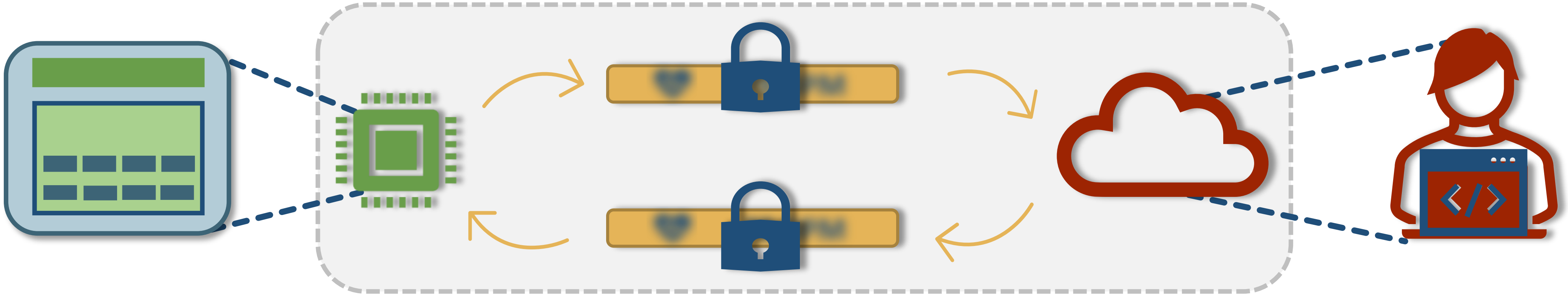
Client-Optimized Algorithms & Acceleration for Encrypted Compute Offloading

ASPLOS '22 | Lausanne, Switzerland | March 3, 2022

McKenzie van der Hagen – mckenziv@andrew.cmu.edu
Brandon Lucia – blucia@andrew.cmu.edu

Electrical & Computer ENGINEERING

Client-aided HE for Opaque Compute Offloading



- CHOCO enables privacy-preserving computation for **resource-constrained devices**
- CHOCO utilizes **Homomorphic Encryption (HE)** and Client-Aided Encrypted Computing
- CHOCO introduces client-optimized **encrypted algorithms & hardware acceleration**
- CHOCO makes client responsibility **competitive with local compute**
- CHOCO benefits **generalize to diverse applications**

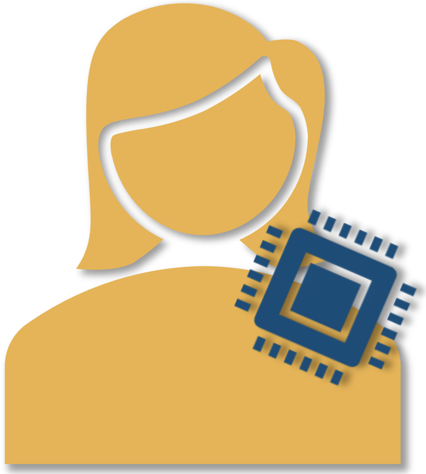
Outline

- Introduction
  - Resource-Constrained Devices & Privacy-Preserving Offload
- Encrypted Computing Background
  - Fully Homomorphic Encryption & Client-Aided Encrypted Computing
- CHOCO System Implementation
  - Algorithm Optimizations & Hardware Acceleration
- Results & Conclusions
  - Active Client Computation, Communication & Applications

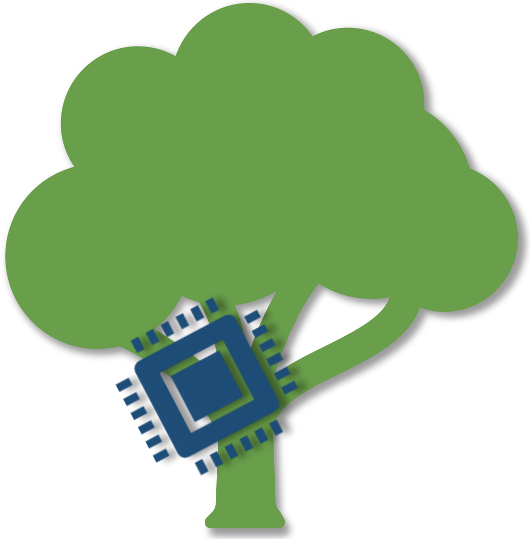
Outline

- **Introduction**
  - **Resource-Constrained Devices & Privacy-Preserving Offload**
- Encrypted Computing Background
- CHOCO System Implementation
- Results & Conclusions

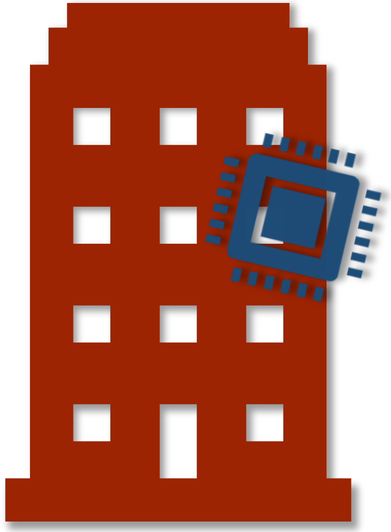
Resource-Constrained Devices are Everywhere



