A ~~Magical~~ parallel variant of SIDH

Daniel Cervantes-Vázquez Eduardo Ochoa-Jiménez Francisco Rodríguez-Henrqúez

September 10, 2018

Story plot

- We present here a ~~magical~~ parallel variant of the Supersingular Isogeny Diffie-Hellman (SIDH) protocol, which is also applicable to the Supersingular Isogeny Key Encapsulation (SIKE) protocol.

Story plot

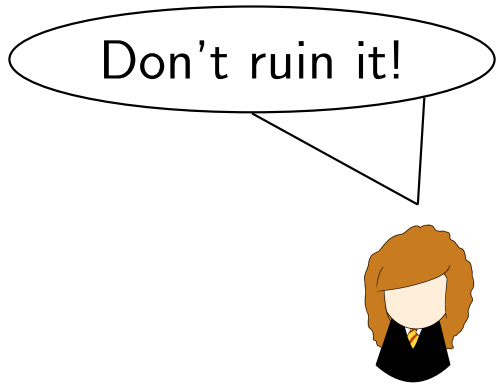
- We present here a ~~magical~~ parallel variant of the Supersingular Isogeny Diffie-Hellman (SIDH) protocol, which is also applicable to the Supersingular Isogeny Key Encapsulation (SIKE) protocol.
- This variant is illustrated by Hermione, Ron and Harry, who have learned from their charm class how to cast the "Curvaverto" spell.

Story plot

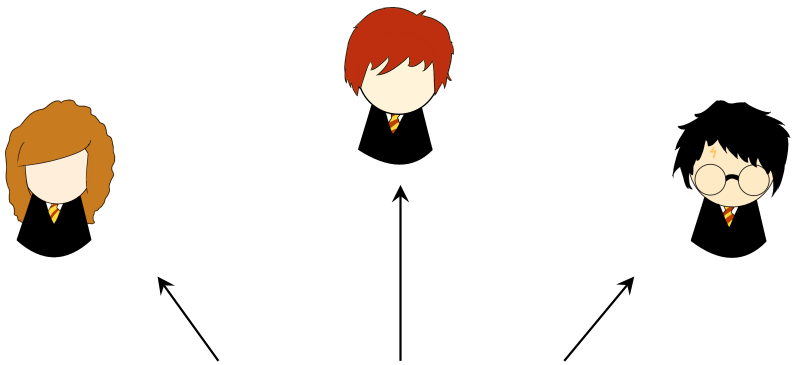
- We present here a ~~magical~~ parallel variant of the Supersingular Isogeny Diffie-Hellman (SIDH) protocol, which is also applicable to the Supersingular Isogeny Key Encapsulation (SIKE) protocol.
- This variant is illustrated by Hermione, Ron and Harry, who have learned from their charm class how to cast the "Curvaverto" spell.
- Given a magical stone called Kernel (a bunch of points belonging to an Elliptic Curve), then the *Curvaverto* spell transforms an Elliptic Curve and two magical stones into another Curve.

Story plot

- We present here a ~~magical~~ parallel variant of the Supersingular Isogeny Diffie-Hellman (SIDH) protocol, which is also applicable to the Supersingular Isogeny Key Encapsulation (SIKE) protocol.
- This variant is illustrated by Hermione, Ron and Harry, who have learned from their charm class how to cast the "Curvaverto" spell.
- Given a magical stone called Kernel (a bunch of points belonging to an Elliptic Curve), then the *Curvaverto* spell transforms an Elliptic Curve and two magical stones into another Curve.



