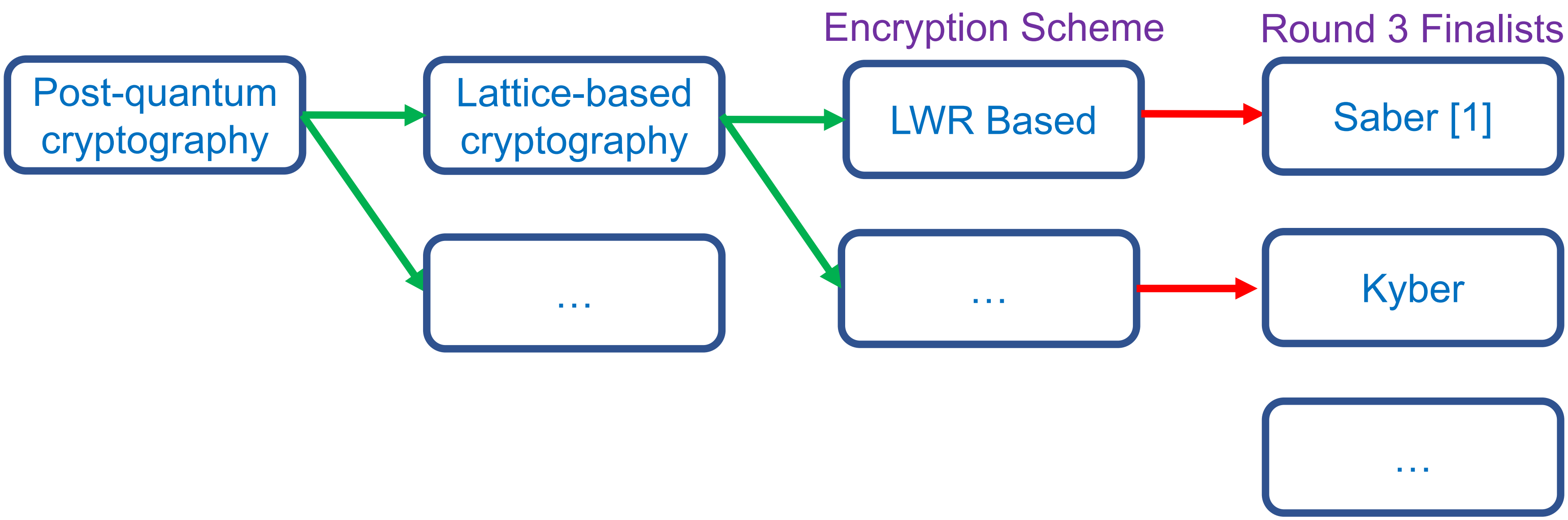
Compact Coprocessor for KEM Saber: Novel Scalable Matrix Originated Processing

- Pengzhou He[1] , Chiou-Yng Lee[2] , Jiafeng Xie[1] (corresponding author)
- [1]: Department of Electrical & Computer Engineering, Villanova University
- [2]: Department of Computer Information & Network Engineering, Lunghwa University of Science & Technologysity

Content

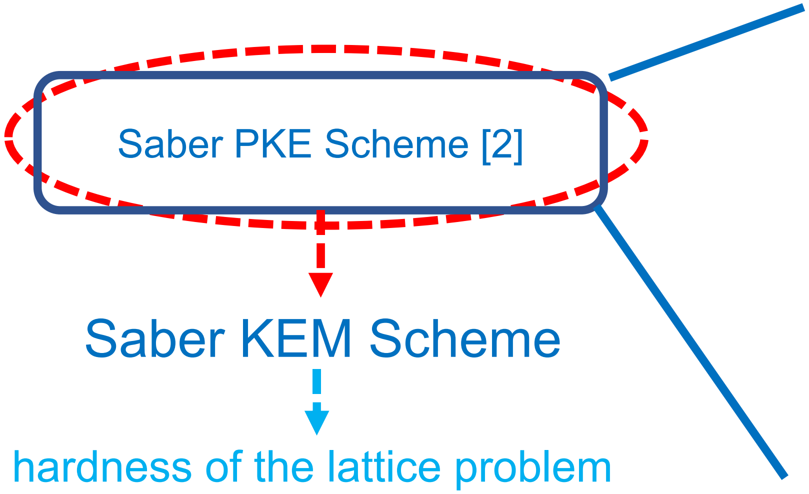
- Preliminary
- Proposed Method
- Proposed PQC Structures
- Implementation & Comparison
- Conclusions
- Future Work

