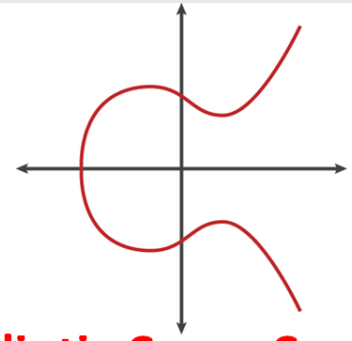
Sujoy Sinha Roy

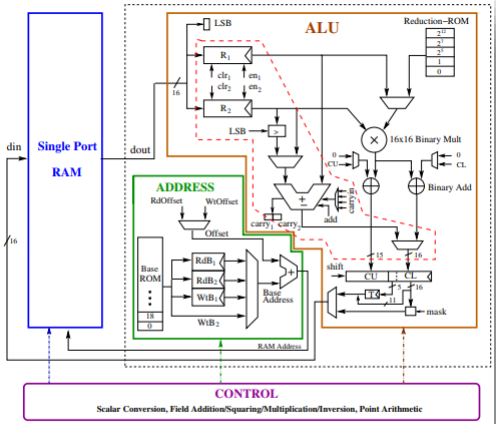
PhD, 2012-2017
KU Leuven

Post Doc., 2017-2018
KU Leuven

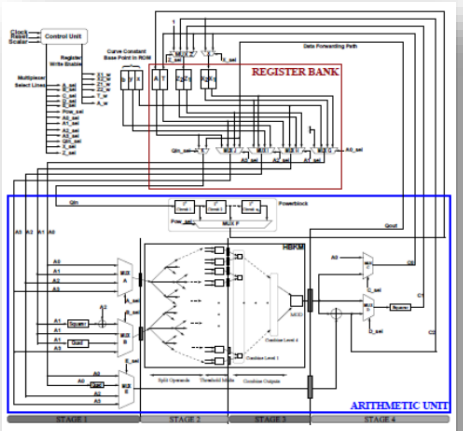
Lecturer, 2018 Sept -
University of Birmingham



Elliptic Curve Crypto



Lightweight ECC

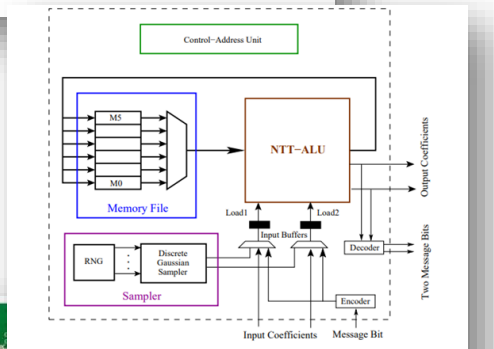


High-Speed ECC

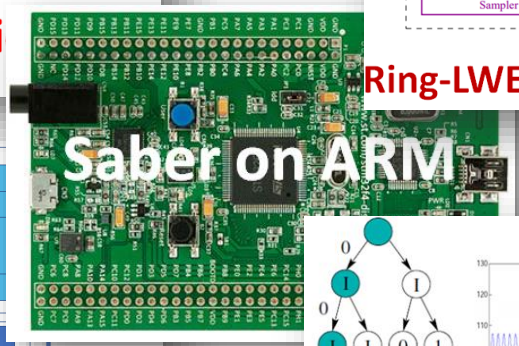
$$\begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix} \cdot \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \\ \vdots \\ e_m \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \\ \vdots \\ b_m \end{pmatrix}$$

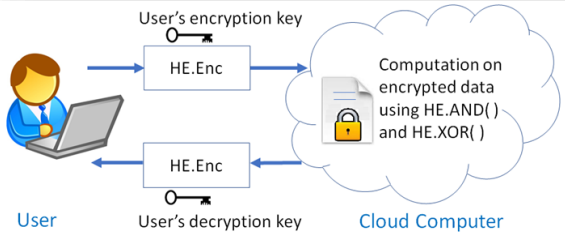
Lattice-based Cryptography

Saber: Module-LWR based key exchange, CPA-secure encryption and CCA-secure KEM (NIST Submissi

