Lumora: A family of permutation based wide-block ciphers for PQC zkSNARK applications

Susanta Samanta and Guang Gong

Department of Electrical and Computer Engineering
University of Waterloo
CANADA
https://uwaterloo.ca/scholar/ggong

Permutation-based Crypto 2025
May 4, 2025

Outline

- **Motivation** from practical MPC, zkSNARKs, and FHE requirements
- **Lumora**: a family of permutation based wide-block ciphers
- Concluding **remarks**

Motivation

Current applications, listed below, request minimized multiplication depth in their underline symmetric-key algorithms toward to be practical.

- **Multi-party** computation (MPC),
- **zero-knowledge** Succinct Noninteractive Argument of Knowledge (zkSNARK) proofs, and
- fully homomorphic **encryption** (FHE).

Motivation (cont.)

- In zkSNARK schemes for Rank-1 Constraint Satisfaction (R1CS), the prover/verifier's complexity only depends
  - on the **number** of multiplication gates in a fan-in two circuit,
  - the **size of the underlying finite field** is not so relevant or it can be easily satisfied that condition without increasing the complexity.
- Examples include **Stark (2018)**, Aurora (2019), Fractal (2019) Polaris (2022), Sparrow (2024), etc..
- zkSNARK-friendly ciphers aim to minimize multiplicative complexity (e.g., MiMC).
- According to the constraint system, the **inverse function** over a finite field $\mathbb{F}_{2^n}$ only counts as one constraint, however, it has **degree** $n-1$! This motivates our choice for nonlinear S-box operations.

MiMC [1]

- **MiMC-n/n.** A block cipher, defined over $\mathbb{F}_q = GF(q), q = 2^n$ or $q = p$ a prime.
- Let $x, k \in \mathbb{F}_q$. The round function of MiMC-n/n is as follows:

$$\begin{aligned} & f(x) = x^3, \ \text{more general } f(x) = x^d, \gcd(d, q-1) \\ & f_i(x) = f(x + k + t_i), t_i \in \mathbb{F}_q, i = 0, \cdots, r-1, t_0 = 0, \end{aligned} \tag{1}$$

The encryption function is iterated f_i r times.

[1] M. Albrecht+. MiMC: Efficient encryption and cryptographic hashing with minimal multiplicative complexity. Advances in Cryptology - ASIACRYPT 2016.

Two instantiations

HadesMiMC[2]

- AES like structure:

1. Add the subkey
2. An Sbox over $\mathbb{F}_p$
3. Mix affine layer: a $l \times l$ maximum distance separable (MDS) matrix over $\mathbb{F}_p$

- The ith round function is defined as

$$G_i(x) = MF_i(x), F_i(x) = F(x + k_i) \in \mathbb{F}_p^l, x, k_i \in \mathbb{F}_p^l, i = 0, 1, \cdots, r - 1,$$

where k_i is a round key generated by a key scheduling algorithm.

- Parameters: $m = \lfloor \log p \rfloor = 255$, $n = lm$-bit plaintext block and key block, $l = 3, 5$.

[2] L. Grassi+. On a generalization of substitution-permutation networks: The HADES design strategy. Advances in Cryptology - EUROCRYPT 2020.

Poseidon [3]

- an instantiation of HadesMiMC hash using the **sponge structure**.
- Tailored for **Groth16 zkSNARK** with BLS12-381, BN254, Ed25519 curves for trusted set-up.

[3] L.Grassi+. Poseidon: A new hash function for Zero-Knowledge proof systems. USENIX Security 21.

What other ciphers with large internal states?

- **Keccak-1600** [4]
  - **Large** 1600-bit internal state
  - Suited for **64-bit parallelism**
  - **Not clear** if it can be converted to a permutation over $\mathbb{F}_{2^{64}}$ with MiMC.
- **Snow V** [5]
  - Finite stata based structure with **1024-bit state**, 32-bit registers.
  - It uses **two LFSRs** with degree 16 over $\mathbb{F}_{2^{32}}$ with the **full AES-128** as a round function.
  - So, the **block size** for nonlinear permutation is actually to work on $\mathbb{F}_{2^8}$.
  - Thus, it is unknown whether this structure can be made MiMC.

---

[4] G. Bertoni+. The KECCAK reference. 2011 `https://keccak.team/files/Keccak-reference-3.0.pdf`.

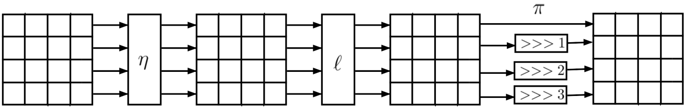
[5] P. Ekdahl+. A new SNOW stream cipher called SNOW-V. IACR Transactions on Symmetric Cryptology, 2019.