Parameters



$$p := \quad 2^{e_2} \quad 3^{e_3} \quad 5^{e_5} \quad f - 1$$

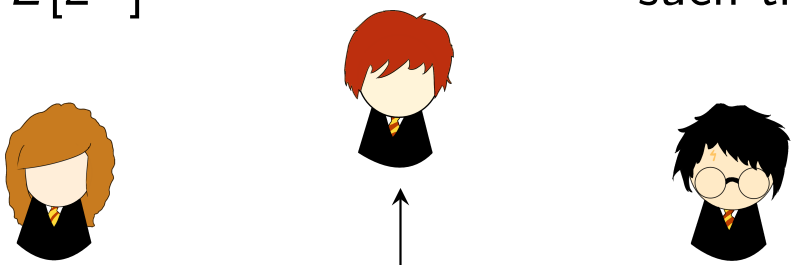
Such that $3^{e_3}5^{e_5} \approx 2^{e_2}$
and $3^{e_3} \approx 5^{e_5}$

Parameters

Choose P_3 and Q_3
such that $\langle P_3, Q_3 \rangle = E[3^{e_3}]$

Choose P_2 and Q_2
such that $\langle P_2, Q_2 \rangle = E[2^{e_2}]$

Choose P_5 and Q_5
such that $\langle P_5, Q_5 \rangle = E[5^{e_5}]$



$$p := 2^{e_2} 3^{e_3} 5^{e_5} f - 1$$

Such that $3^{e_3}5^{e_5} \approx 2^{e_2}$
and $3^{e_3} \approx 5^{e_5}$

Define $S := P_3 + P_5$ and $T := Q_3 + Q_5$
to be the public parameters of Ron and Harry

eSIDH

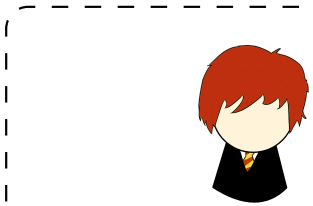
$K_2 := P_2 + [n_2]Q_2$

Get ϕ_H and E_H

E_H

E_0

eSIDH



Parallel

$K_3 := P_3 + [n_3]Q_3$ $K_5 := P_5 + [n_5]Q_5$

Get ϕ_R and E_R. Send $\phi_R(K_5)$ to Harry.