Preliminary Knowledge



[1]: https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions

Saber KEM Scheme

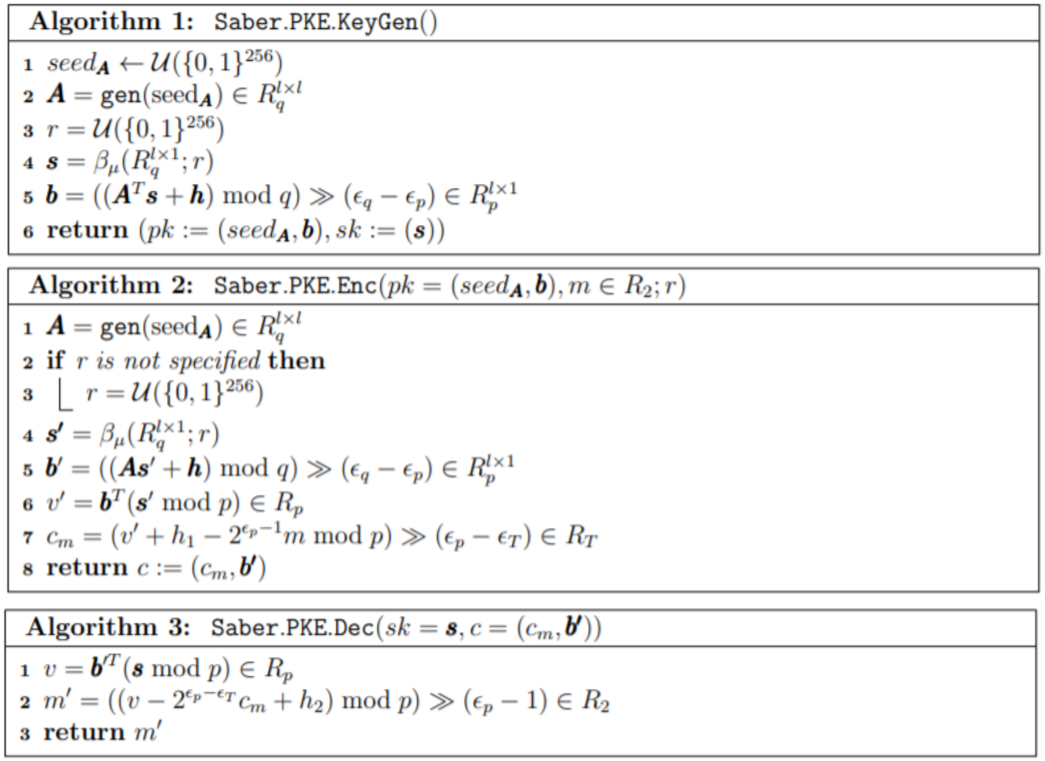


Saber PKE Scheme [2]

