,

Time-Optimal Design of Finite Field Arithmetic for SIKE on Cortex-M4

Mila Anastasova[1], Reza Azarderakhsh[1] and Mehran Mozaffari Kermani[2]

[1] CEECS Department and I-SENSE at Florida Atlantic University, Boca Raton, FL, USA
(manastasova2017, razarderakhsh)@fau.edu

[2] CES Department at University of South Florida, Tampa, FL, USA,
mehran2@usf.edu

August 2022

Content

Introduction

- Public key cryptography is essential for data confidentiality and integrity of data conveyed across an unsecured channel.
- The classical cryptographic protocols, however, such as RSA and ECC relying on the difficulty of factoring large prime numbers and the Elliptic Curve Discrete Logarithm Problem (ECDLP), are vulnerable to quantum attacks.
- Thus, a transition to post-quantum robust protocols was initialized by NIST [9] to offer secure data transmission in the era of large-scale quantum computers.

Supersingular Isogeny Key Encapsulation

Key Generation	**Encapsulation**	**Decapsulation**
Input : - *Output* : s, sk_A, pk_A 1. $sk_A \in_R \mathbb{Z}/2^{e_A}\mathbb{Z}$ 2. $\phi_A : E_0 \rightarrow E_A$ with $ker(\phi_A) = \langle P_A + [sk_A]Q_A \rangle$ 3. $pk_A = (E_A, \phi_A(P_B), \phi_A(Q_B))$ 4. $s \in_R \{0,1\}^t$	*Input* : pk_A *Output* : c, ss 1. $m \in_R \{0,1\}^t$ 2. $r = H(m \| pk_A) mod 3^{e_B}$ 3. $\phi_B : E_0 \rightarrow E_B$ with $ker(\phi_B) = \langle P_B + [r]Q_B \rangle$ 4. $pk_B = \{E_B, \phi_B(P_A), \phi_B(Q_A)\}$ 5. $\phi'_B : E_A \rightarrow E_{AB}$ with $ker(\phi'_B) = \langle \phi_A(P_B) + [r]\phi_A(Q_B) \rangle$ 6. $c = (pk_B \| K(j(E_{AB})) \oplus m)$ 7. $ss = (J(m \| c))$	*Input* : s, sk_B, pk_B, c *Output* : ss 1. $\phi'_A : E_B \rightarrow E_{BA}$ with $ker(\phi'_A) = \langle \phi_B(P_A) + [sk_A]\phi_B(Q_A) \rangle$ 2. $m' = c_1 \oplus K(j(E_{BA}))$ 3. $r' = H(m' \| pk_A) mod 3^{e_B}$ 4. $\phi''_A : E_0 \rightarrow E_{B'}$ with $ker(\phi''_A) = \langle P_B + [r']Q_B \rangle$ 5. $pk'_B = \{E_{B'}, \phi''_A(P_A), \phi''_A(Q_A)\}$ 6. IF $pk'_B = pk_B$ $ss = (J(m' \| c))$ ELSE $ss = (J(s \| c))$

Figure – SIKE algorithm [5]. H, K and J denote hash functions. $p = 2^{e_A}3^{e_B} - 1$, $E_0/\mathbb{F}_{p^2}$, $\{P_A, Q_A\}$ and $\{P_B, Q_B\}$ are public parameters.

- The SIKE protocol, which is based on pseudorandom walks on isomorphic graphs, assures that both communication parties reach a shared secret based on a curve $j-$invariant.

Related Work

- The pyramidal structure of the isogeny-based protocol permits optimization of its many levels.
  - As a result, there have been several research and engineering groups devoted to optimizing the higher-level isogeny optimizations of the SIKE protocol in order to find an optimal solution for the calculation of the heavy isogeny maps [3, 2, 4, 10].
- Focusing on the lowest layer of the computational pyramid of SIKE, there are several implementations, targeting resource constrained devices.
  - The authors in [1, 7] provide implementation solutions for the low-level finite filed arithmetic of SIKE and achieve a record speedup on the target platform, running the SIKE protocol in 139MCCs for security Level I.