E_H

E_0

E_R

eSIDH

Use $\phi_R(K_5)$ to get E_{RH} and ϕ_{RH}

E_H

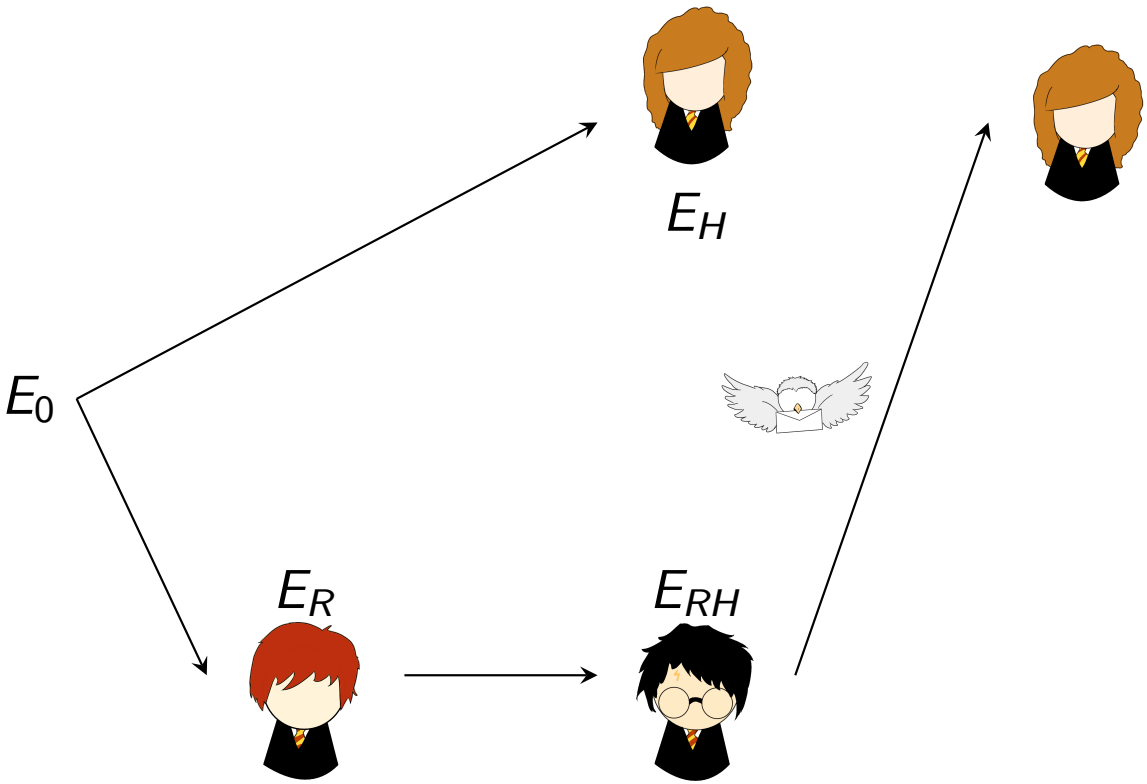
E_0

E_R

E_{RH}

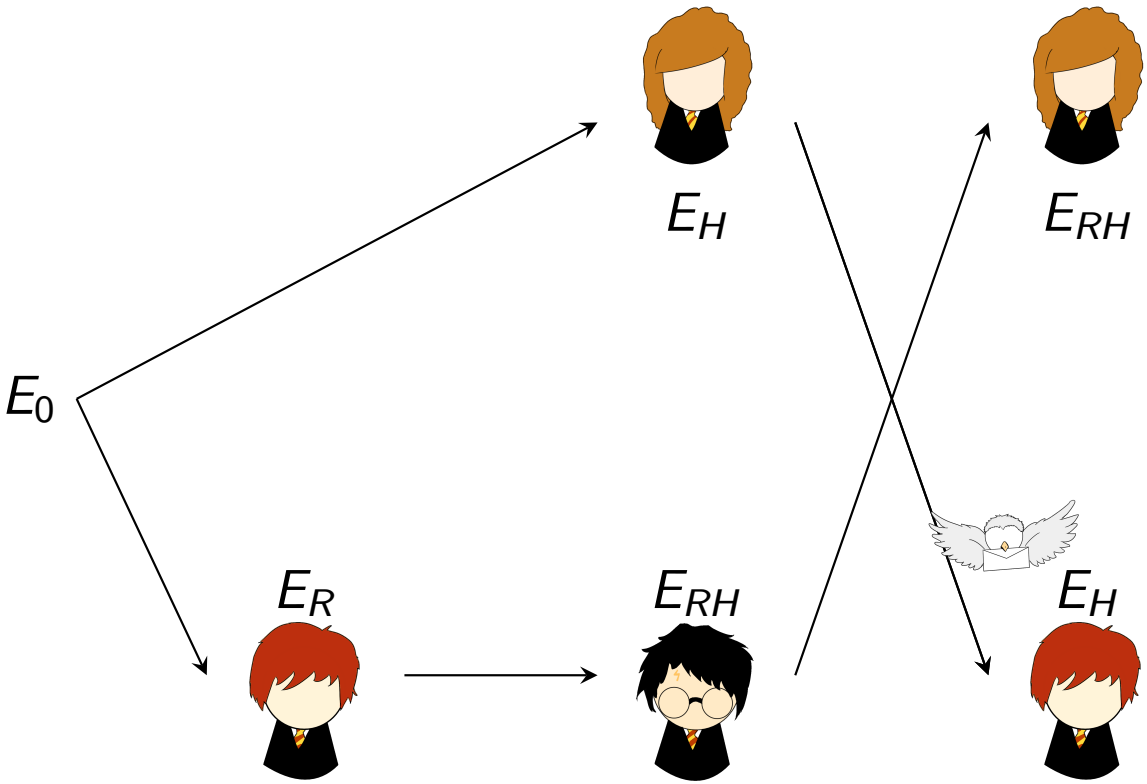