Health Monitoring

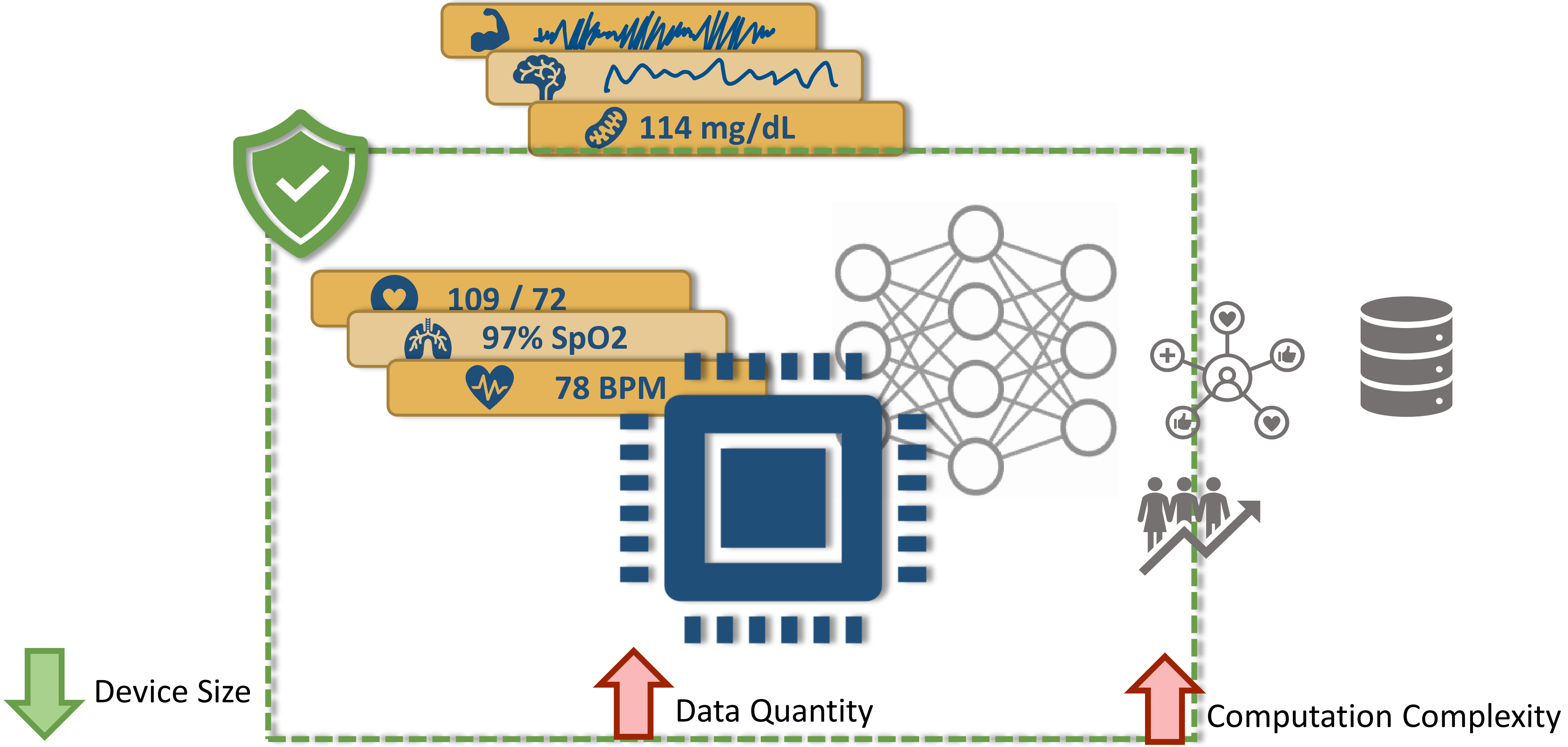


Wildlife Monitoring

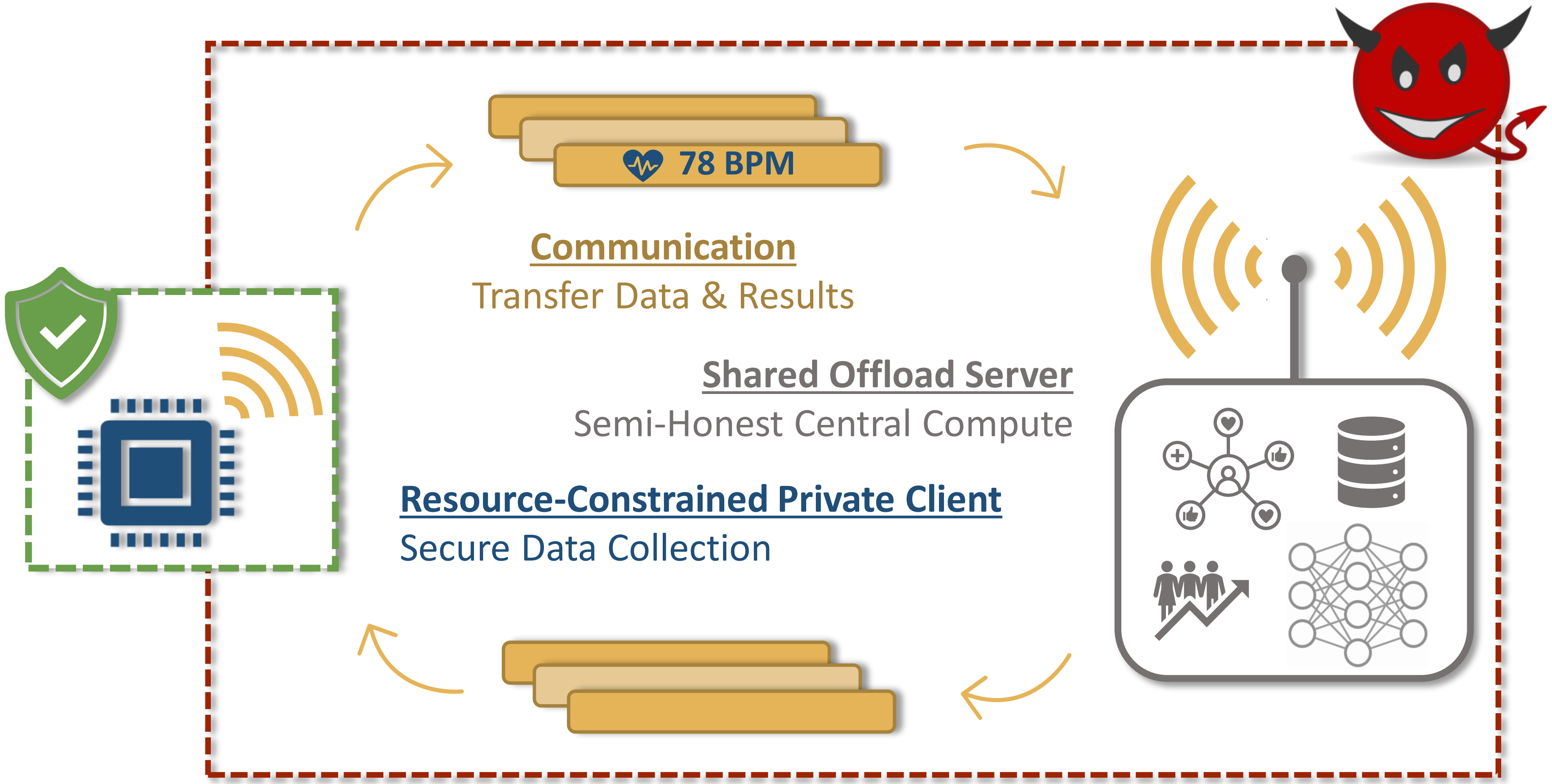


Infrastructure Monitoring

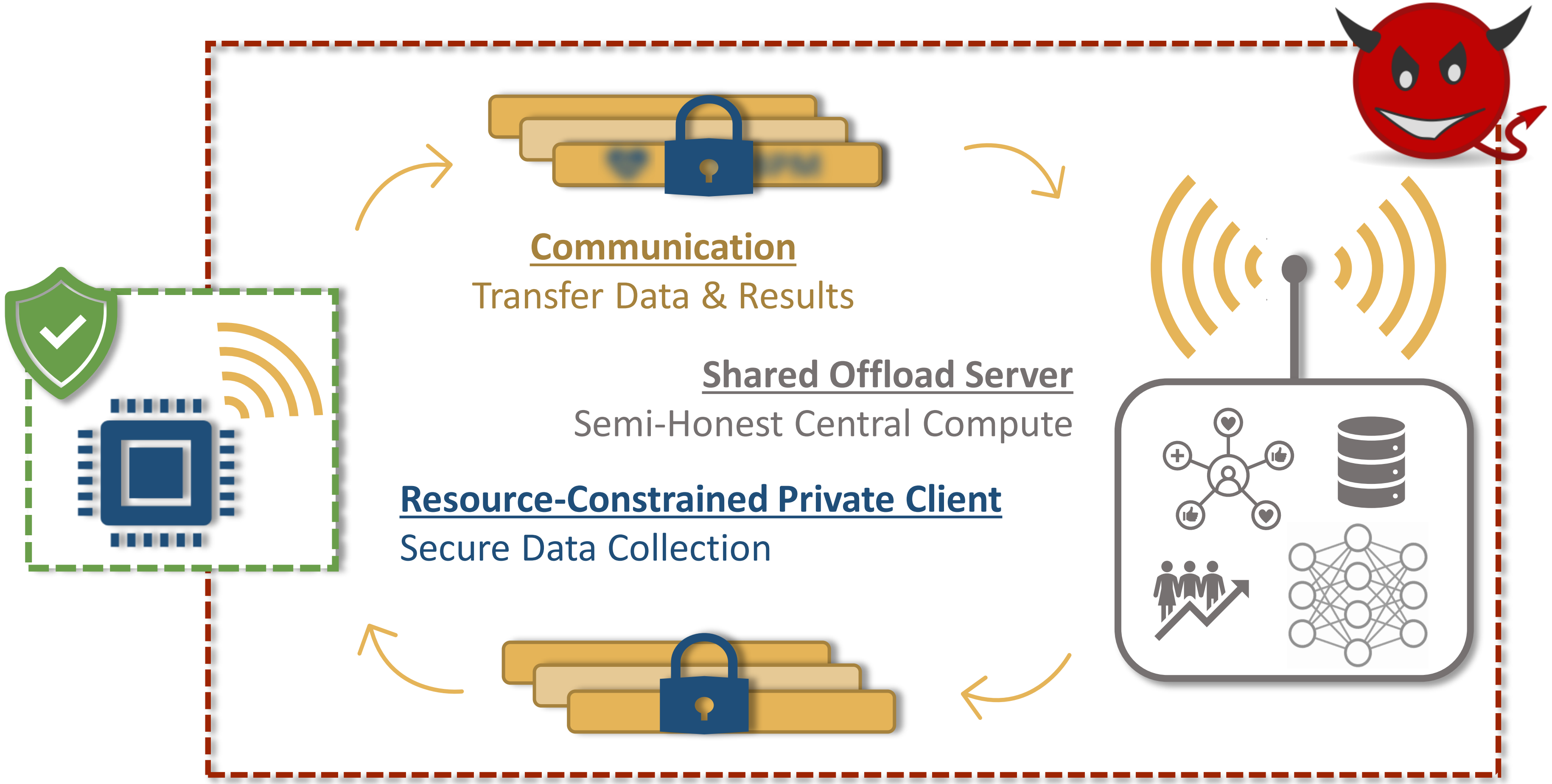
Computational Demands Exhaust Sensor Resources



Privacy-Preserving Computation Offload



Privacy-Preserving Computation Offload



FHE: [Fully] Homomorphic Encryption

Outline

- **Introduction**
- **Resource-Constrained Devices & Privacy-Preserving Offload**

- Encrypted Computing Background

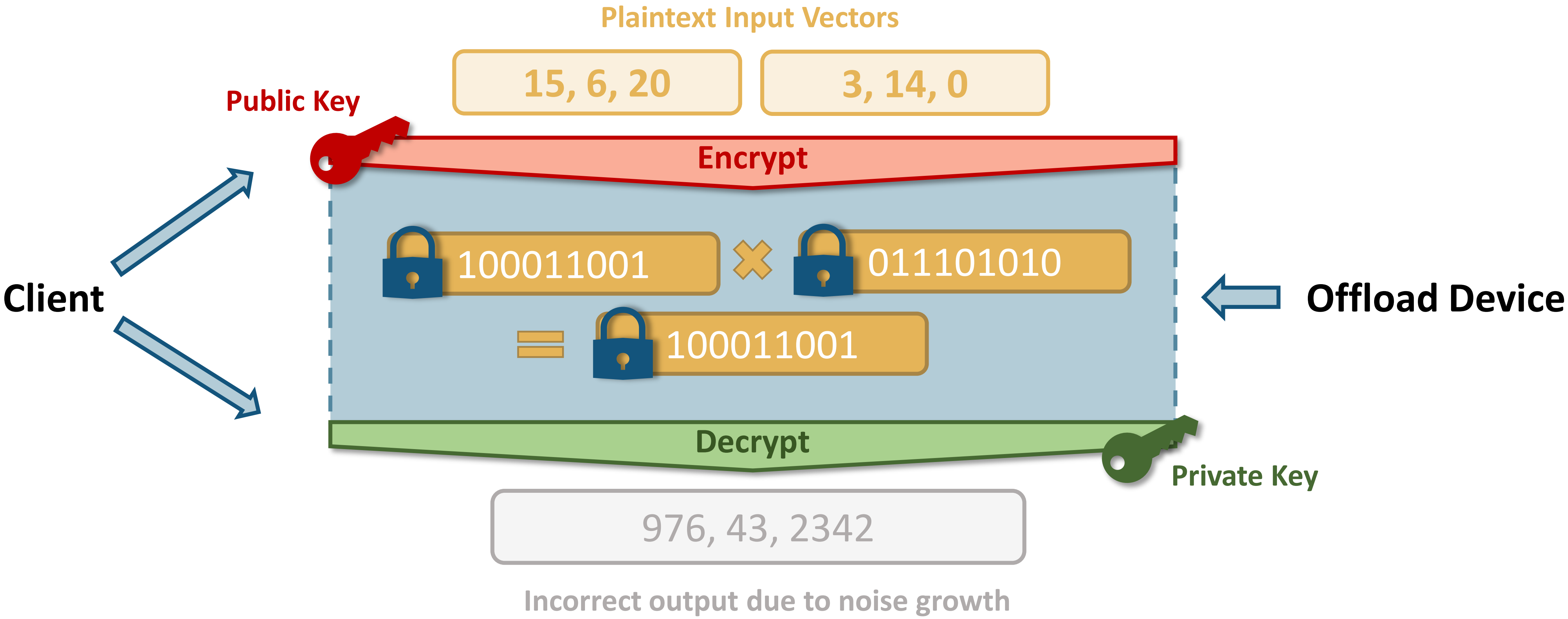
- CHOCO System Implementation

- Results & Conclusions

Outline

- Introduction
- **Encrypted Computing Background**
  - **Fully Homomorphic Encryption & Client-Aided Encrypted Computing**
- CHOCO System Implementation
- Results & Conclusions