Saber KEM Scheme

hardness of the lattice problem

scheme	n	k	NIST security level[2]
Lighter Saber	256	2	1
Saber	256	3	3
Fire Saber	256	4	5



```
Algorithm 1: Saber.PKE.KeyGen()
$seed_{\boldsymbol{A}} \leftarrow \mathcal{U}(\{0,1\}^{256})$
$\boldsymbol{A} = \mathrm{gen}(\mathrm{seed}_{\boldsymbol{A}}) \in R_q^{l \times l}$
$r = \mathcal{U}(\{0,1\}^{256})$
$\boldsymbol{s} = \beta_\mu(R_q^{l \times 1}; r)$
$\boldsymbol{b} = ((\boldsymbol{A}^T \boldsymbol{s} + \boldsymbol{h}) \bmod q) \gg (\epsilon_q - \epsilon_p) \in R_p^{l \times 1}$
return $(pk := (seed_{\boldsymbol{A}}, \boldsymbol{b}), sk := (\boldsymbol{s}))$
```

```
Algorithm 2: Saber.PKE.Enc($pk = (seed_{\boldsymbol{A}}, \boldsymbol{b}), m \in R_2; r$)
$\boldsymbol{A} = \mathrm{gen}(\mathrm{seed}_{\boldsymbol{A}}) \in R_q^{l \times l}$
if $r$ is not specified then
    $r = \mathcal{U}(\{0,1\}^{256})$
$\boldsymbol{s}' = \beta_\mu(R_q^{l \times 1}; r)$
$\boldsymbol{b}' = ((\boldsymbol{A}\boldsymbol{s}' + \boldsymbol{h}) \bmod q) \gg (\epsilon_q - \epsilon_p) \in R_p^{l \times 1}$
$v' = \boldsymbol{b}^T (\boldsymbol{s}' \bmod p) \in R_p$
$c_m = (v' + h_1 - 2^{\epsilon_p - 1} m \bmod p) \gg (\epsilon_p - \epsilon_T) \in R_T$
return $c := (c_m, \boldsymbol{b}')$
```

```
Algorithm 3: Saber.PKE.Dec($sk = \boldsymbol{s}, c = (c_m, \boldsymbol{b}')$)
$v = \boldsymbol{b}'^T (\boldsymbol{s} \bmod p) \in R_p$
$m' = ((v - 2^{\epsilon_p - \epsilon_T} c_m + h_2) \bmod p) \gg (\epsilon_p - 1) \in R_2$
return $m'$
```

[2]:Jan-Pieter D'Anvers, Angshuman Karmakar, Sujoy Sinha Roy, and Frederik Vercauteren. SABER. Proposal to NIST PQC Standardization, Round2, 2019. https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/round-2-submissions

Major Arithmetic Operation

KeyGen.5: $\boldsymbol{b} = ((\boldsymbol{A}^T\boldsymbol{s} + \boldsymbol{h}) \bmod q) \gg (\epsilon_q - \epsilon_p) \in R_p^{l\times 1}$

Coeff. Involved: n*l*l, n*l

Enc.5: $\boldsymbol{b}' = ((\boldsymbol{A}\boldsymbol{s}' + \boldsymbol{h}) \bmod q) \gg (\epsilon_q - \epsilon_p) \in R_p^{l\times 1}$

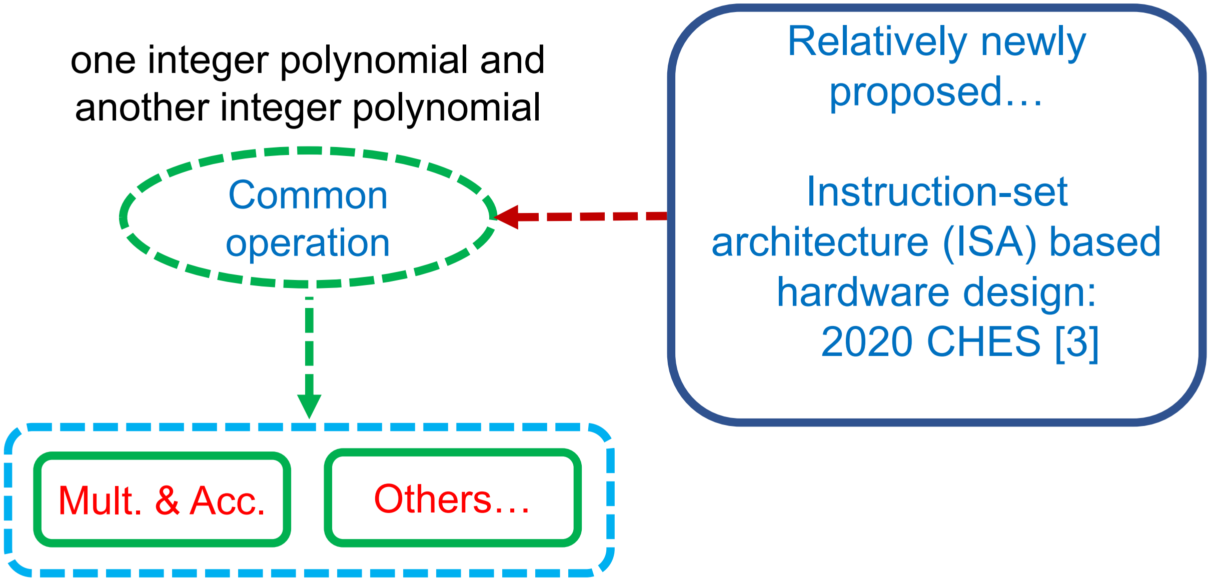
Coeff. Involved: n*l*l, n*l

Enc.6: $v' = \boldsymbol{b}^T(\boldsymbol{s}' \bmod p) \in R_p$

Coeff. Involved: n*l, n*l

Dec.1: $v = \boldsymbol{b}'^T(\boldsymbol{s} \bmod p) \in R_p$

Coeff. Involved: n*l, n*l



one integer polynomial and another integer polynomial

Common operation

Mult. & Acc.