eSIDH

$(E_{RH}, \phi_{RH}(P_2), \phi_{RH}(Q_2))$



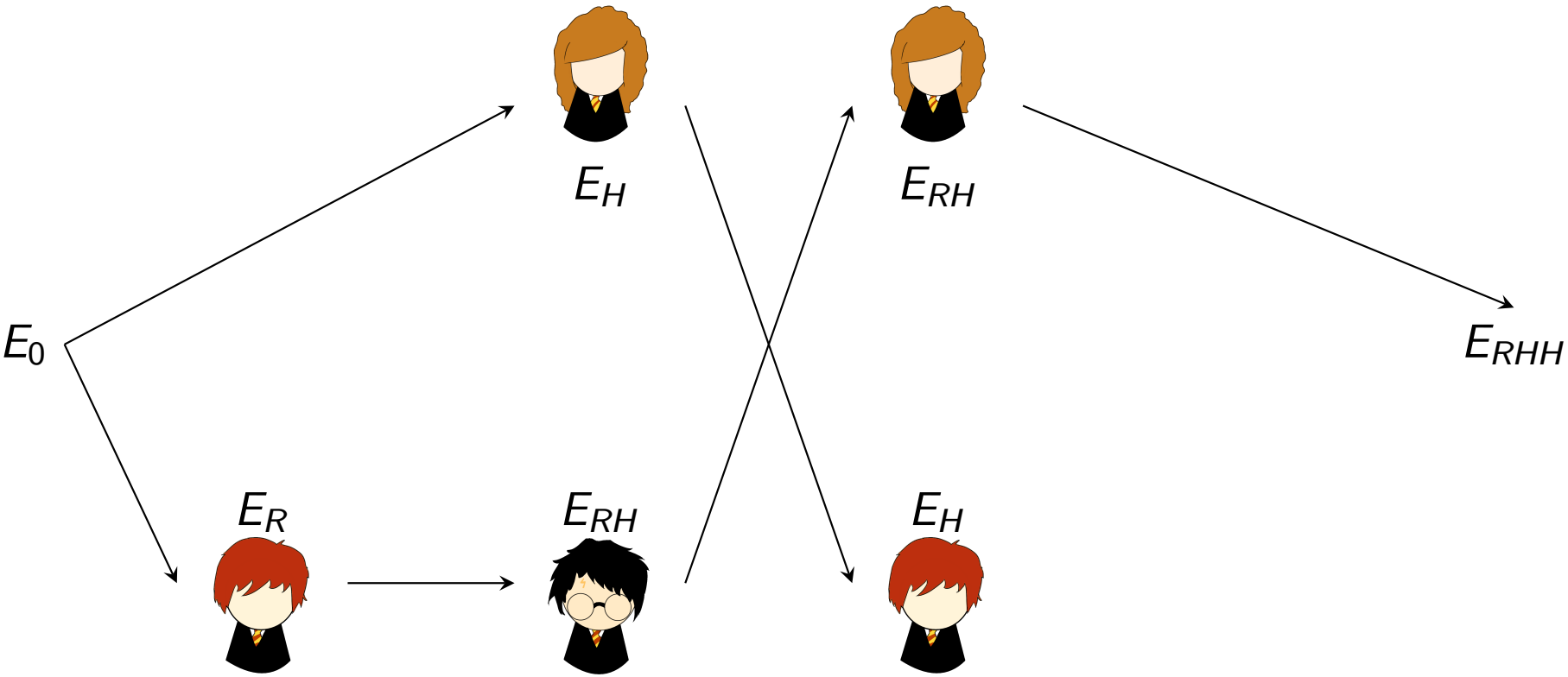
eSIDH

$(E_H, \phi_H(S), \phi_H(T))$