ARM Cortex-M4

NIST recommended microcontroller for benchmarking.

Features

ARMv7-M architecture

16 32-bit core registers

32 32-bit FP registers

1 CC per instruction except memory accesses

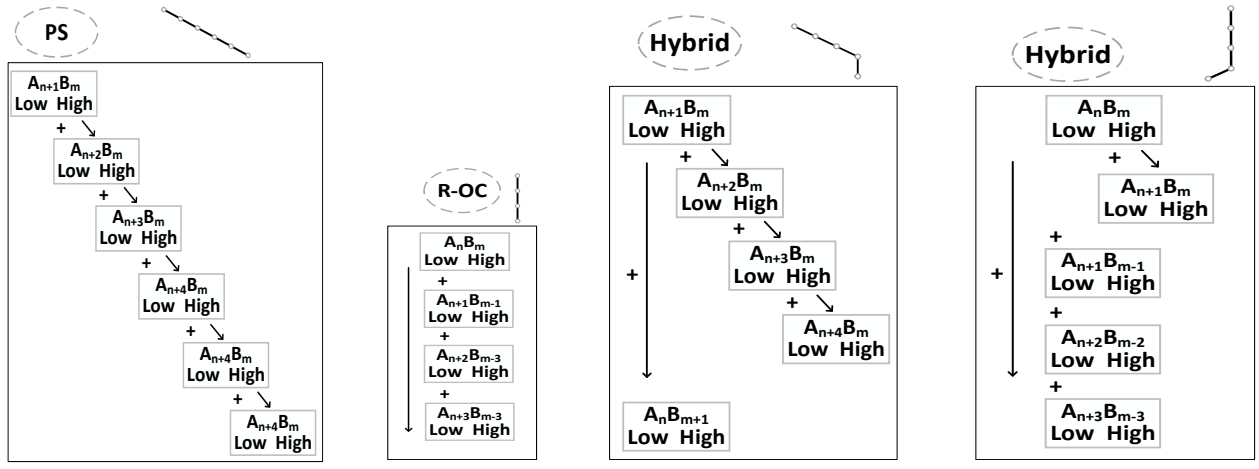
Implementation strategies

Use the entire register set.

Operate on larger operand sets.

Re-organize the instruction flow for efficient design.

Proposed Design - Notation



(a) Product Scanning (PS)

(b) R-OC, identical to OS

(c) Hybrid (4 PS&1 OS).

(d) Hybrid (1 PS&4 OS).

Figure – Deployed list of inner multi-precision loop execution flows along with the associated assembly instruction set.