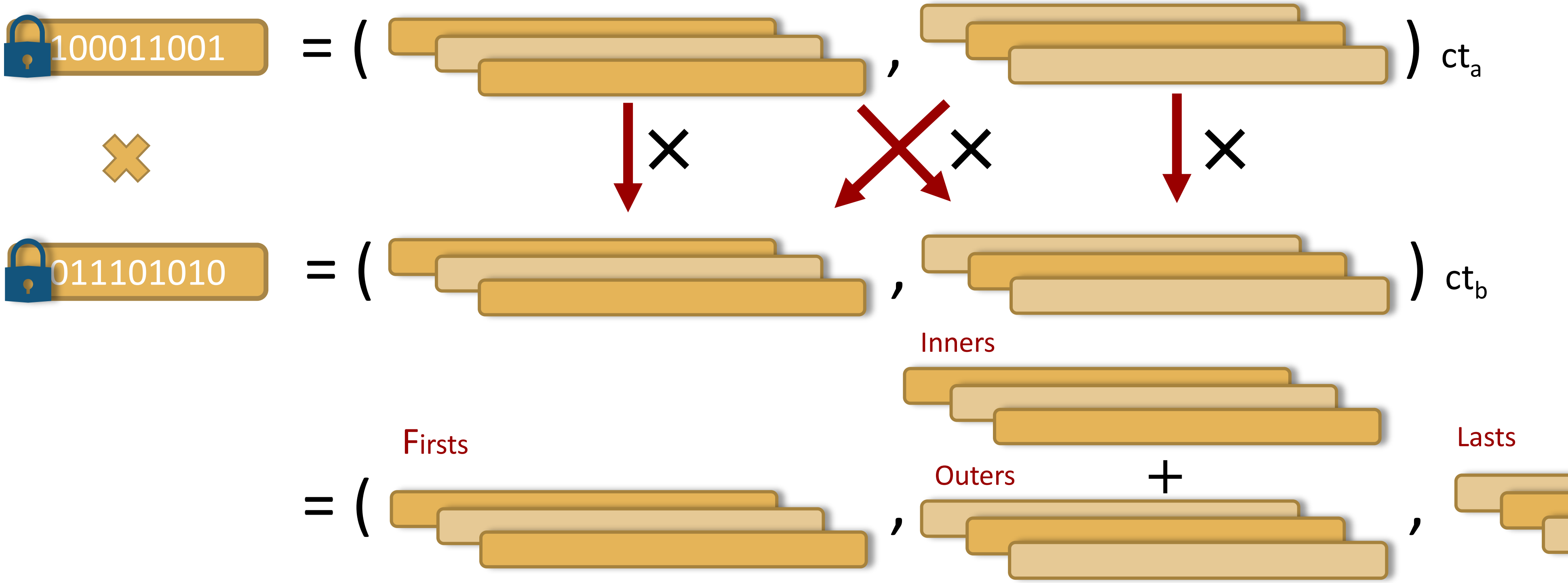
Homomorphic Encryption (HE)



Polynomial Operations

$$ct_a \times ct_b = (c0_a, c1_a) \times (c0_b, c1_b)$$

$$= ([c0_a \times c0_b]_q \, , [c0_a \times c1_b + c1_a \times c0_b\,]_q \, , [c1_a \times c1_b]_q \,)$$

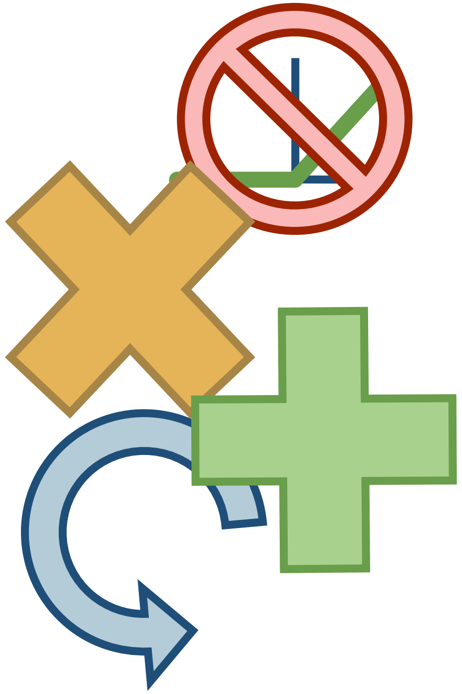


Polynomial Multiplication & Addition with Thousands of Coefficients

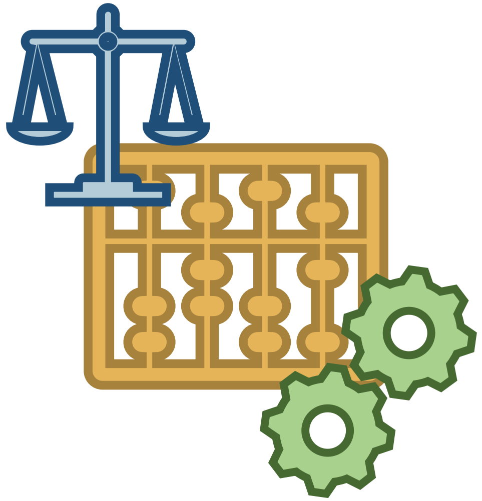
$$= (c0_c, c1_c, c2_c) \rightarrow \text{Relinearization} \rightarrow (c0_c, c1_c) = ct_c =$$ 100011001