Others…

Relatively newly proposed…

Instruction-set architecture (ISA) based hardware design: 2020 CHES [3]

[3]:Roy, Sujoy Sinha, and Andrea Basso. "High-speed Instruction-set Coprocessor for Lattice-based Key Encapsulation Mechanism: Saber in Hardware." IACR Transactions on Cryptographic Hardware and Embedded Systems (2020): 443-466.

Challenges and Goals

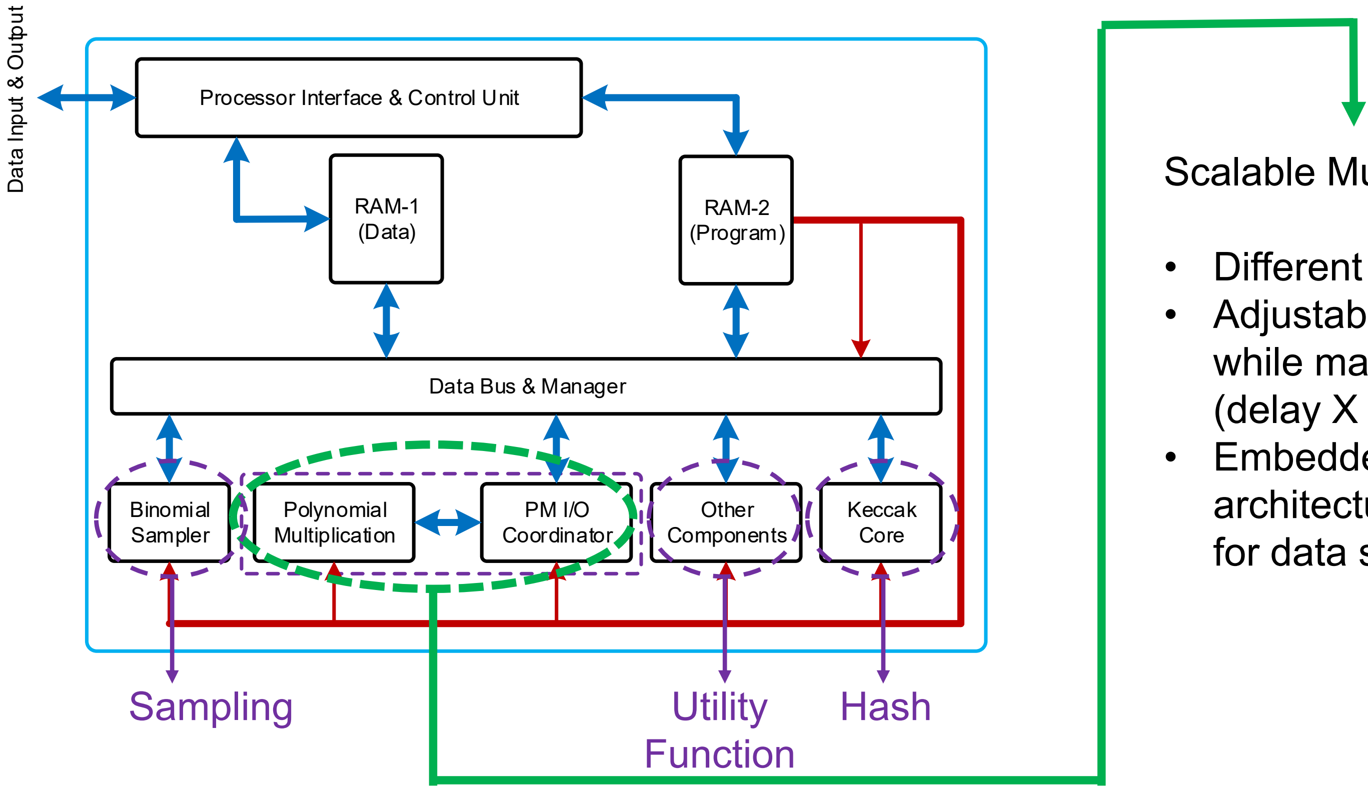
Low resource usage and high-performance hardware implementation of Saber KEM (PKE included) scheme