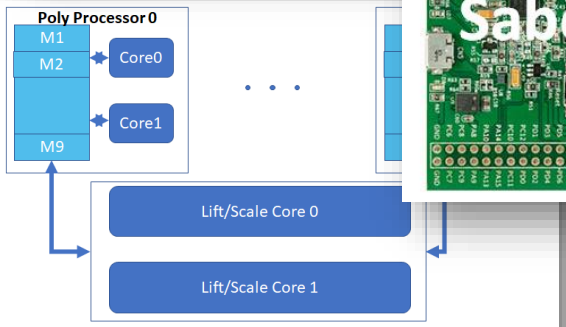


Ring-LWE Cryptoprocessor

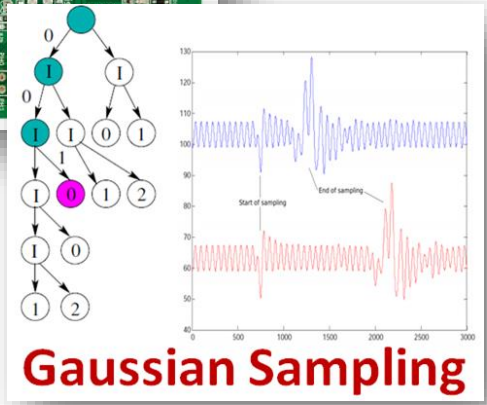




Homomorphic Encryption



Ring-LWE Homomorphic Processor



Gaussian Sampling

Ideal Lattice-based Diffie-Hellman key-exchange

Public: polynomial a

s ← error( )

e ← error( )

b=a·s + e

s’ ← error( )

e’ ← error( )

b’=a·s’ + e’

c=b’·s = a·s·s’ + e’·s

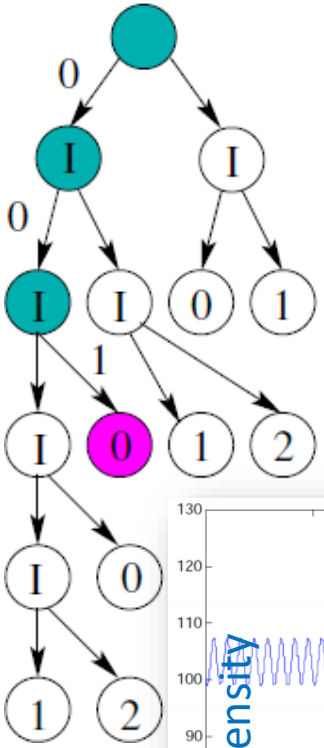
c’=b·s’ = a·s·s’ + e·s’