Multi-precision Multiplication

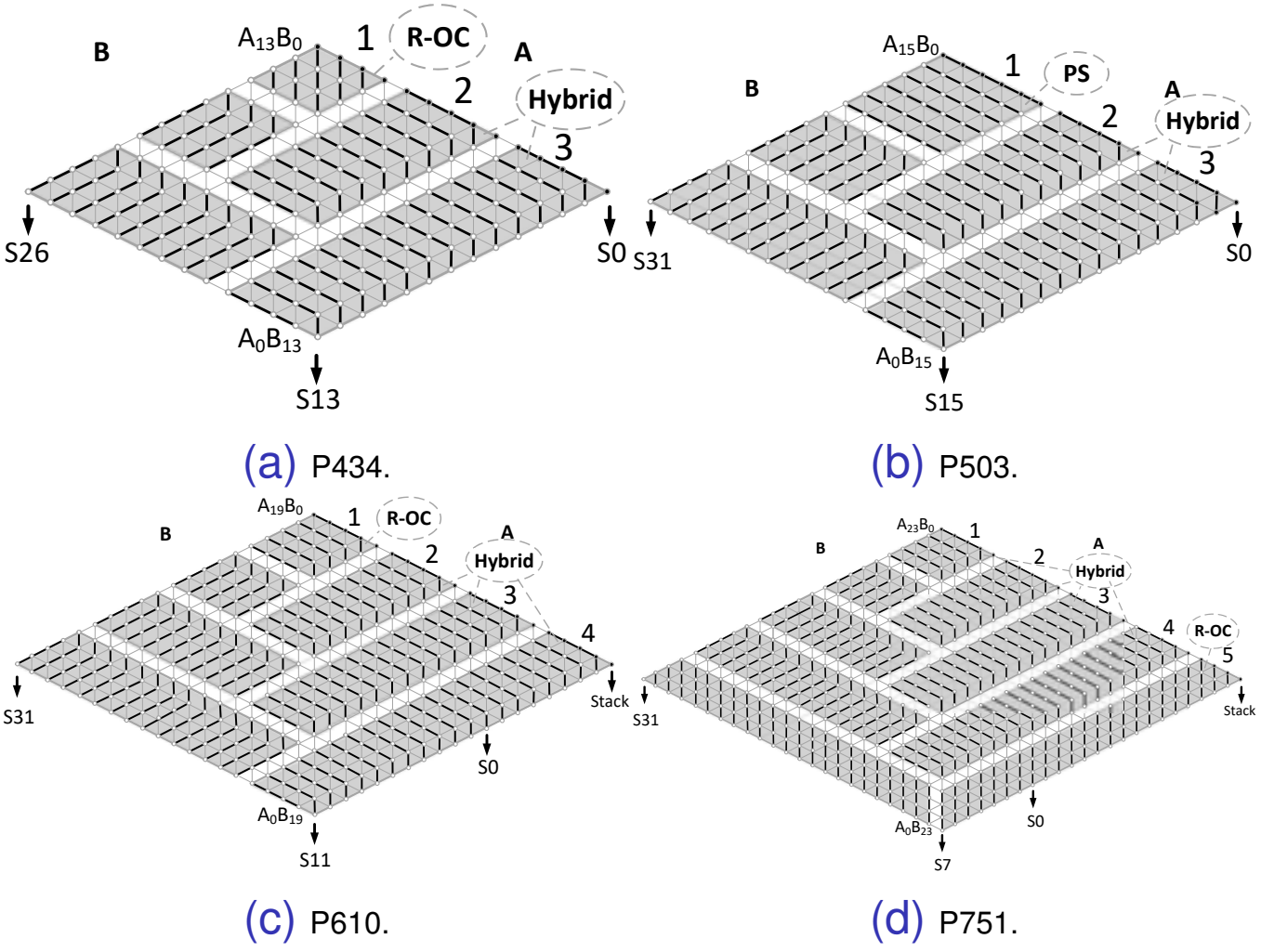