First topologically scalable structure (suitable for different applications) reported for Saber

Novel algorithm-hardware co-design driven process for the Saber PQC

THREE MAJOR GOALS

Proposed Method: Compact Coprocessor



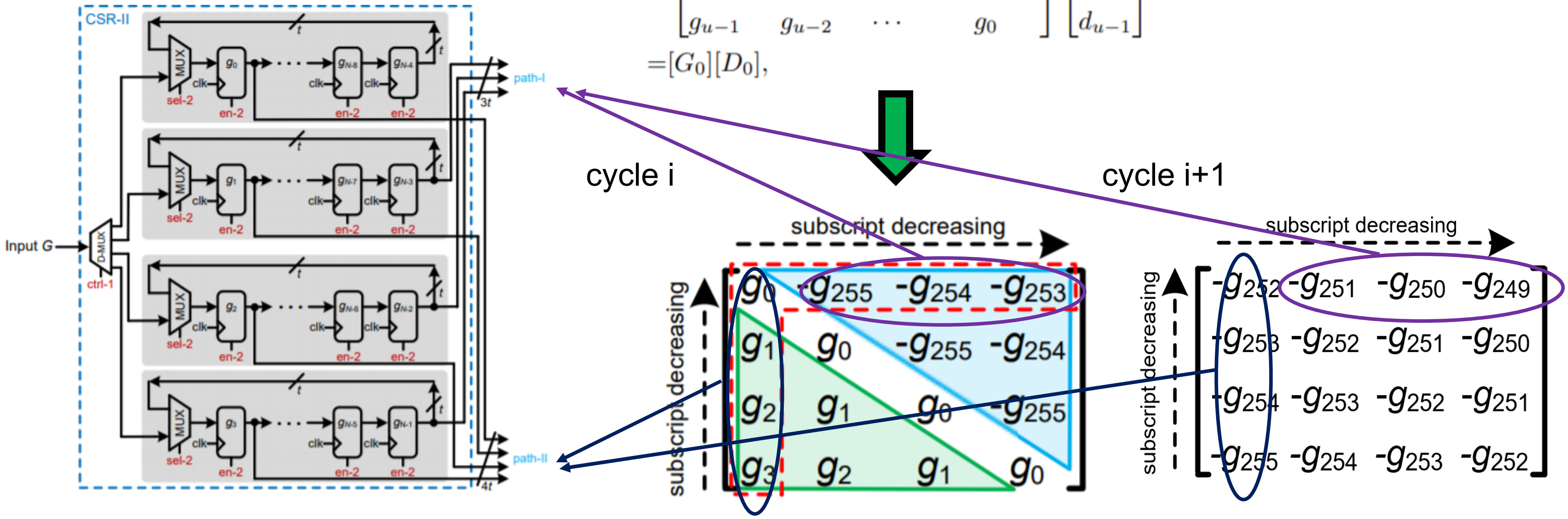
Scalable Multiplier Features

- Different length in rank of the quotient ring
- Adjustable ratio in terms of Speed/Usage while maintaining overall high performance (delay X area)
- Embedded into an ISA-based hardware architecture while maintaining low-latency for data store/fetch with a RAM

Proposed Method: Matrix Property - I

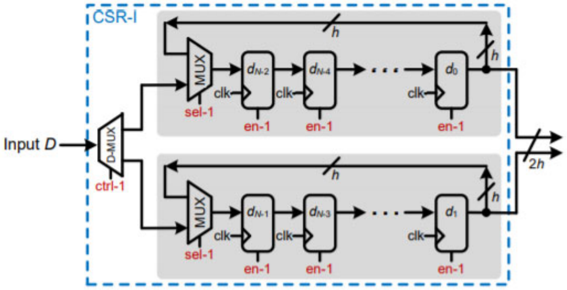
- Consider polynomial multiplication on quotient ring *T=GD mod (x^n+1), the matrix representation can be*

