Efficiency of SIDH/Isogeny Signatures

Robert Gorrie

Department of Computing & Software, McMaster University

November 16th, 2017

Isogeny Based Signatures

Inversion Batching

Signature Compression

Isogeny Based Signatures

Inversion Batching

Signature Compression

Current Performance of SIDH

Scheme	Size (bytes)			Security	
	public key	secret key	$\operatorname{signature}$	certificate	(bits)
Lattice-based					
GLP 18	1 536	256	1 186	3.0 KiB	100
Ring-TESLA-II	3 328	1920	1488	$5.1\mathrm{KiB}$	128
TESLA#-I 🖪	3328	2112	1616	$5.2\mathrm{KiB}$	128
BLISS 15	7 168	2048	1559	$9.0\mathrm{KiB}$	128
TESLA-416 [2]	1331200	1011744	1280	$1332.8\mathrm{KiB}$	128
Hash-based					
XMSS 9	912	19	2451	$3.6\mathrm{KiB}$	82
SPHINCS 5	1,056	1,088	41,000	$42.3\mathrm{KiB}$	> 128
Rainbow [11]	44160	86240	37	$44.5\mathrm{KiB}$	80

Isogeny Based Signatures

- Yoo et. al provide an isogeny based signature scheme built off the Microsoft SIDH 1.0 Library.
- The scheme is constructed using the ZKPol protocol provided in the original SIDH paper in tandem with Unruh's PQ secure Fiat-Shamir transform.
- The scheme involves performing 248 (seperate) instances of SIDH key exchange with an arbitrary thirdparty
- These instances are parallelizable but overall extremely computationally expensive

Isogeny Signature Parameter Sizes

Scheme	Public-key size	Private-key size	Signature size	
Hash-based	1,056	1,088	41,000	
Code-based	192,192	1,400,289	370	
Lattice-based	7,168	2,048	5,120	
Ring-LWE-based	7,168	4,608	3,488	
Multivariate-based	99,100	74,000	424	
Isogeny-base	768	48	141,312	

Isogeny Based Signatures

Inversion Batching

Signature Compressior

Partial Inversion Procedure

1:
$$t_0 \leftarrow a_0^2$$

2: $t_1 \leftarrow a_1^2$

2:
$$t_1 \leftarrow a_1^-$$

3:
$$den \leftarrow t_0 + t_1$$

4:
$$den \leftarrow den^{-1}$$

5:
$$a_0 \leftarrow a_0 * den$$

6:
$$a_1$$
 ← a_1 * den

Batched Inversion Procedure

If we combine these two procedures we can reduce n \mathbb{F}_{p^2} inversions to:

- 1 \mathbb{F}_p inversion
- 3(n-1) \mathbb{F}_p multiplications
- $2n \mathbb{F}_p$ multiplications
- $2n \mathbb{F}_p$ squarings

Or, roughly $1 \mathbb{F}_p$ inversion and $7n \mathbb{F}_p$ multiplications

Performance Increase

The following are measured in billions of clock cycles

Procedure	Without Batching	With Batching	
Signature Sign	15.74	15.56	
Sign Parallel	10.23	10.13	
Signature Verify	11.18	10.8	
Verify Parallel	7.27	7.11	

- In the serial setting we see a 1.1% and a 3.5% performance increase for Signing and Verifying, respectively.
- Comparatively, in the parallel setting we see a 0.9% and a 2.3% performance increase.

Isogeny Based Signatures

Inversion Batching

Signature Compression

SIDH Key Compression

A recent paper from Microsoft Research showed that SIDH public keys could be compressed to 330 bytes while retaining 128 bits of security in the quantum setting.

Signature Compression

Results

Isogeny Based Signatures

Inversion Batching

Signature Compressior