HE Challenges & Limitations

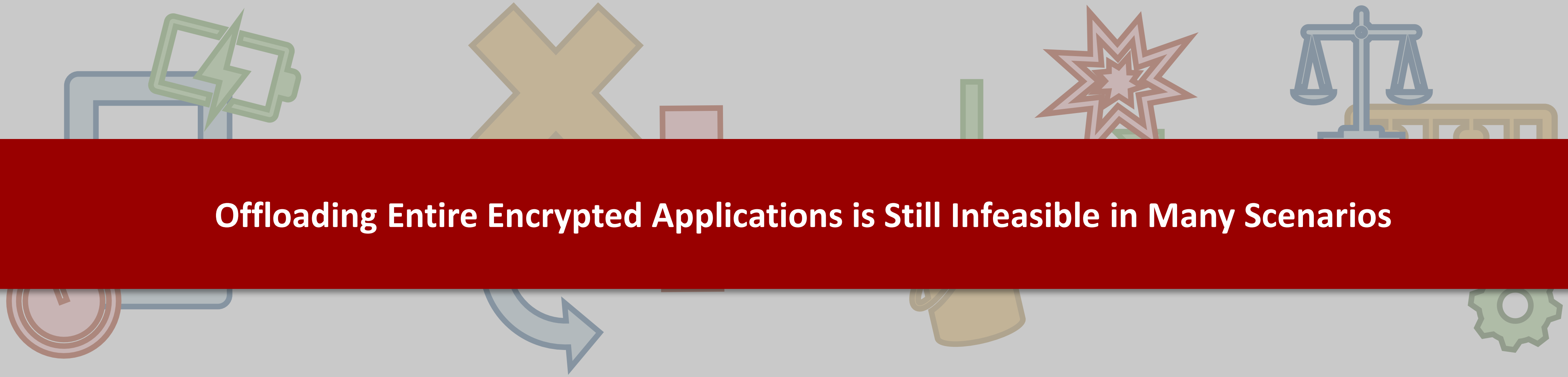

High Computation Costs


Linear Operations


Noise Growth & Arithmetic Depth


Parameter Selection

HE Challenges & Limitations



Offloading Entire Encrypted Applications is Still Infeasible in Many Scenarios

High Computation Costs

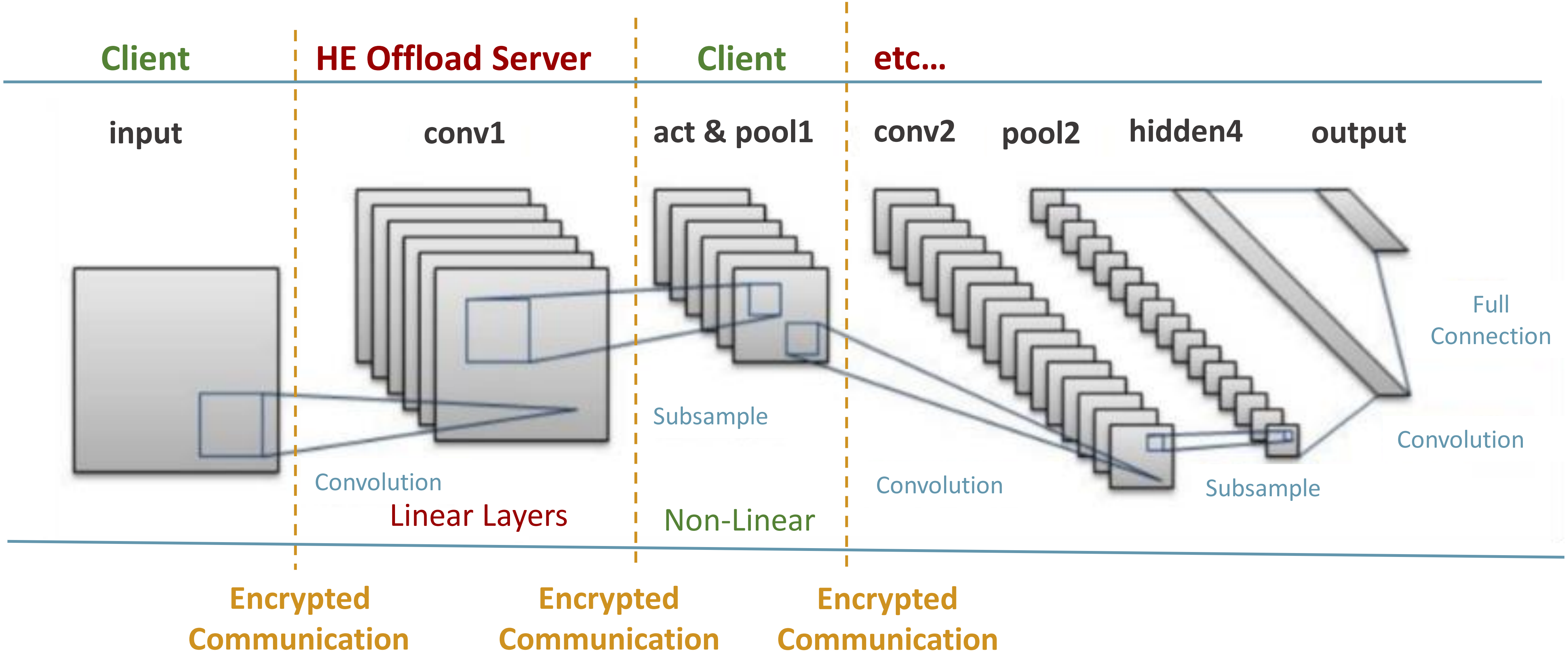
Linear Operations

Noise Growth

Parameter Selection

Client-Aided Encrypted Computing

Client-Aided Encrypted Inference

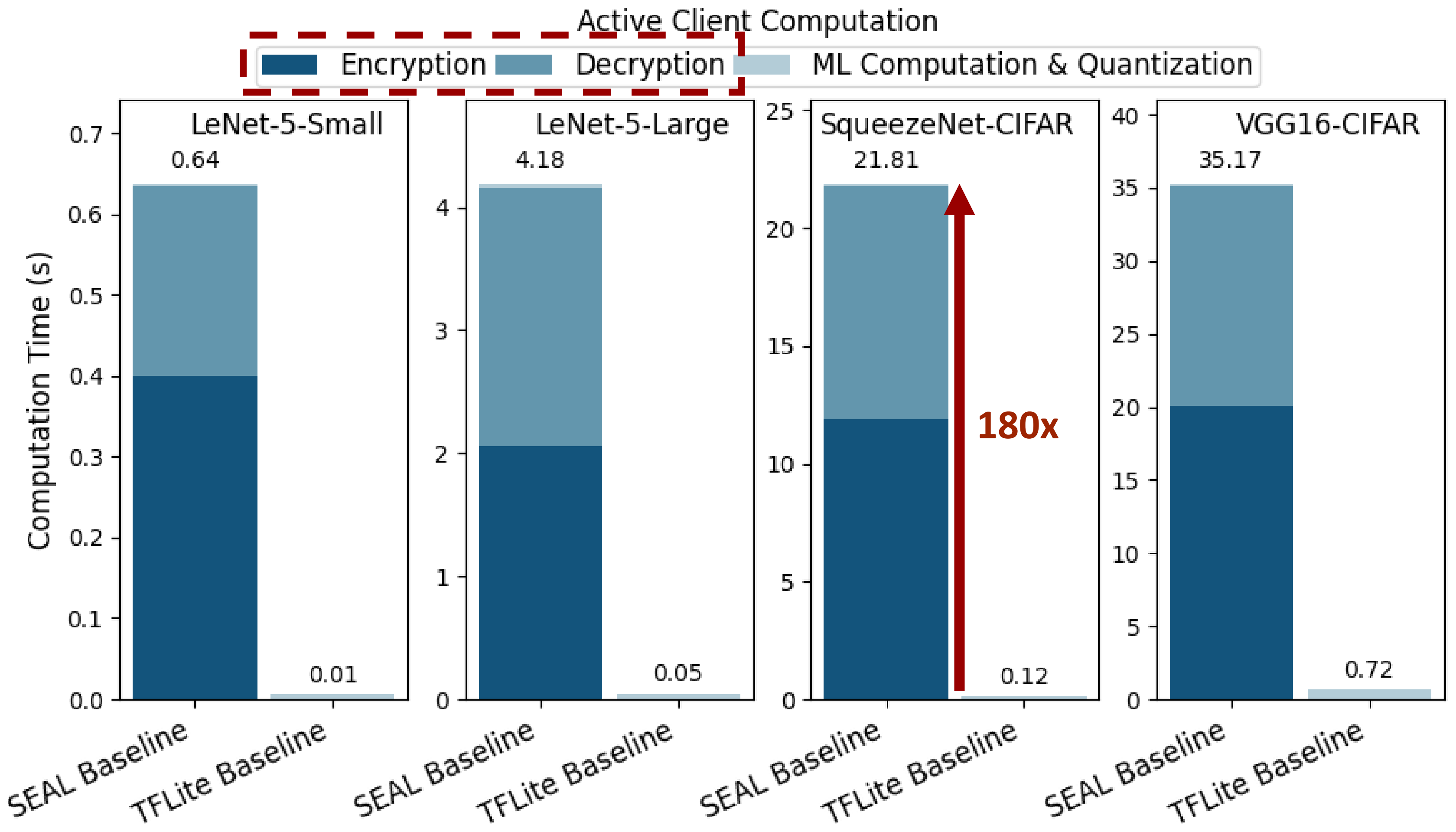


- **Systematically limits arithmetic depth & regularly refreshes noise**

Image: https://www.pyimagesearch.com/2016/08/01/lenet-convolutional-neural-network-in-python/

Quantifying Client Responsibility

- ARM Cortex-A7 CPU Client
- **Up to 180x** overhead to offload compute
- Dominated by Homomorphic Encryption (HE) operations

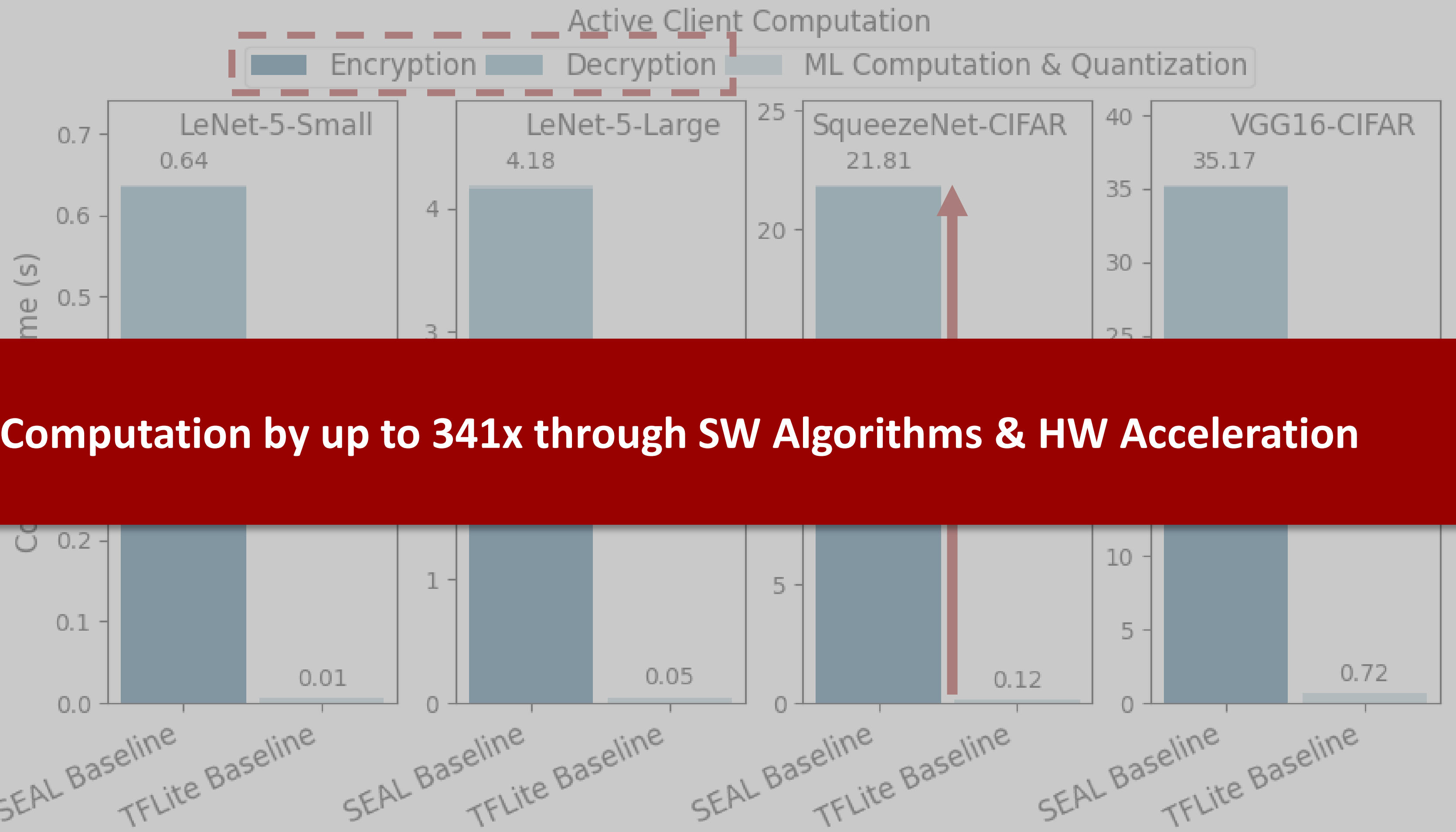


Microsoft Research. 2019. *Simple Encrypted Arithmetic Library (release 3.4)*, https://github.com/Microsoft/SEAL
M. Abadi et al. 2015. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems (release 2.2).* https://www.tensorflow.org/.