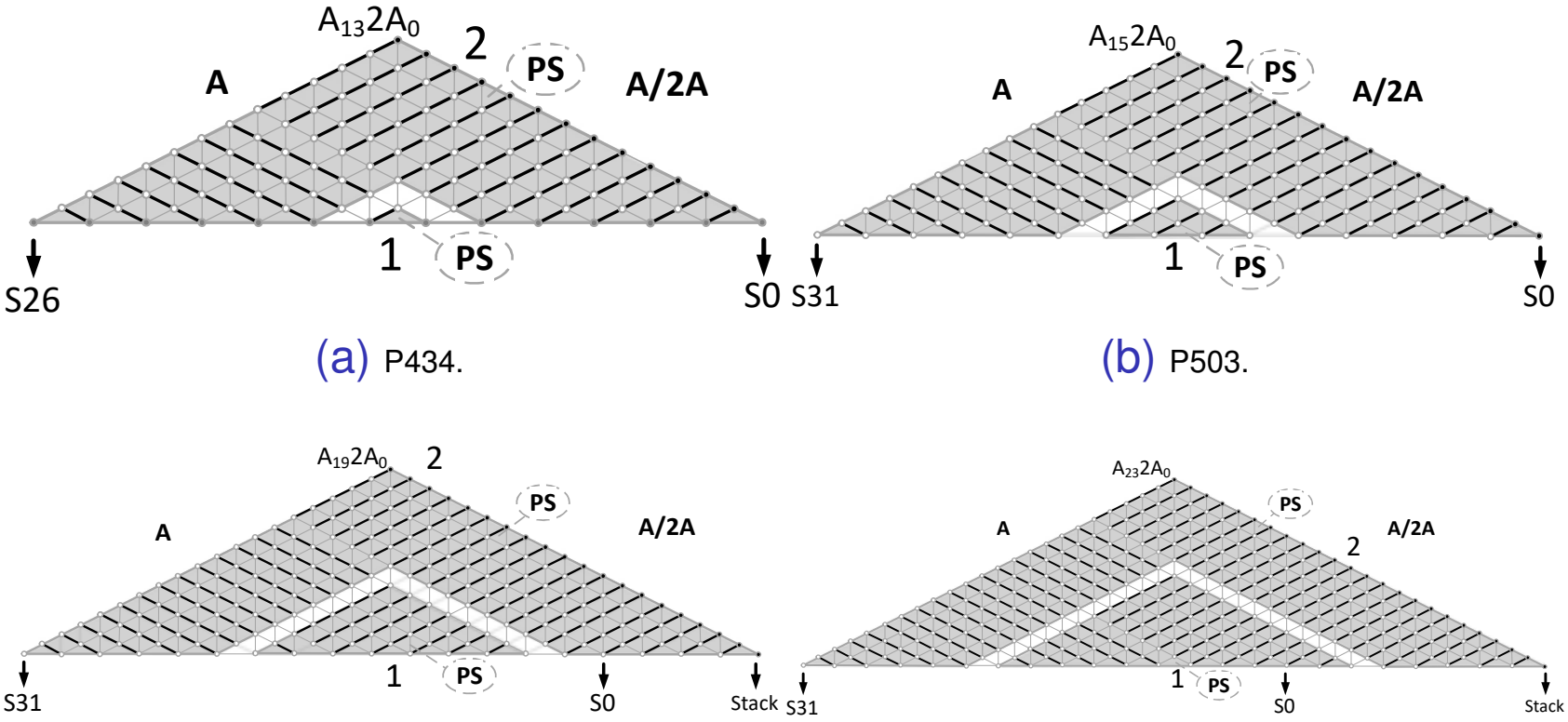