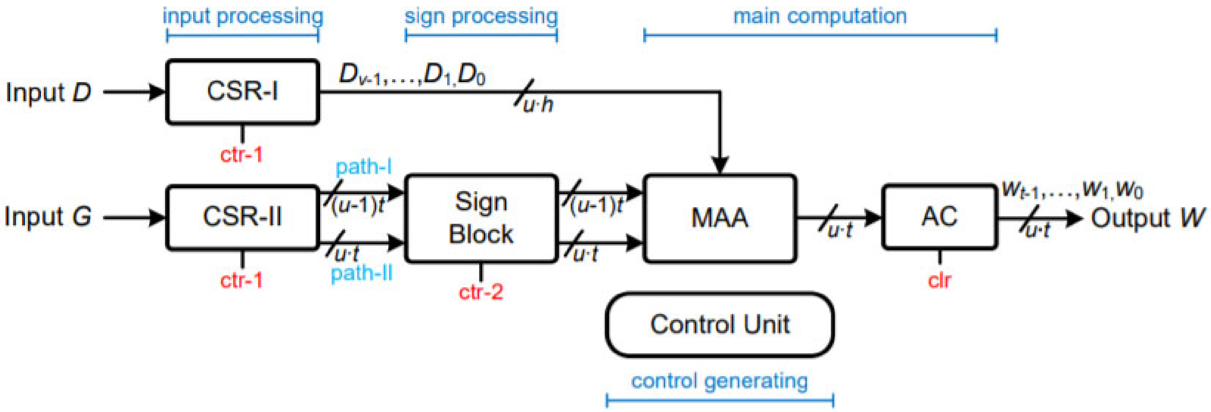
$$
\begin{aligned}
[T_0^{(0)}] &= \begin{bmatrix} g_0 & -g_{N-1} & \cdots & -g_{N-u+1} \\ g_1 & g_0 & \cdots & -g_{N-u+2} \\ \vdots & \vdots & \ddots & \vdots \\ g_{u-1} & g_{u-2} & \cdots & g_0 \end{bmatrix} \begin{bmatrix} d_0 \\ d_1 \\ \vdots \\ d_{u-1} \end{bmatrix} \\
&= [G_0][D_0],
\end{aligned}
$$