Lumora

- **Lumora** is a family of permutation based wide-block ciphers.
- It adopts the **AES-like** structures: the round function is composed of SubCell, MixColumns operation and the ShiftRows operation.
- Each instantiation follows a **unified structure**; only the **block size varies**, defined over the binary extension field $\mathbb{F}_{2^n}$, with $n \in \{16, 32, 64\}$
- We **denote** each cipher in the family as **Lumora**$(16n, n)$, where $16n$ represents the block size, and n indicates the size of the underlying finite field $\mathbb{F}_{2^n}$

Design Specification

- Each encryption round of a **Lumora** cipher is composed of three different transformations in the following order:
  - a nonlinear transformation η,
  - a linear transformation ℓ, and
  - a cell permutation π



Lumora (cont.)

- The cipher receives a $16n$-bit plaintext $P = b_0 b_1 b_2 \cdots b_{16n-2} b_{16n-1}$ as the cipher state I, where b_0 is the most significant bit.
- The cipher state can also be expressed as sixteen n-bit (here $n = 16, 32, 64$) cells as

$$I = \begin{bmatrix} s_0 & s_4 & s_8 & s_{12} \\ s_1 & s_5 & s_9 & s_{13} \\ s_2 & s_6 & s_{10} & s_{14} \\ s_3 & s_7 & s_{11} & s_{15} \end{bmatrix}, \ s_i \in \mathbb{F}_{2^n}$$

Design Specifications: The SubCell transformation η

- This is a nonlinear transformation in which an Sbox $S : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$ is applied to each cell of the cipher internal state

$$s_i \leftarrow S(s_i) \quad \text{for } i = 0, 1, \ldots, 15.$$

s_0	s_4	s_8	s_{12}
s_1	s_5	s_9	s_{13}
s_2	s_6	s_{10}	s_{14}
s_3	s_7	s_{11}	s_{15}

$$\xrightarrow{S}$$

$S(s_0)$	$S(s_4)$	$S(s_8)$	$S(s_{12})$
$S(s_1)$	$S(s_5)$	$S(s_9)$	$S(s_{13})$
$S(s_2)$	$S(s_6)$	$S(s_{10})$	$S(s_{14})$
$S(s_3)$	$S(s_7)$	$S(s_{11})$	$S(s_{15})$

- The Sbox S is defined as $S(X) = L(Y) + b$
  - $Y = X^{-1}$ is the multiplicative inverse of X in $\mathbb{F}_{2^n}$ (with $Y = 0$ when $X = 0$).
  - L is a linearized polynomial over the finite field $\mathbb{F}_{2^n}$, which is also a permutation.
  - b is a nonzero element in $\mathbb{F}_{2^n}$.

Unified structure of the linearized polynomial L

- Consider the block matrix

$$M_L = \begin{bmatrix} \mathbf{0}_m & \mathbf{0}_m & \mathbf{I_m} & \mathbf{I_m} \\ \mathbf{I_m} & \mathbf{0}_m & \mathbf{0}_m & \mathbf{0}_m \\ \mathbf{I_m} & \mathbf{I_m} & \mathbf{0}_m & \mathbf{0}_m \\ \mathbf{0}_m & \mathbf{0}_m & \mathbf{I_m} & \mathbf{0}_m \end{bmatrix} = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \otimes I_m$$

  where $\mathbf{0}_m$ denotes the $m \times m$ zero matrix, and $\mathbf{I_m}$ is the $m \times m$ identity matrix, with $m = \frac{n}{4}$

- That is, M_L is an $n \times n$ binary matrix.

The linearized polynomial L

The linearized polynomial L over $\mathbb{F}_{2^n}$ is defined as the linearized polynomial corresponding to the binary matrix M_L

Theorem[6] Given matrix M and a basis $\mathbf{e}$, the coefficient $\ell^{(t)}$ are given by

$$\ell^{(t)} = \sum_{i=1}^{n} \sum_{j=1}^{n} M_{ij} d[j]^{2^t} e[i],$$

where $\mathbf{d}$ is the dual basis of $\mathbf{e}$.

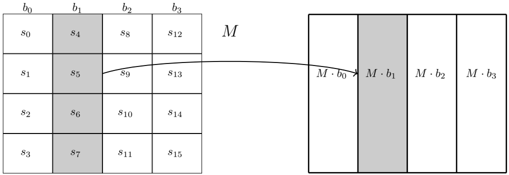
[6] Joan Daemen and Vincent Rijmen (2002). The Design of Rijndael: AES - The Advanced Encryption Standard. Information Security and Cryptography. Springer.