Quantifying Client Responsibility

- ARM Cortex-A7 CPU Client
- **Up to 180x** overhead to

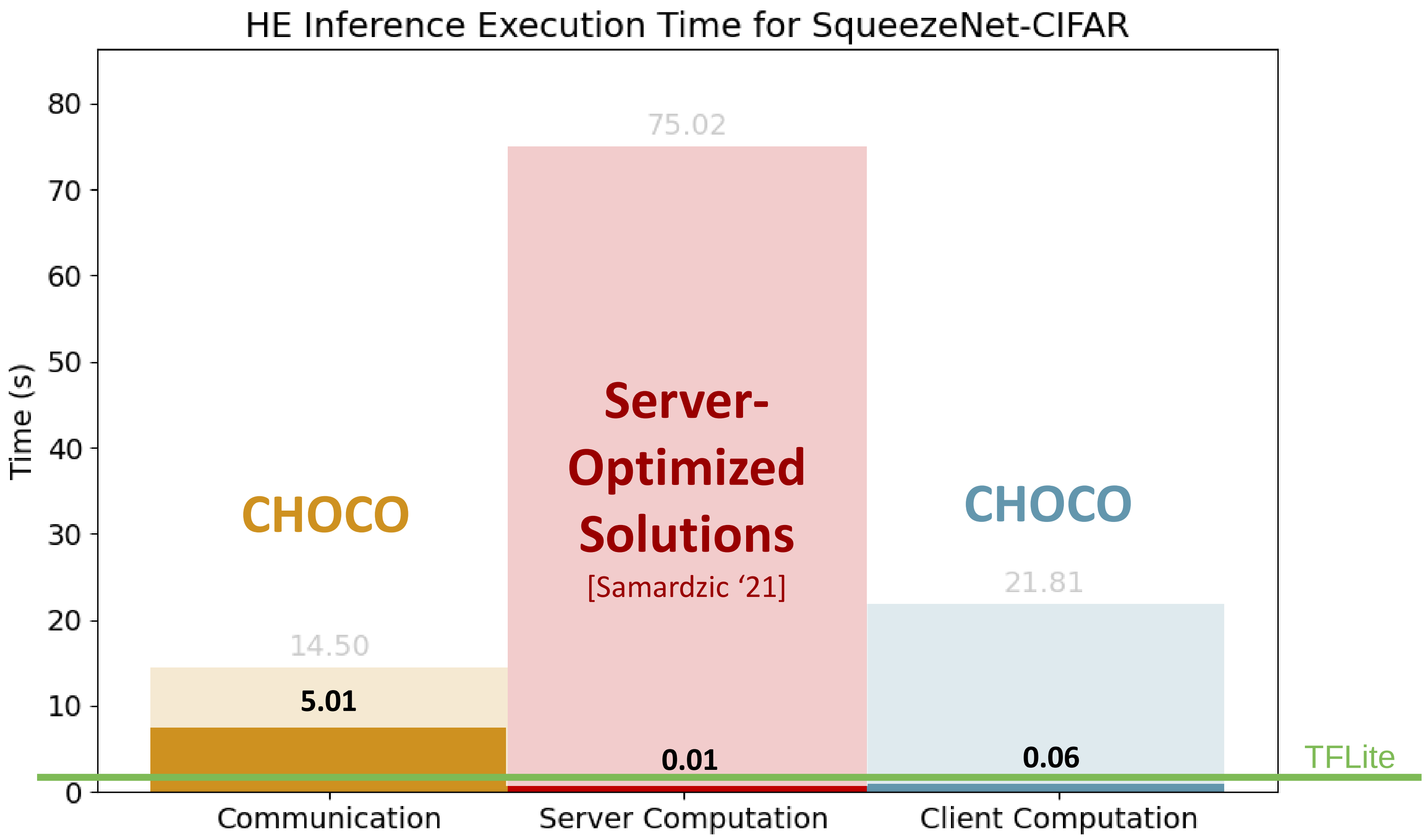
Encryption (HE) operations



CHOCO Reduces Client-Side Computation by up to 341x through SW Algorithms & HW Acceleration

Microsoft Research. 2019. *Simple Encrypted Arithmetic Library (release 3.4)*, https://github.com/Microsoft/SEAL
M. Abadi et al. 2015. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems (release 2.2).* https://www.tensorflow.org/.

Complete Client-Aided System Improvements



Nikola Samardzic, Axel Feldmann, Aleksandar Krastev, Srinivas Devadas, Ronald Dreslinski, Christopher Peikert, and Daniel Sanchez. 2021. *F1: A Fast and Programmable Accelerator for Fully Homomorphic Encryption*. Association for Computing Machinery, New York, NY, USA, 238–252.

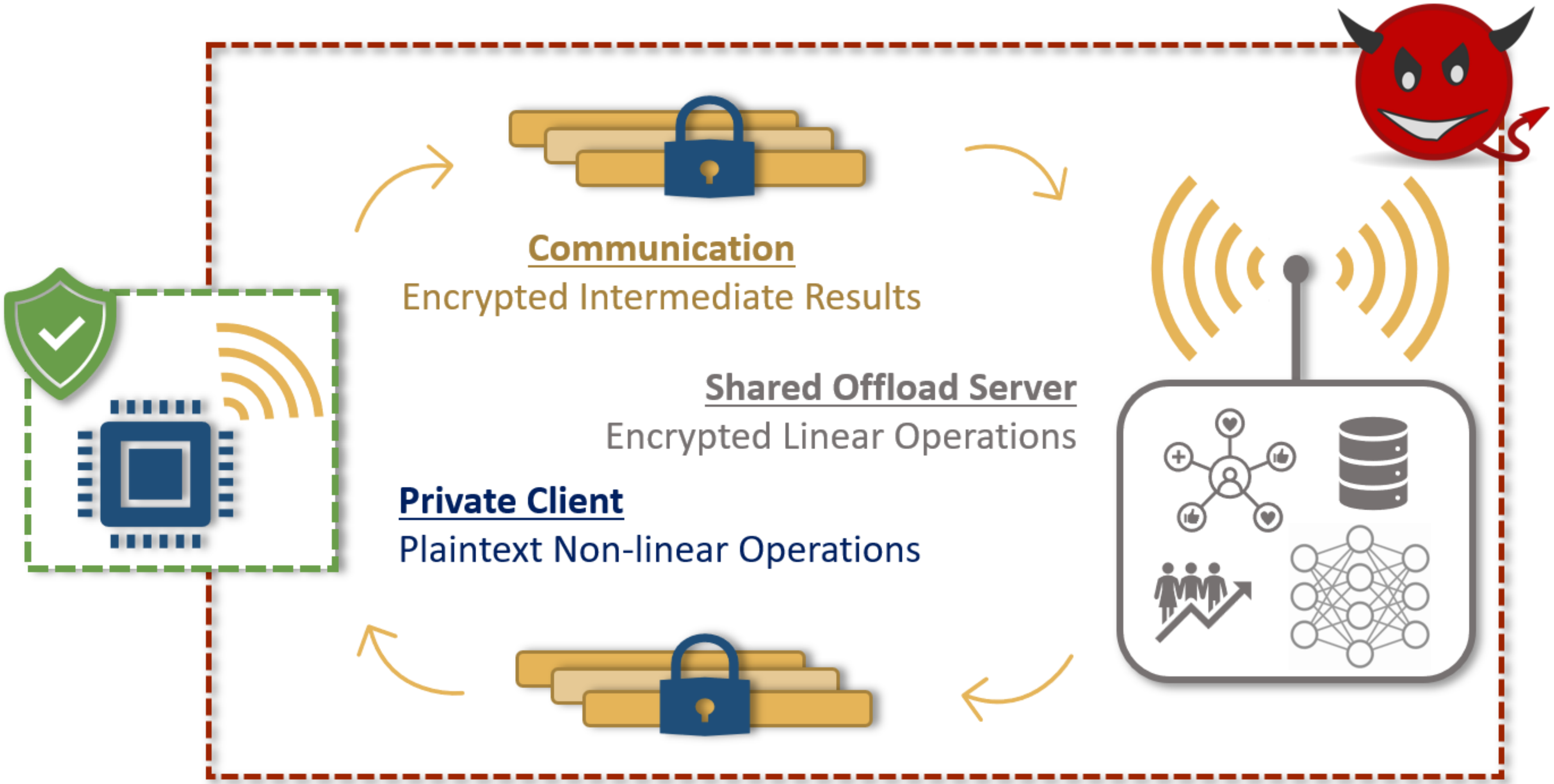
Outline

- Introduction
- **Encrypted Computing Background**
  - **Fully Homomorphic Encryption & Client-Aided Encrypted Computing**
- CHOCO System Implementation
- Results & Conclusions