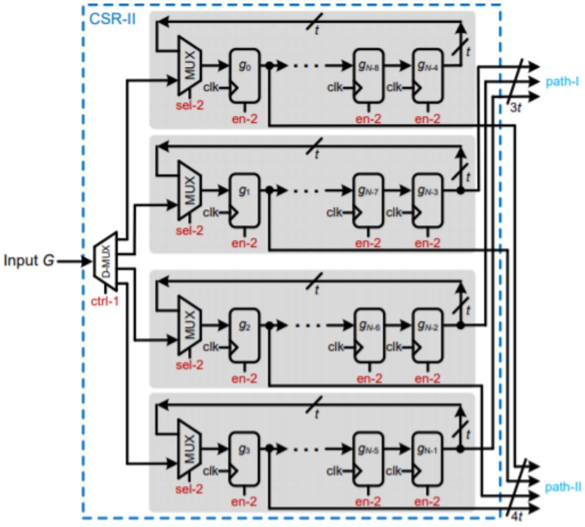


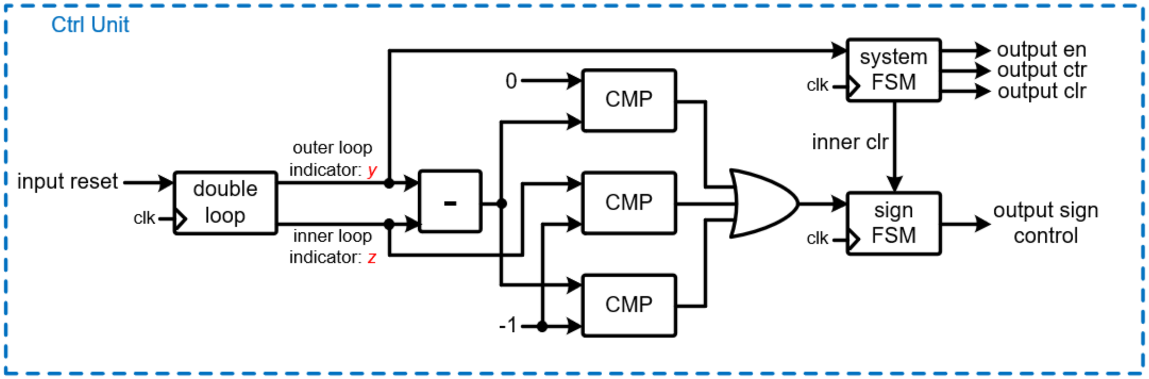
$$
\begin{bmatrix} g_0 & -g_{255} & -g_{254} & -g_{253} \\ g_1 & g_0 & -g_{255} & -g_{254} \\ g_2 & g_1 & g_0 & -g_{255} \\ g_3 & g_2 & g_1 & g_0 \end{bmatrix}
\qquad
\begin{bmatrix} -g_{252} & -g_{251} & -g_{250} & -g_{249} \\ -g_{253} & -g_{252} & -g_{251} & -g_{250} \\ -g_{254} & -g_{253} & -g_{252} & -g_{251} \\ -g_{255} & -g_{254} & -g_{253} & -g_{252} \end{bmatrix}
$$

Proposed Method: Implementation









Implementation Results

Design	Device	Scheme	Freq (MHz)	Time(KeyGen/Enc/Dec)µs	LUT	FF	DSP	BRAM
[3]	UltraScale+	Saber	150	18.4/26.9/33.6	24.9k	10.7k	0	3
				36.4/44.1/53.6				
				60.2/68.4/82.0				
[4]	Artix-7	Kyber	21-	-/14.3/20.9	11,864	10,348	15	8
				-/19.2/26.5	11,884	10,380		
				-/27.4/35.2	12,183	12,441		
[5]	Artix-7	Kyber	59	12,034/16,458/14,746	1,842	1,634	34	5
				-				
				37,339/44,390/41,169				
This work	UltraScale+	Saber	250	36.3/46.2/57.1	10.1k	7.7k	0	3
				48.9/63.2/78.5				
				61.5/80.2/100.0				

[3]:Roy, Sujoy Sinha, and Andrea Basso. "High-speed Instruction-set Coprocessor for Lattice-based Key Encapsulation Mechanism: Saber in Hardware." IACR Transactions on Cryptographic Hardware and Embedded Systems (2020): 443-466.

[4]:V. B. Dang, F. Farahmand, M. Andrzejczak, K. Mohajerani, D. T. Nguyen, and K. Gaj, "Implementation and benchmarking of round 2 candidates in the nist post-quantum cryptography standardization process using hardware and software/hardware co-design approaches," Cryptology ePrint Archive: Report 2020/795, 2020.

[5]: E. Alkim, H. Evkan, N. Lahr, R. Niederhagen, and R. Petri, "Isa extensions for finite field arithmetic," IACR Transactions on Cryptographic Hardware and Embedded Systems, pp. 219–242, 2020.

The proposed implementation achieved **32.8%, 28.3%, and 26.8%** overall performance enhancement compared with [3], the state-of-the-art design in case of *u*=8.

Detailed information about proposed implementation using Vivado is shown below under a target frequency of 250MHz when *u*=8.

Building Block	LUTs	FFs	CLBs	DSPs	BRAMs
Keccak Core	5655	2984	888	0	0
Sampler	229	88	77	0	2
Multiplier	2162	1656	448	0	0
I/O Coordinator	69	81	44	0	1
Others	1996	2890	483	0	2
Overall	10111	7699	1940	0	3
(% of overall FPGA device)	3.69	1.40	5.66	0	0.33

Conclusions

We presented the first resource constraint and topologically scalable design scheme that can be applied to all Saber PQC variants (namely Light Saber, Saber and Fire Saber)

The proposed hardware structure achieved 25%+ enhancement compared to the state-of-the-art design.

The proposed method involved algorithm and hardware improvement to the KEM Saber scheme's implementation.

Future Works

Novel complexity reduction strategy for the Saber KEM PQC

Novel secure implementation strategy for the Saber KEM PQC

New generation of hardware design methodology for PQC

Research Sponsors

NSF

NIST
National Institute of
Standards and Technology

THANK YOU!

- Contact: jiafeng.xie@villanova.edu (corresponding author)