The linearized polynomial L

- **Note:** In the **Lumora** family, the underlying finite field is $\mathbb{F}_{2^n}$, where $n \in \{16, 32, 64\}$.
- For these values of n, all coefficients of the linearized polynomial L are nonzero.
- Moreover, none of the coefficients of L are equal to α, the root of the primitive polynomial that defines the field $\mathbb{F}_{2^n}$.
- We choose $b = \alpha$.

Design Specifications: The MixColumn transformation ℓ

- This is a linear operation that operates separately on each of the four columns of the state
  - It uses a 4×4 MDS matrix M. We have
$$(s_i, s_{i+1}, s_{i+2}, s_{i+3})^t \leftarrow M \cdot (s_i, s_{i+1}, s_{i+2}, s_{i+3})^t$$
    for $i = 0, 4, 8, 12$.



Unified structure of the matrix M

- The matrix M is given as the product of the following 4 sparse matrices M_1, M_2, M_3 and M_4 of order 4 i.e. $M = M_1 M_2 M_3 M_4$, where

$$M_1 = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 0 & 0 & 1 & \alpha \\ 1 & 0 & 0 & 0 \\ \alpha^{-1} & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, M_3 = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ \alpha^{-1} & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad M_4 = M_1,$$

  where α is a nonzero element in the field $\mathbb{F}_{2^n}$

- The matrix

$$M = \begin{bmatrix} \alpha^{-1}+1 & \alpha^{-1} & 1 & \alpha^{-1}+1 \\ \alpha+1 & \alpha & \alpha^{-1} & \alpha^{-1} \\ \alpha & \alpha+1 & \alpha^{-1}+1 & \alpha^{-1} \\ \alpha^{-1} & \alpha^{-1} & \alpha^{-1}+1 & 1 \end{bmatrix}$$

- The **irreducible factors** appearing in the minors of M form the set

$$\{\alpha, \alpha+1, \alpha^2+\alpha+1, \alpha^3+\alpha+1, \alpha^3+\alpha^2+1\}$$

- Hence, if α is chosen as a root of a primitive polynomial that defines the field $\mathbb{F}_{2^n}$, where $n \in \{16,\ 32,\ 64\}$, then M is an **MDS matrix**.

The MixColumns in **Lumora**$(16n, n)$

- $$M_1 = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 0 & 0 & 1 & \alpha \\ 1 & 0 & 0 & 0 \\ \alpha^{-1} & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, M_3 = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ \alpha^{-1} & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad M_4 = M_1,$$

  where α is a root of the primitive polynomial that defines the underlying field $\mathbb{F}_{2^n}$ of *Lumora*$(16n, n)$

- In the MixColumns operation, the matrix multiplication is performed sequentially; that is,

  $$M \cdot \hat{b} = M_1 \cdot \left(M_2 \cdot \left(M_3 \cdot \left(M_4 \cdot \hat{b}\right)\right)\right).$$

  - Thus, we need only three multiplication over the finite field

Design Specifications: The ShiftRows operation π

- It rotates row i of the array state i cell positions to the right for $i = 0, 1, 2, 3$. More specifically, we have $s_i \leftarrow s_{(13 \cdot i \bmod 16)}$ for $i = 0, 1, \ldots, 15$

s_0	s_4	s_8	s_{12}
s_1	s_5	s_9	s_{13}
s_2	s_6	s_{10}	s_{14}
s_3	s_7	s_{11}	s_{15}

$\xrightarrow{\pi}$

s_0	s_4	s_8	s_{12}
s_{13}	s_1	s_5	s_9
s_{10}	s_{14}	s_2	s_6
s_7	s_{11}	s_{15}	s_3

Word Operation of **Lumora**$(16n, n)$

- Note that for the Sbox S, we have

$$S(y) = T \circ \sigma(y), \text{ where } \sigma(y) = y^{-1} \text{ and } T(y) = L(y) + b$$

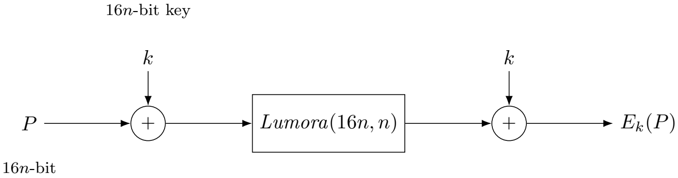
- The round function of **Lumora**$(16n, n)$: $\pi \circ \ell \circ \eta$