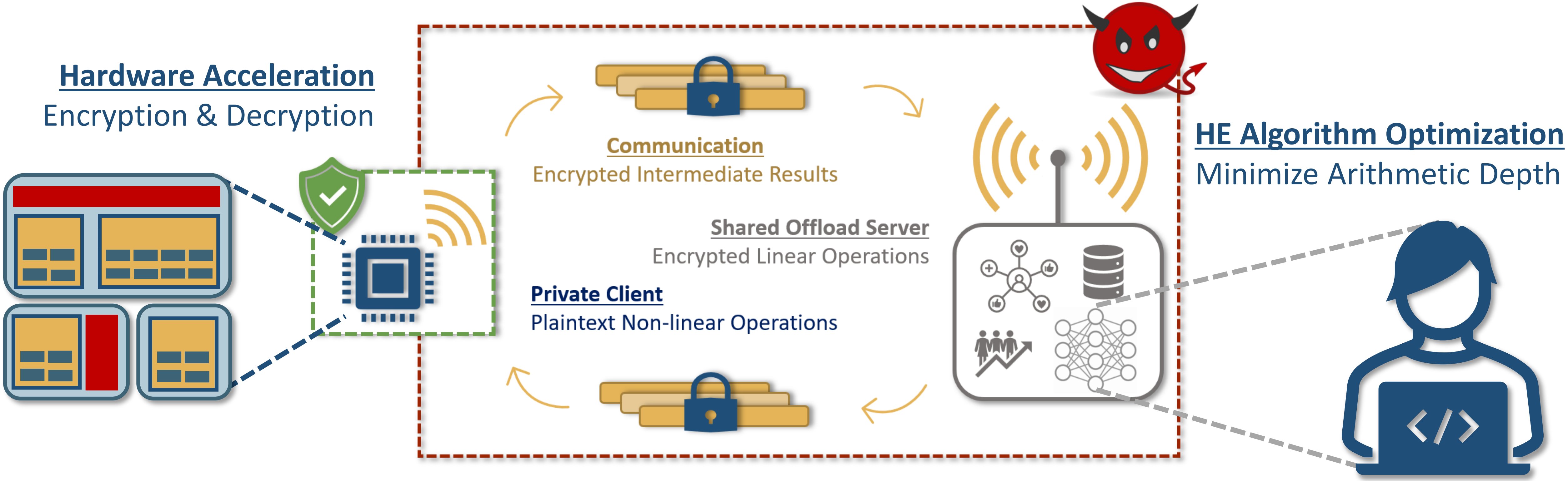
Outline

- Introduction
- Encrypted Computing Background
- **CHOCO System Implementation**
- **Algorithm Optimizations & Hardware Acceleration**
- Results & Conclusions