Figure – Proposed architecture for multi-precision multiplication.

Multi-precision Squaring



(a) P434. (b) P503.

(c) P610. (d) P751.

Figure – Proposed architecture for the implementation of multi-precision squaring for all SIKE primes.

Multi-precision Reduction

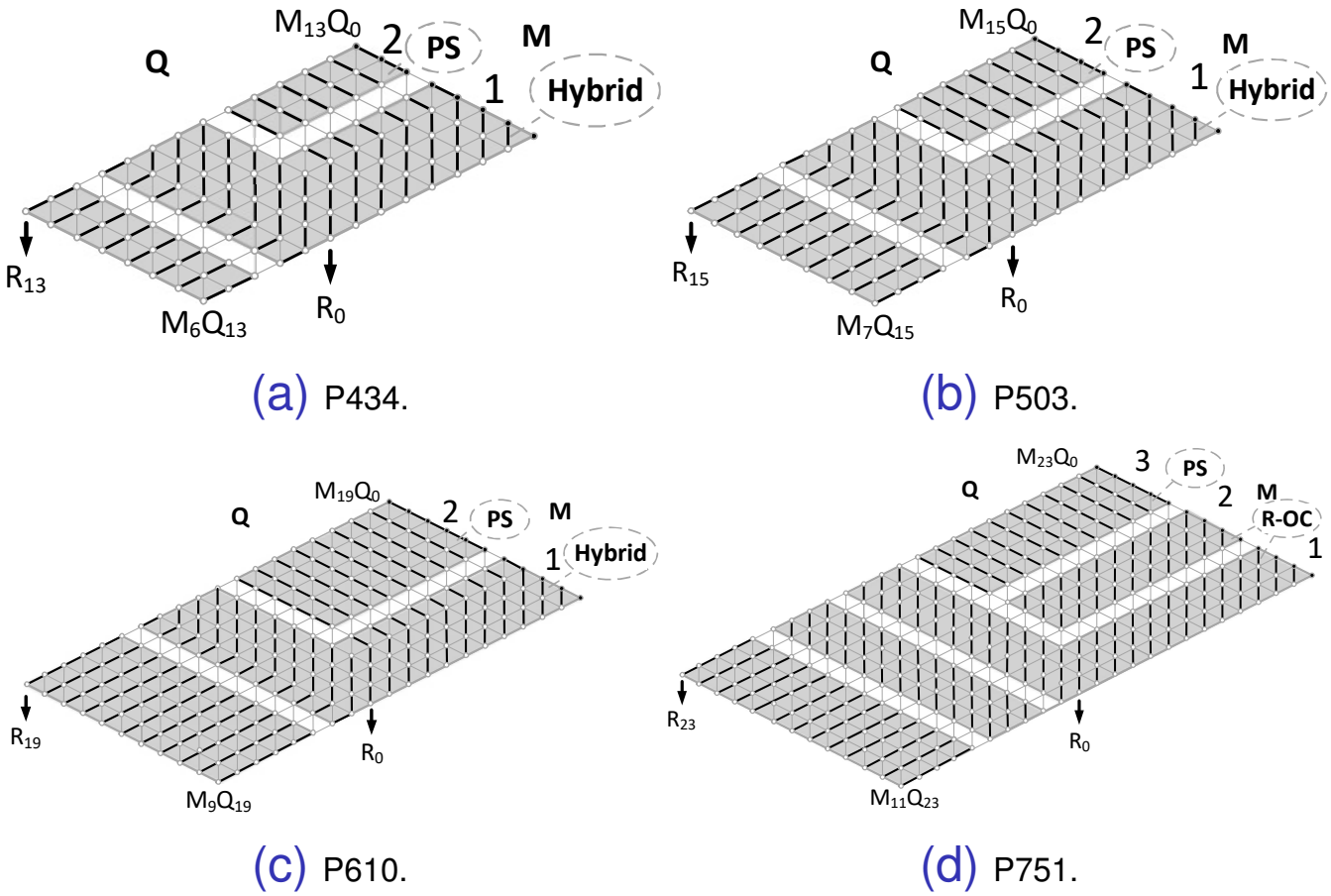


Figure – Proposed architecture for the implementation of multi-precision reduction for all SIKE primes.

Results Subroutine Latency

Table – SIKE finite field arithmetic latency targeting STM32F407VG

Implementation	Latency [CC]							
	$\mathbb{F}_p$mul	$\mathbb{F}_p$sqr	$\mathbb{F}_p$mul	$\mathbb{F}_p$sqr	$\mathbb{F}_p$mul	$\mathbb{F}_p$sqr	$\mathbb{F}_p$mul	$\mathbb{F}_p$sqr
	SIKEp434		SIKEp503		SIKEp610		SIKEp751	
SIDH v3.3 [6]	17,964	17,964	23,364	23,364	35,047	35,047	49,722	49,722
Seo et al. [8]	1,110	981	1,333	1,139	-	-	2,744	2,242
Seo et al. [8]	1,011	889	1,221	1,024	1,869	1,535	2,577	2,066
Anastasova et al. [1]	769	594	952	734	1,506	1,171	2,103	1,543
This work	**702**	**563**	**895**	**684**	**1,435**	**997**	**2,062**	**1,444**

Performance Results @24MHz

Table – Report of SIKE timing results in terms of clock cycles and speedup percentage on STM32F407 running @24MHz

Implementation	Timing [cc×10^6]							
	KeyGen	**Encaps**	**Decaps**	**Total**	**KeyGen**	**Encaps**	**Decaps**	**Total**
	SIKEp434				SIKEp503			
SIDH v3.3 [6]	650	1,065	1,136	2,202	985	1,623	1,726	3,350
Seo et al. [8]	74	122	130	252	104	172	183	355
Seo et al. [8]	54	87	94	181	74	121	129	250
Anastasova et al. [1]	41	67	72	139	58	96	102	197
This work	**39.0**	**63.6**	**68.0**	**131.6**	**55.9**	**91.8**	**97.7**	**189.5**