- For $\hat{y} = (y_i, y_{i+1}, y_{i+2}, y_{i+3}) \in (\mathbb{F}_{2^n})^4$, the word operation W is defined as

$$W(\hat{y}) = \begin{bmatrix} \alpha^{-1}+1 & \alpha^{-1} & 1 & \alpha^{-1}+1 \\ \alpha+1 & \alpha & \alpha^{-1} & \alpha^{-1} \\ \alpha & \alpha+1 & \alpha^{-1}+1 & \alpha^{-1} \\ \alpha^{-1} & \alpha^{-1} & \alpha^{-1}+1 & 1 \end{bmatrix} \begin{bmatrix} T(y_i^{-1}) \\ T(y_{i+1}^{-1}) \\ T(y_{i+2}^{-1}) \\ T(y_{i+3}^{-1}) \end{bmatrix}$$

$$\text{i.e., } W(\hat{y}) = M_1 \cdot (M_2 \cdot (M_3 \cdot (M_4 \cdot \hat{b}))), \quad \text{where} \quad \hat{b} = \begin{bmatrix} T(y_i^{-1}) \\ T(y_{i+1}^{-1}) \\ T(y_{i+2}^{-1}) \\ T(y_{i+3}^{-1}) \end{bmatrix}$$

The Even–Mansour Construction



Lumora(256, 16)

- The underlying finite field $\mathbb{F}_{2^{16}}$ is defined by the polynomial $t(x) = x^{16} + x^{12} + x^3 + x + 1$, which is a primitive polynomial over $\mathbb{F}_2$

- The matrix $M_L = \begin{bmatrix} \mathbf{0}_4 & \mathbf{0}_4 & \mathbf{I}_4 & \mathbf{I}_4 \\ \mathbf{I}_4 & \mathbf{0}_4 & \mathbf{0}_4 & \mathbf{0}_4 \\ \mathbf{I}_4 & \mathbf{I}_4 & \mathbf{0}_4 & \mathbf{0}_4 \\ \mathbf{0}_4 & \mathbf{0}_4 & \mathbf{I}_4 & \mathbf{0}_4 \end{bmatrix}$ is given by

$$M_L = \begin{bmatrix}
0&0&0&0&0&0&0&0&1&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1\\
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0
\end{bmatrix}$$

- Thus,

$$L(x) = 110x + 481dx^2 + 81e3x^4 + 5b63x^8 + a75x^{16} + b3b4x^{32} + 7305x^{64} + 6ab7x^{128} + b846x^{256} + 665cx^{512} + 9e0cx^{1024} + 8df6x^{2048} + d2b8x^{4096} + 4754x^{8192} + 4c6bx^{16384} + 2689x^{32768}$$

Lumora(256, 16)

- The **differential uniformity** of the Sbox is 4, as in[7], implying maximum differential probability of the Sbox is $\frac{4}{2^{16}} = 2^{-14}$
- The **maximum** absolute correlation of the Sbox is 2^{-7}
- The **Wide-Trail Strategy**[8] guarantees that the minimum number of active S-boxes in any four-round differential (or linear) trail of **Lumora**$(16n, n)$ is lower bounded by 25
  - the maximum differential probability (or absolute correlation) of any four-round differential (or linear) trail of **Lumora**(256, 16) is upper bounded by 2^{-350} (or 2^{-175}), respectively
- Total number of rounds in **Lumora**(256, 16): 10

[7] Kaisa Nyberg (1994). "Differentially uniform mappings for cryptography". In: *Advances in Cryptology — EUROCRYPT '93*, pp. 55–64.

[8] Joan Daemen (1995). *Cipher and hash function design, strategies based on linear and differential cryptanalysis, PhD Thesis*. http://jda.noekeon.org/. K.U.Leuven.

Lumora$(512, 32)$ and **Lumora**$(1024, 64)$

Lumora$(512, 32)$

- The primitive polynomial: $x^{32} + x^{22} + x^2 + x^1 + 1$
- The Sbox: $S = T \circ \sigma$ and $T(y) = L(y) + \alpha$, where L is corresponding to the matrix

$$\begin{bmatrix} \mathbf{0}_8 & \mathbf{0}_8 & \mathbf{I_8} & \mathbf{I_8} \\ \mathbf{I_8} & \mathbf{0}_8 & \mathbf{0}_8 & \mathbf{0}_8 \\ \mathbf{I_8} & \mathbf{I_8} & \mathbf{0}_8 & \mathbf{0}_8 \\ \mathbf{0}_8 & \mathbf{0}_8 & \mathbf{I_8} & \mathbf{0}_8 \end{bmatrix}$$

