FUNCTIONAL ENCRYPTION ON FPGAs:
MULTI-CORE ARCHITECTURE FOR INNER-PRODUCT COMPUTATION

Milad Bahadori and Kimmo Järvinen

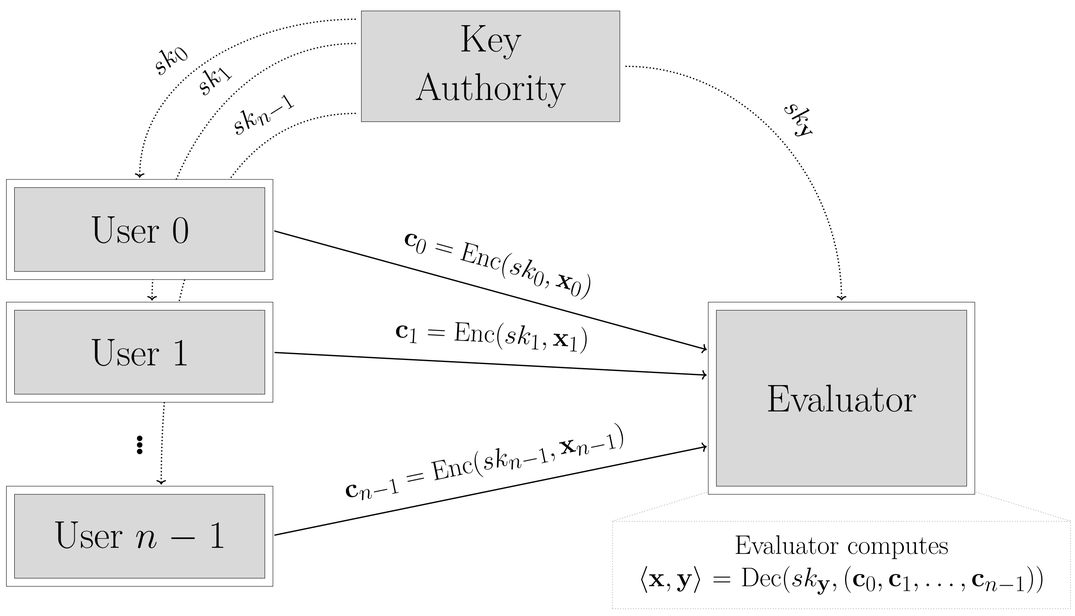
Department of Computer Science, University of Helsinki

FUNCTIONAL ENCRYPTION

- **Traditional encryption is all-or-nothing:** Anyone who has the secret key sk obtains the entire plaintext x from the ciphertext $c = \mathrm{Enc}(x)$ and the others get nothing at all.
- **Functional Encryption (FE) provides more fine-grained control:** It is possible to derive a decryption key sk_f that allows to compute $f(x)$ from c without leaking anything else about x.
- **Multi-Input FE (MIFE)** allows n users to independently encrypt input vectors $\mathbf{x}_i = (x_{i,0}, \ldots, x_{i,m-1})$ so that an evaluator can decrypt $f(\mathbf{x}_0, \ldots, \mathbf{x}_{n-1})$.
- **Efficient FE schemes exist only for limited functionalities:** In this work, we focus on **MIFE for inner-products** that supports $sk_{\mathbf{y}}$ for specific $\mathbf{y} = (\mathbf{y}_0, \ldots, \mathbf{y}_{n-1})$ with $\mathbf{y}_i = (y_{i,0}, \ldots, y_{i,m-1})$ that allows computing

$$f_{\mathbf{y}}(\mathbf{x}_0, \ldots, \mathbf{x}_{n-1}) = \langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} x_{i,j} \cdot y_{i,j}$$

from the ciphertexts $\mathbf{c}_i = \mathrm{Enc}(sk_i, \mathbf{x}_i)$ (see the figure below).