Performance Results @24MHz

Table – Report of SIKE timing results in terms of clock cycles and speedup percentage on STM32F407 running @24MHz

Implementation	Timing [cc×10^6]							
	KeyGen	**Encaps**	**Decaps**	**Total**	**KeyGen**	**Encaps**	**Decaps**	**Total**
	SIKEp610				SIKEp751			
SIDH v3.3 [6]	1,819	3,348	3,368	6,716	3,296	5,347	5,742	11,089
Seo et al. [8]	-	-	-	-	282	455	491	946
Seo et al. [8]	131	241	243	484	225	365	392	757
Anastasova et al. [1]	106	195	196	391	182	295	317	613
This work	**102.5**	**188.1**	**189.3**	**377.4**	**179.4**	**290.5**	**312.1**	**602.7**

Conclusions

- We observe 8.71%, 5.99%, 4.46%, and 2.04% of latency reduction for the execution of modular multiplication based on prime lengths of 434-, 503-, 610-, and 751-bits, respectively. We achieve 5.38%, 6.43%, 14.64%, and 6.42% of speedup compared to the counterparts in [1] for the modular squaring routine.
- We integrate the suggested multi-precision multiplication, squaring and reduction routines in the SIKE implementation and we obtain more than 5.6% of speedup for SIKEp434. We report 3.93%, 3.48%, and 1.61% of latency reduction for SIKEp503, SIKEp610, and SIKEp751, respectively.

References

[1] M. Anastasova, R. Azarderakhsh, and M. M. Kermani. Fast strategies for the implementation of SIKE round 3 on ARM Cortex-M4. *IEEE Transactions on Circuits and Systems I : Regular Papers*, 68(10) :4129–4141, 2021.

[2] C. Costello and H. Hisil. A Simple and Compact Algorithm for SIDH with Arbitrary Degree Isogenies. In *Advances in Cryptology – ASIACRYPT 2017 - 23rd International Conference on the Theory and Application of Cryptology and Information Security*, pages 303–329, 2017.

[3] C. Costello, P. Longa, and M. Naehrig. Efficient algorithms for supersingular isogeny Diffie-Hellman. In *Annual International Cryptology Conference*, pages 572–601. Springer, 2016.

[4] C. Costello, P. Longa, M. Naehrig, J. Renes, and F. Virdia. Improved Classical Cryptanalysis of the Computational Supersingular Isogeny Problem. Cryptology ePrint Archive, Report 2019/298. `https://eprint.iacr.org/2019/298`.

[5] D. Jao, R. Azarderakhsh, M. Campagna, C. Costello, L. De Feo, B. Hess, A. Jalali, B. Koziel, B. LaMacchia, P. Longa, M. Naehrig, J. Renes, V. Soukharev, and D. Urbanik. Supersingular Isogeny Key Encapsulation. Submission to the NIST Post-Quantum Standardization Project, 2017. `https://sike.org/`.

[6] Microsoft Team. Sidh library. `https://github.com/Microsoft/PQCrypto-SIDH`.

[7] H. Seo, M. Anastasova, A. Jalali, and R. Azarderakhsh. Supersingular isogeny key encapsulation (SIKE) round 2 on ARM Cortex-M4. *IEEE Transactions on Computers*, 70(10) :1705–1718, 2020.

[8] H. Seo, A. Jalali, and R. Azarderakhsh. SIKE round 2 speed record on ARM Cortex-M4. In *International Conference on Cryptology and Network Security*, pages 39–60. Springer, 2019.

[9] The National Institute of Standards and Technology (NIST). Post-quantum cryptography standardization, 2017-2018. `https://csrc.nist.gov/Projects/post-quantum-cryptography`.

[10] J. Tian, P. Wang, Z. Liu, J. Lin, Z. Wang, and J. Groszschaedl. Efficient software implementation of the SIKE protocol using new data representation. *IEEE Transactions on Computers*, 2021.