Client-aided HE for Opaque Compute Offloading

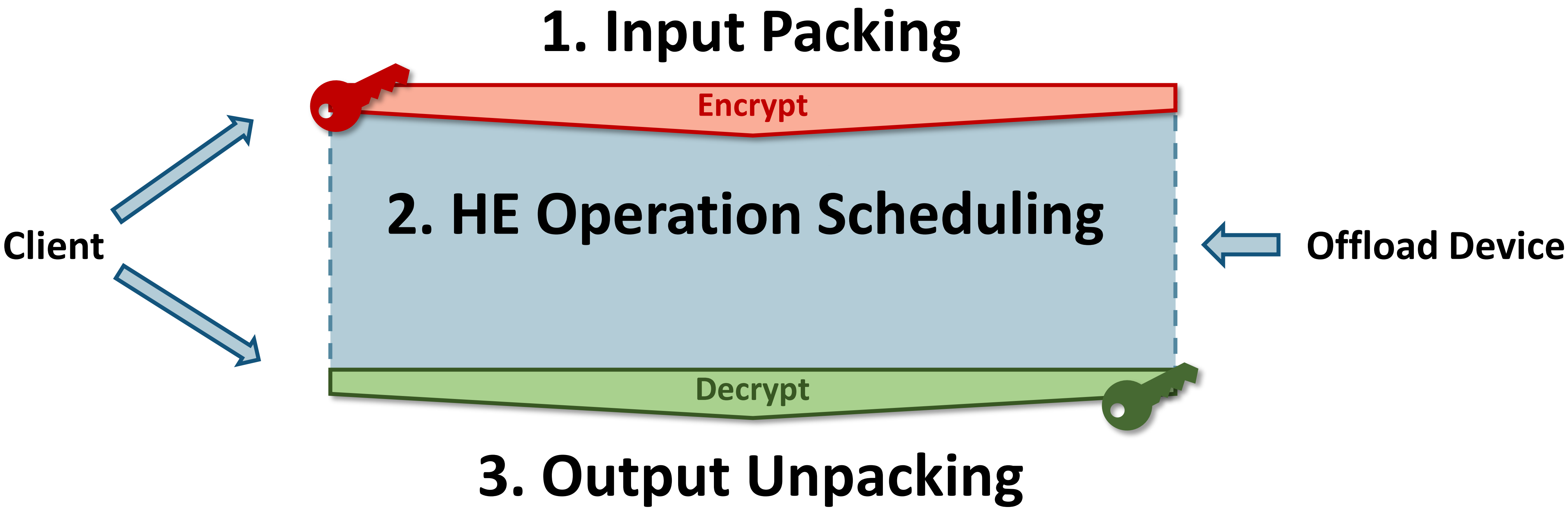


Client-aided HE for Opaque Compute Offloading

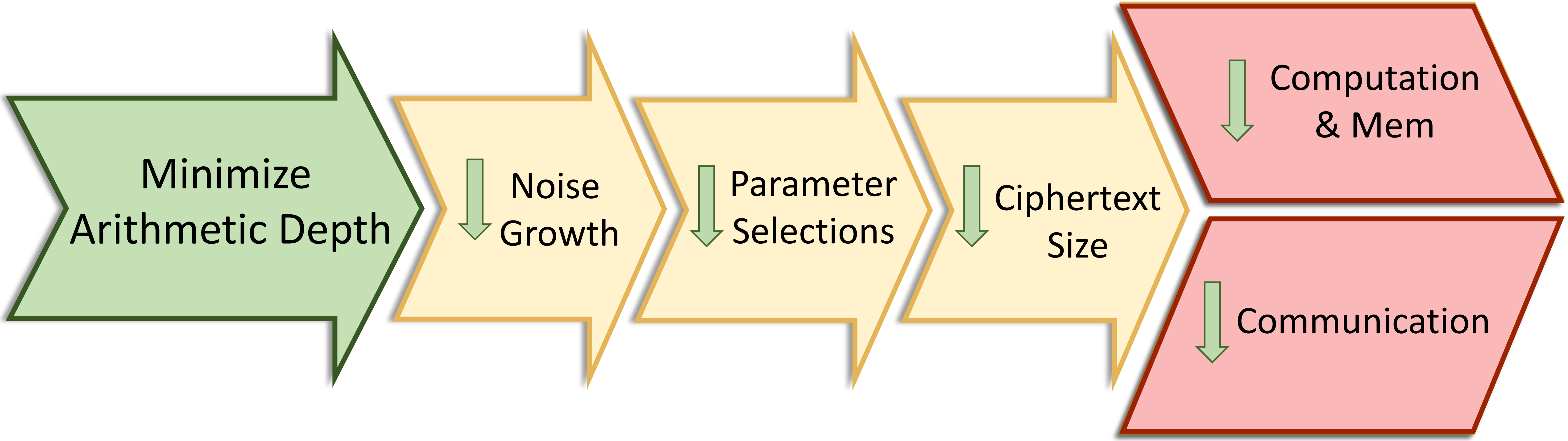


Encrypted Algorithm Optimization

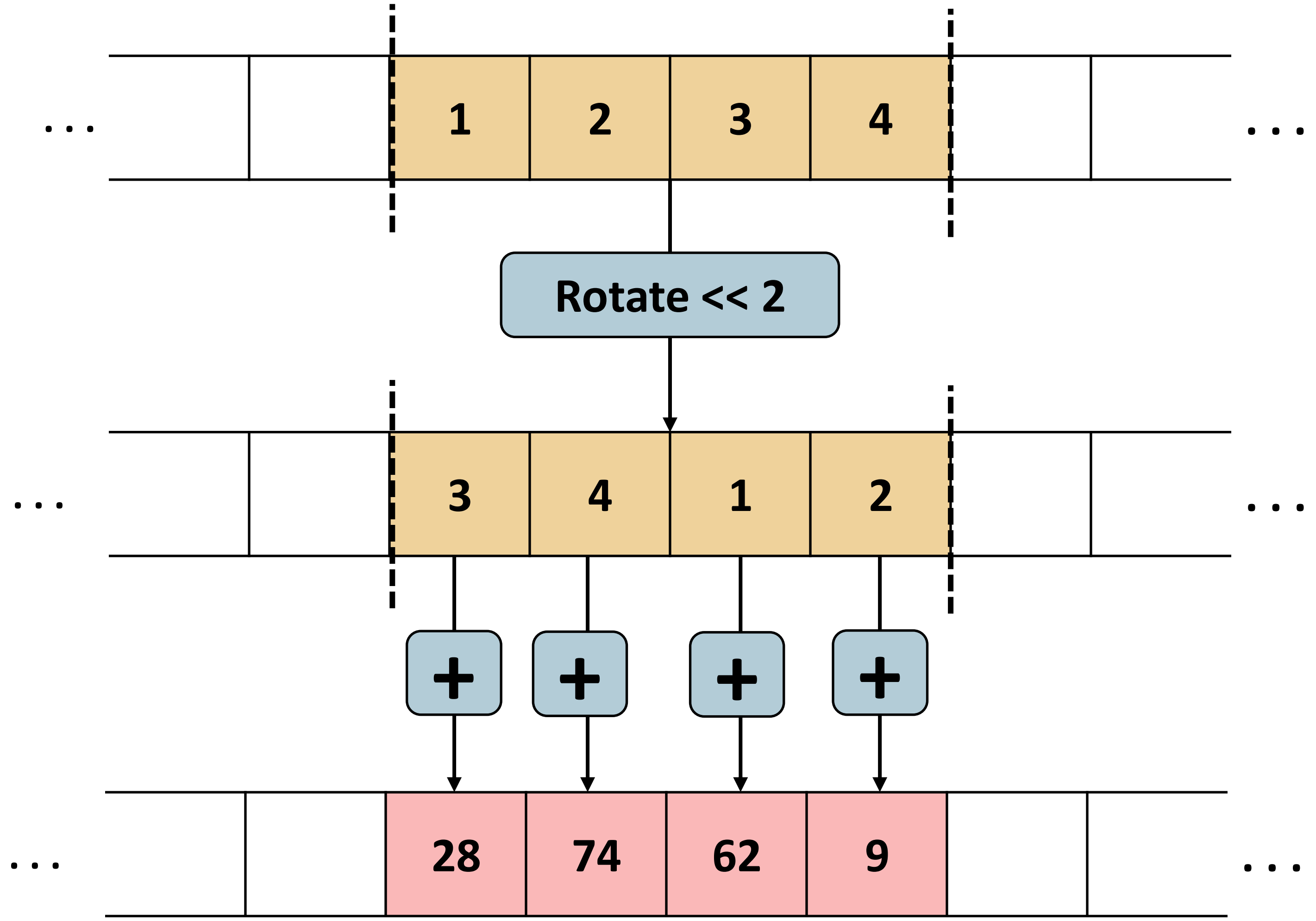
HE Algorithms



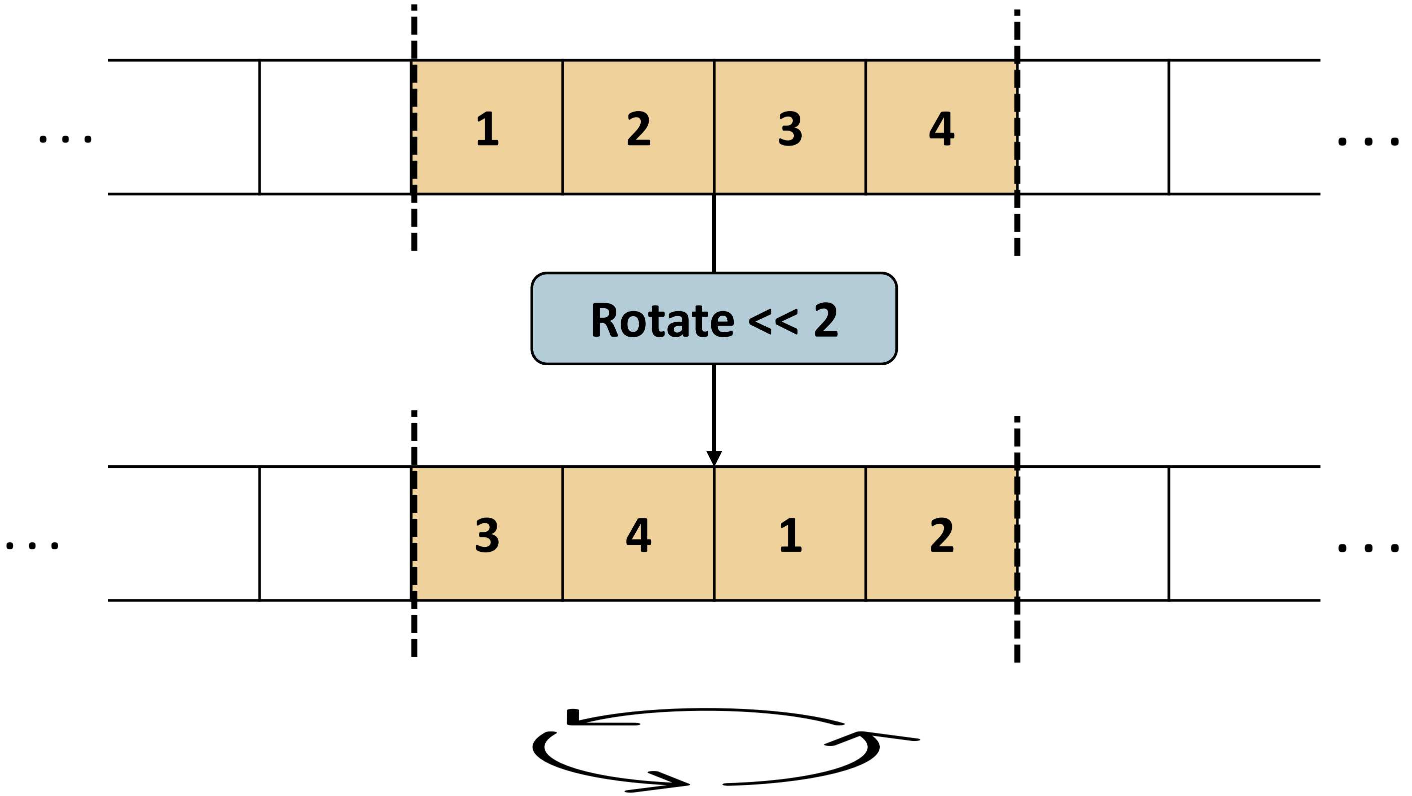
Algorithm Optimizations Impact Client Costs