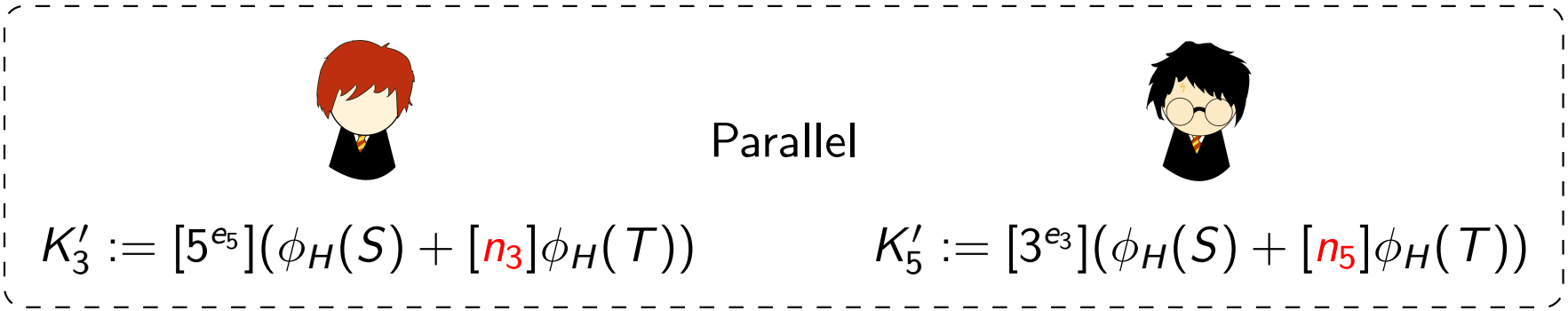


eSIDH

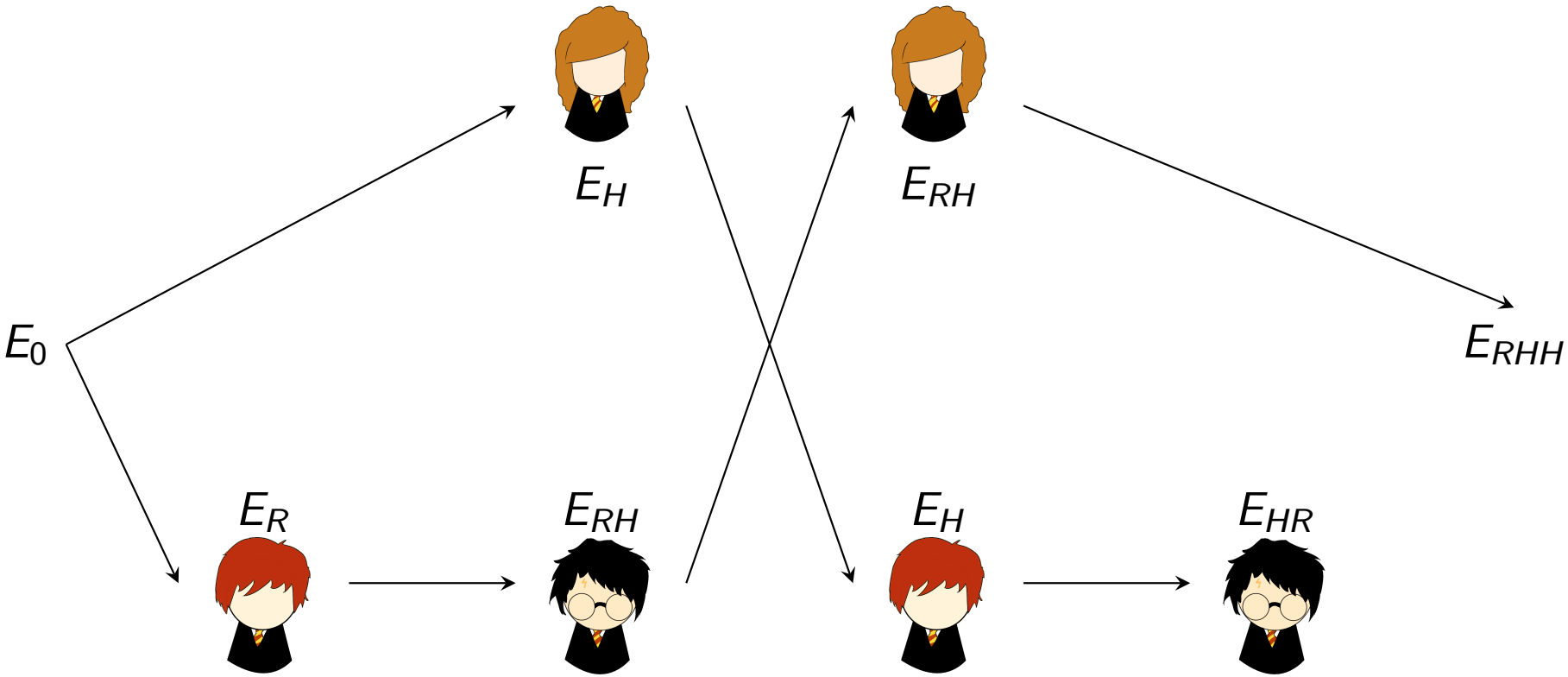
$$K_2' := \phi_{RH}(P_2) + [n_2]\phi_{RH}(Q_2)$$
Get E_{RHH}



