strategies. Row CCP S.O.: best CCP costs over all 1,623,160 serially optimal strategies. Row CCP A.C.: best CCP costs among 5,000,000 randomly sampled canonical strategies. Row CCP P.O: best CCP costs among 5,000,000 randomly sampled PCP optimal strategies. Percent speedup is over the optimal serial cost of 34256.4.

Future research

- Implement to verify results
- Try to find a formula for $C^K(n)$ under CCP