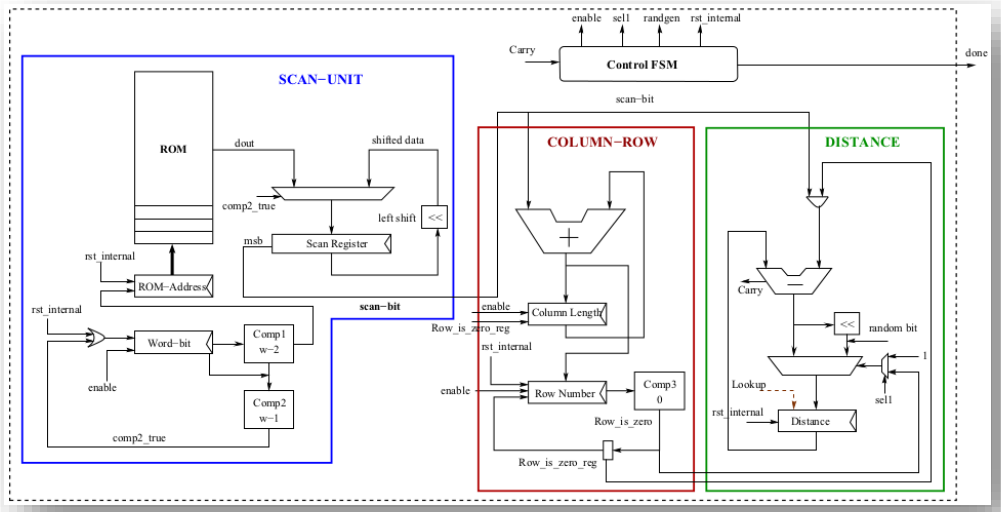
Shared poly with noise

Difference = | e’·s - e·s’ |

Gaussian Sampling

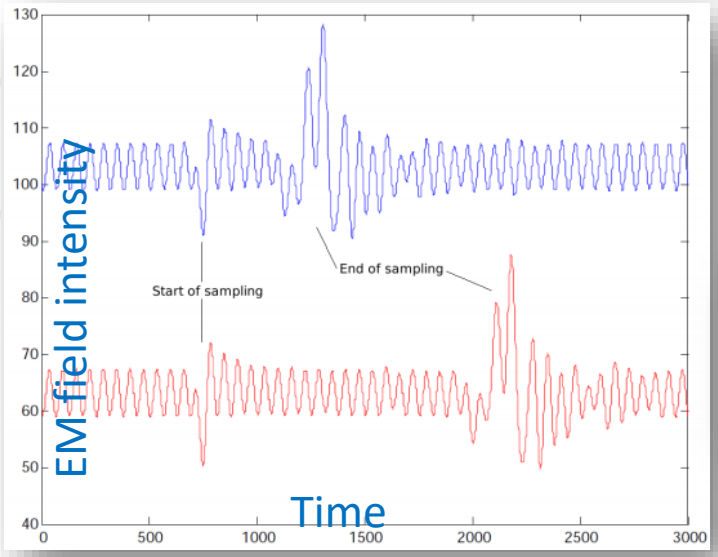
Knuth-Yao Random walk

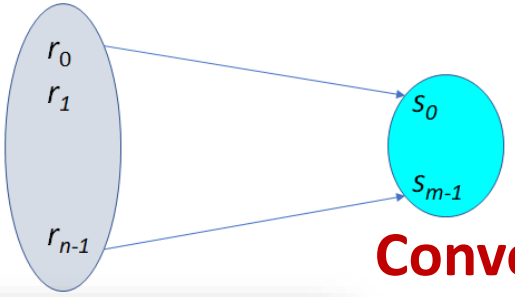




Area: 44 slices
Cycles (avg.) 1.5

Tiny HW architecture





$$s_0 = f^0(r_0, r_1, \ldots, r_{n-1})$$
$$\ldots$$
$$s_{m-1} = f^{m-1}(r_0, r_1, \ldots, r_{n-1})$$

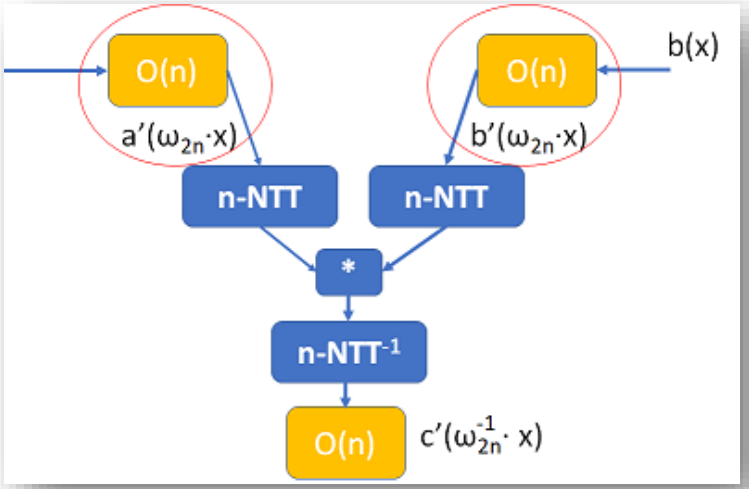
Convolution + Boolean Decomp + Bitslicing

Constant-time SW

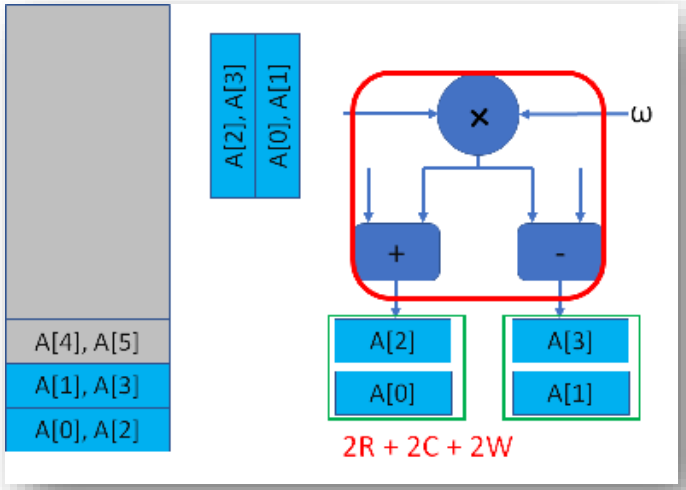
Polynomial Multiplication using NTT

Before O(n) + O(n) + O(n log n) + O(n)

Eliminated precomputation



Reduced memory access overhead



Now ~~O(n)~~ + ~~O(n)~~ + ½O(n log n) + O(n)