eSIDH



Get ϕ_R' and E_R'. Send $\phi_R'(K_5')$ to Harry.



eSIDH

Use $\phi'_R(K'_5)$ to get E_{HRH}

E_H

E_{RH}

E_0

$E_{RHH} \cong E_{HRH}$

E_R

E_{RH}

E_H

E_{HR}

Primes and Times

Our proposals	**[SIKE17]** proposals
$P_{509} = 2^{250}3^{79}5^{55}2^{6} - 1$	$P_{503} = 2^{250}3^{159} - 1$
$P_{765} = 2^{372}3^{119}5^{81}2^{16} - 1$	$P_{751} = 2^{372}3^{239} - 1$
$P_{1013} = 2^{486}3^{157}5^{108}2^{26} - 1$	$P_{964} = 2^{486}3^{301} - 1$

Table: Our proposals for eSIDH primes in comparison with the current state-of the art

Protocol phase		SIKE17 p_{503}	**Ours** p_{509} Non Parallel	**AF**	SIKE17 p_{751}	**Ours** p_{765} Non Parallel	**AF**	**Ours** p_{1013} Non Parallel
KeyGen	Alice	8.24	7.48	1.10	23.68	22.21	1.06	49.24
	Bob	9.26	8.26	1.12	26.67	24.53	1.08	55.18
KeyAgr	Alice	6.71	6.08	1.10	19.44	18.20	1.06	40.83
	Bob	7.82	7.73	1.01	22.76	22.98	0.99	52.05

Table: Performance comparison of this proposal against SIKE17 (using the version 3 of the CLN library). Reported running time (in 10^6 clock cycles) was measured in an Intel Skylake proccessor at 4.0 GHz. We report here the sequential version performance using 1 core.

Primes and Times

Our proposals	**[SIKE17]** proposals
$P_{509} = 2^{250}3^{79}5^{55}2^{6} - 1$	$P_{503} = 2^{250}3^{159} - 1$
$P_{765} = 2^{372}3^{119}5^{81}2^{16} - 1$	$P_{751} = 2^{372}3^{239} - 1$
$P_{1013} = 2^{486}3^{157}5^{108}2^{26} - 1$	$P_{964} = 2^{486}3^{301} - 1$

Table: Our proposals for eSIDH primes in comparison with the current state-of the art

Protocol phase		SIKE17 p_{503}	**Ours** p_{509} Parallel	**AF**	SIKE17 p_{751}	**Ours** p_{765} Parallel	**AF**	**Ours** p_{1013} Parallel
KeyGen	Alice	8.24	5.91	1.39	23.68	16.68	1.42	36.35
	Bob	9.26	5.58	1.66	26.67	15.99	1.67	34.73
KeyAgr	Alice	6.71	5.40	1.24	19.44	15.20	1.28	32.88
	Bob	7.82	5.74	1.36	22.76	16.55	1.37	35.75

Table: Performance comparison of this proposal against SIKE17 (using the version 3 of the CLN library). Reported running time (in 10^6 clock cycles) was measured in an Intel Skylake proccessor at 4.0 GHz. We report here the parallel version performance using 3 cores.

Work in Progress

- Working on Ron-Harry side (Bob's side), extend this proposal to other combinations of small primes [instead of the current $(3, 5)$].
- Look for more Montgomery-friendly primes.
- Further optimize the single-core version of this proposal.