- Total number of rounds: 8

Lumora$(1024, 64)$

- The primitive polynomial: $x^{64} + x^4 + x^3 + x + 1$
- The Sbox: $S = T \circ \sigma$ and $T(y) = L(y) + \alpha$, where L is corresponding to the matrix

$$\begin{bmatrix} \mathbf{0}_{16} & \mathbf{0}_{16} & \mathbf{I_{16}} & \mathbf{I_{16}} \\ \mathbf{I_{16}} & \mathbf{0}_{16} & \mathbf{0}_{16} & \mathbf{0}_{16} \\ \mathbf{I_{16}} & \mathbf{I_{16}} & \mathbf{0}_{16} & \mathbf{0}_{16} \\ \mathbf{0}_{16} & \mathbf{0}_{16} & \mathbf{I_{16}} & \mathbf{0}_{16} \end{bmatrix}$$

- Total number of rounds: 6

FAEST Signature Algorithm

- **FAEST** is one of NIST 14 2nd round additional signature candidates.
- It uses **AES** or AES in EM mode as the one-way function circuits and vector oblivious linear evaluation (VOLE) based, named as **VOLE-in-the-head** for the ZKP.
- Core idea:
  - **Secret Key**: AES encryption key k
  - **Public Key**: (m, c), where $c = \text{AES}_k(m)$
  - Signature: the proof in non-interactive zero-knowledge proof that signer knows k such that $\text{AES}_k(m) = c$.

FAEST Signature Algorithm (cont.)

- FAEST comes in several variants, offering trade-offs between security, speed, and signature size.
- **Security** Parameter
  - Defines the target security level (aligned with AES-128, AES-192, or AES-256)
  - Higher levels $\rightarrow$ stronger security, but with larger proofs and slower performance
- **Even–Mansour Variant**
  - Treats the block cipher as a public permutation: Key is public, input is secret.
  - Simplifies zero-knowledge proofs by avoiding key schedule simulation.
  - For 192 and 256-bit security, it uses Rijndael (larger block size).
- ShiftRows, MixColumns, AddRoundKey in AES or Rijndael: All are linear over $\mathbb{F}_2$
- Sbox in AES or Rijndael: In the zero-knowledge proof scheme, one inverse only counts as one **constraint** in R1CS relation, i.e., $x \cdot y = 1 \iff y = x^{-1}$.

FAEST Signature Algorithm (cont.)

Variant	Block Size	Key Size	Enc. Rounds	# Constraints in Enc.	# Constraints in Key Gen	Total Constraints
FAEST-128	128 bits	128 bits	10	$10 \times 16 =$ **160**	**40**	**200**
FAEST-192	128 bits	192 bits	12	$12 \times 16 =$ **192**	**32**	**224**
FAEST-256	128 bits	256 bits	14	$14 \times 16 =$ **224**	**52**	**276**
FAEST-EM-128	128 bits	Public	10	$10 \times 16 =$ **160**	**0** (key is public)	**160**
FAEST-EM-192	192 bits (Rijndael)	Public	12	$12 \times 24 =$ **288**	**0**	**288**
FAEST-EM-256	256 bits (Rijndael)	Public	14	$14 \times 32 =$ **448**	**0**	**448**

- **Note: Lumora**$(256, 16)$ has only **160** constraints in total!

Work in Progress

Cryptanalysis and implementation of Lumora

- Additional cryptanalysis on **Lumora**$(16n, n)$:
  - Algebraic attacks (**degree** progressing, $\cdots$,)
  - Integral attacks
  - Invariant **subspaces**
  - $\vdots$
- **Implementation** considerations and challenges (hardware-software co-design, tower field computation vs FFT, $\cdots$)

Applications of Lumora

- **Lumora** based FAEST style PQC DSA and performance comparisons
- Exploring other potential applications of **Lumora**$(16n, n)$ by embedding **Lumora** into zkSNARK schemes (e.g., Polaris/Aurora based PQC DSA)

Conclusion

- Each **instantiation** of **Lumora**$(16n, n)$ follows a unified .
  - only the block size varies, defined over the binary extension field $\mathbb{F}_{2^n}$, with $n \in \{16, 32, 64\}$
- **Lumora**$(16n, n)$ without key can be directly used in a sponge mode.
- We can also put the permutation in **Feistel/NLFSR** structure with two registers and each with $16n$ bits.
  - In this structure, for example, $n = 32$, we have a 1024-bit internal state, but the number of the rounds should be double in this case.
- **Lumora**$(16n, n)$ has MiMC by design.
- Choice of the inverse function over $\mathbb{F}_{2^n}$ is motivated by determined by R1CS (Rank 1 constraint system) relation.
- **Selecting** $n = 16, 32, 64$ allows efficient implementation via **tower** fields.

Thanks! Questions?