High-Speed lattice-based Cryptoprocessor, CHES 2014

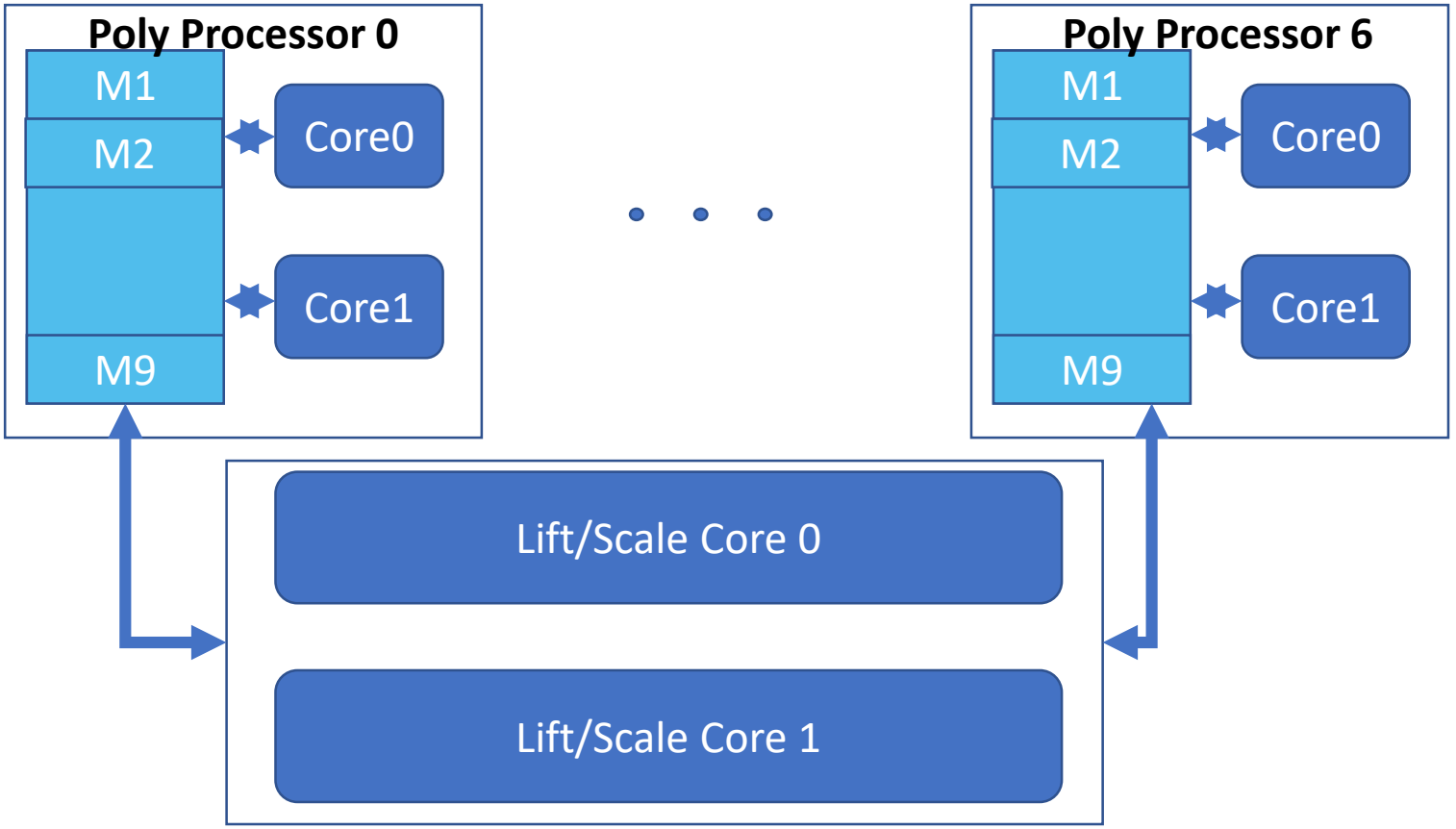
First masking scheme for lattice-based crypto, CHES 2015

Saber: Module-LWR based key exchange, CPA-secure encryption and CCA-secure KEM

LWR problem $b = \left\lfloor \frac{p}{q} A s \right\rceil$ where $p < q$

Efficient, Flexible and Secure

Accelerator for homomorphic evaluation: privacy-preserving cloud computing



400 homomorphic multiplications / s

Faster than Tesla K80 GPU

To appear in High Performance Computing Architecture (HPCA) 2018

Thank you