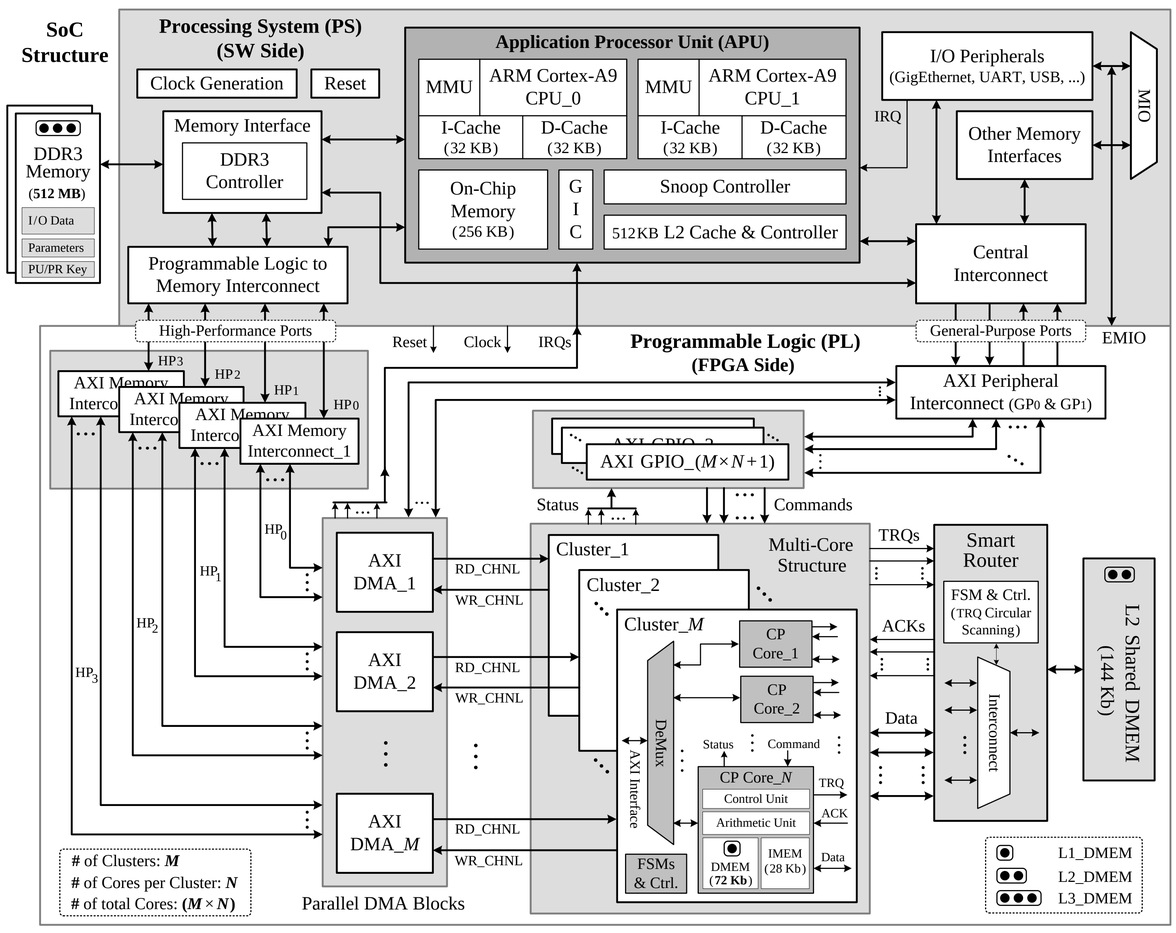


- **MIFE schemes are computationally demanding:** We focus on the MIFE scheme from **[Abd18]** instantiated with the FE scheme for inner-products from **[Agr16]**.
  - Based on additively homomorphic Paillier encryption.
  - Does not require cryptographic pairings.
  - Main operations are exponentiations in $\mathbb{Z}_{N^2}$ where N is an RSA-like modulus (e.g., of size $\kappa = 2048$ bits).
  - Both encryption and decryption are heavy; especially, decryption (inner-product computation), which is computed by the evaluator alone, gets expensive if n is large.

  ⇒ **There is a need for hardware support**

ARCHITECTURE

- **HW/SW codesign:** The architecture is designed mainly for Xilinx Zynq SoCs that combine FPGA resources with ARM Cortex-A9 cores (our prototype uses Avnet ZedBoard)
- **Multi-core design:** The architecture includes parallel cores in order to exploit the inherent parallelism in MIFE encryptions and decryptions and three-level memory structure for efficient inter-operation between the cores and SW
- **Cores are optimized for large integer modular arithmetic:** Each core uses Montgomery modular arithmetic optimized for hardwired DSP multipliers of the FPGA.



RESULTS

- **Architecture compiled for Xilinx Zynq-7020 SoC:** 12 cores ($M = 6, N = 2$) fit into the FPGA and measurements from ZedBoard with $\kappa = 2048$ give the timings below.

OPERATION		LATENCY (clocks) FPGA@122MHz	LATENCY (clocks) ARM@667MHz	TIME (ms)
Enc	Small ($m = 16$)	43863516	270504	360
	Medium ($m = 32$)	65795274	399126	540
	Large ($m = 64$)	131590548	784992	1080
Dec	Small ($n = 4, m = 16$)	45298890	77269998	487
	Medium ($n = 16, m = 32$)	93706204	310868766	1234
	Large ($n = 64, m = 64$)	299424756	1249220622	4328

- **Future work:** (a) More expressive functions and additional features (e.g., function hiding) by implementing support for cryptographic pairings; (b) Use of secure elements for added security and efficiency.

REFERENCES

[**Abd18**] M. Abdalla et al.: "Multi-input functional encryption for inner products: Function-hiding realizations and constructions without pairings." In CRYPTO 2018.

[**Agr16**] S. Agrawal et al.: "Fully secure functional encryption for inner products, from standard assumptions." In CRYPTO 2016.

Contact: {milad.bahadori, kimmo.u.jarvinen}@helsinki.fi

FENTEC