Windowed Rotations

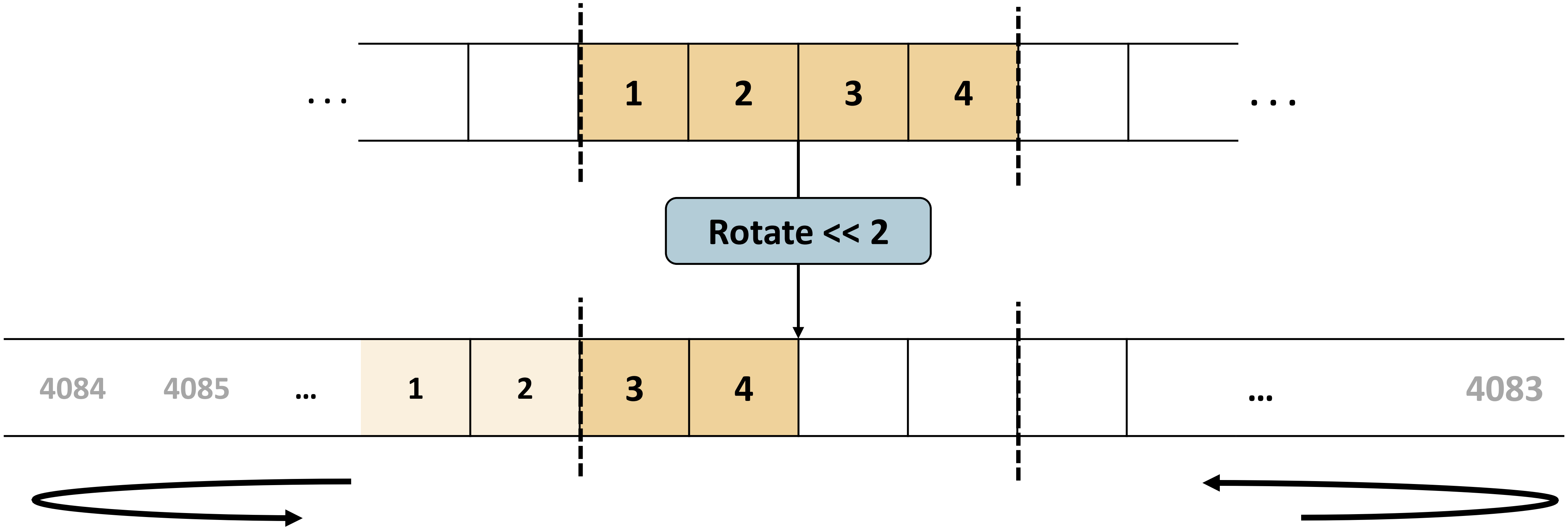


Windowed Rotations



Ideally: Values wrap around within a window of interest

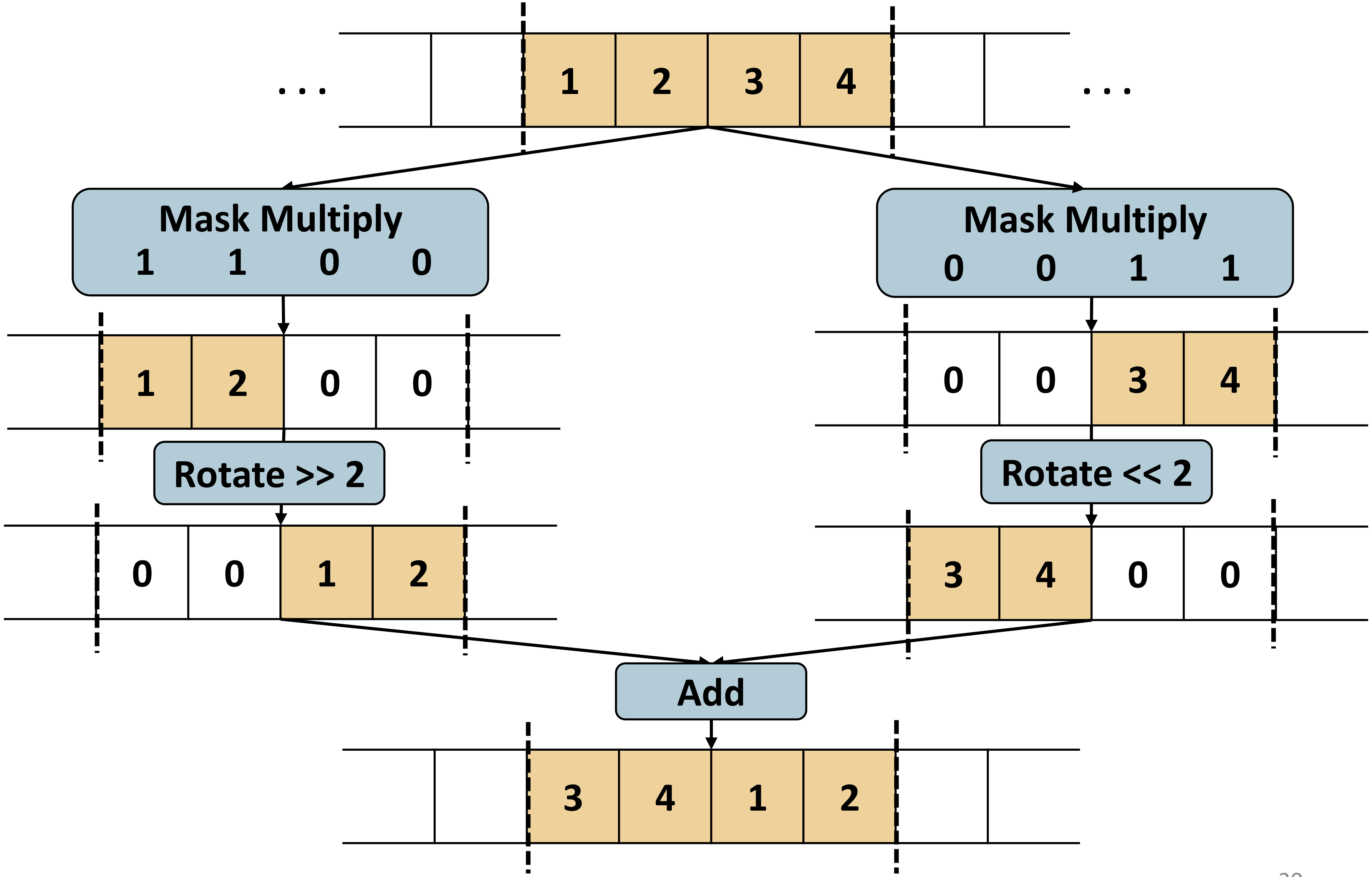
Windowed Rotations



Actually: Values wrap around the entire ciphertext vector

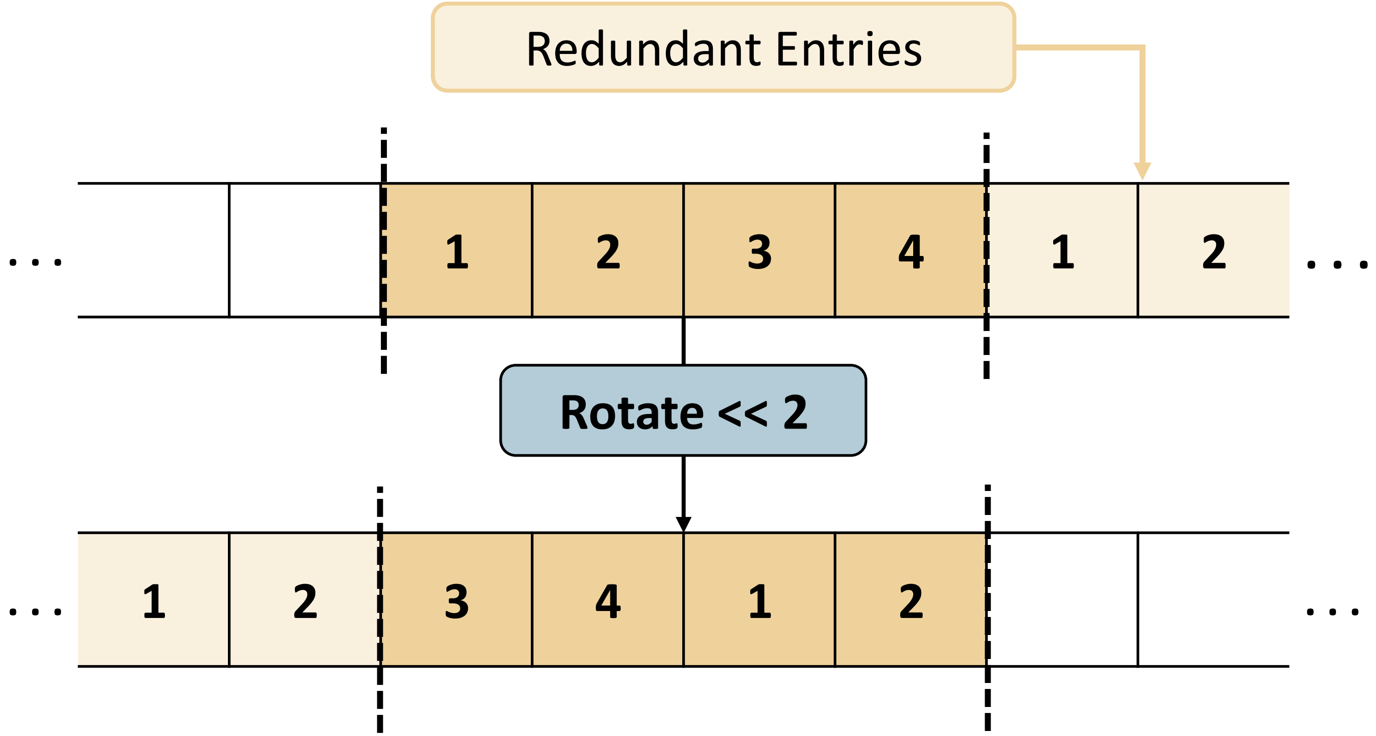
Standard Permutations

- Expensive
- Computation & Noise Growth
- High Arithmetic Depth

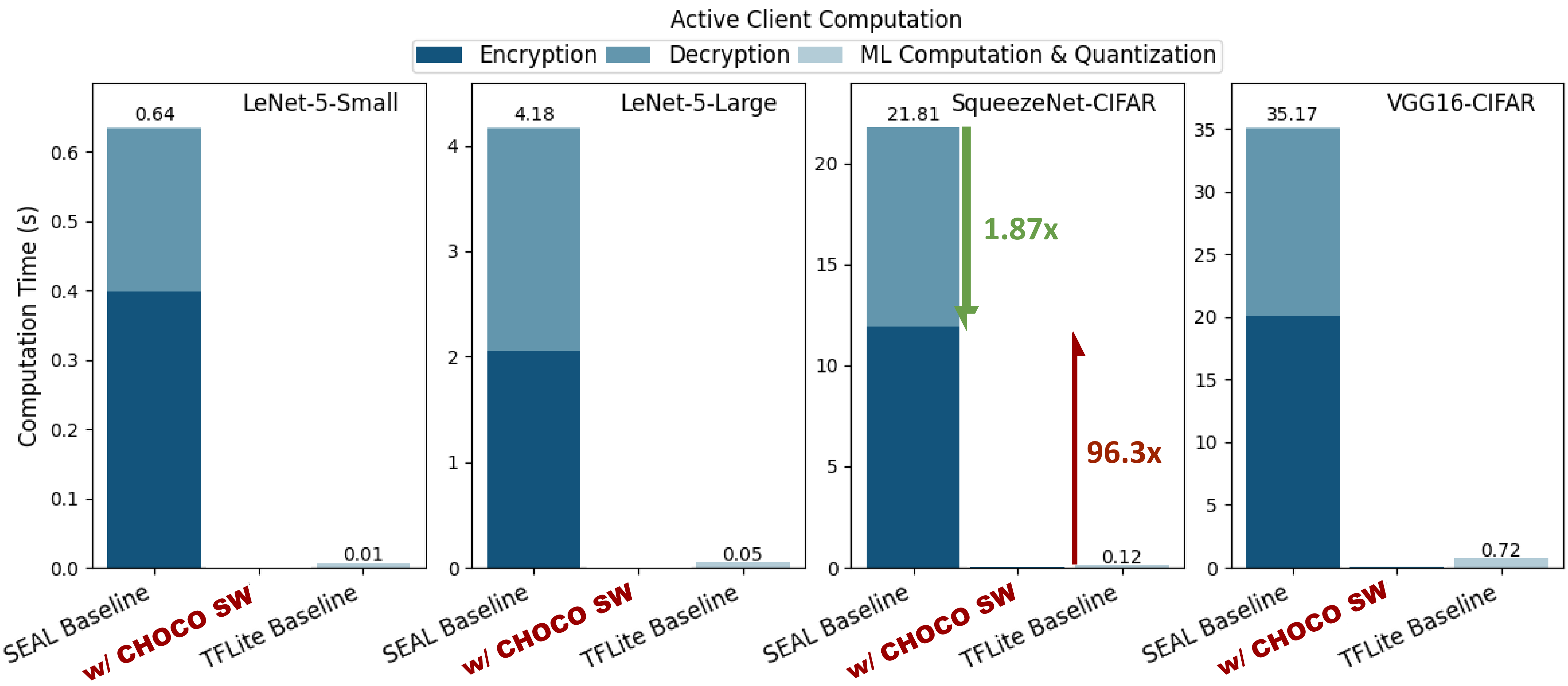


Rotational Redundancy

- Novel Input Packing
- Single HE Rotation
- Low arithmetic depth

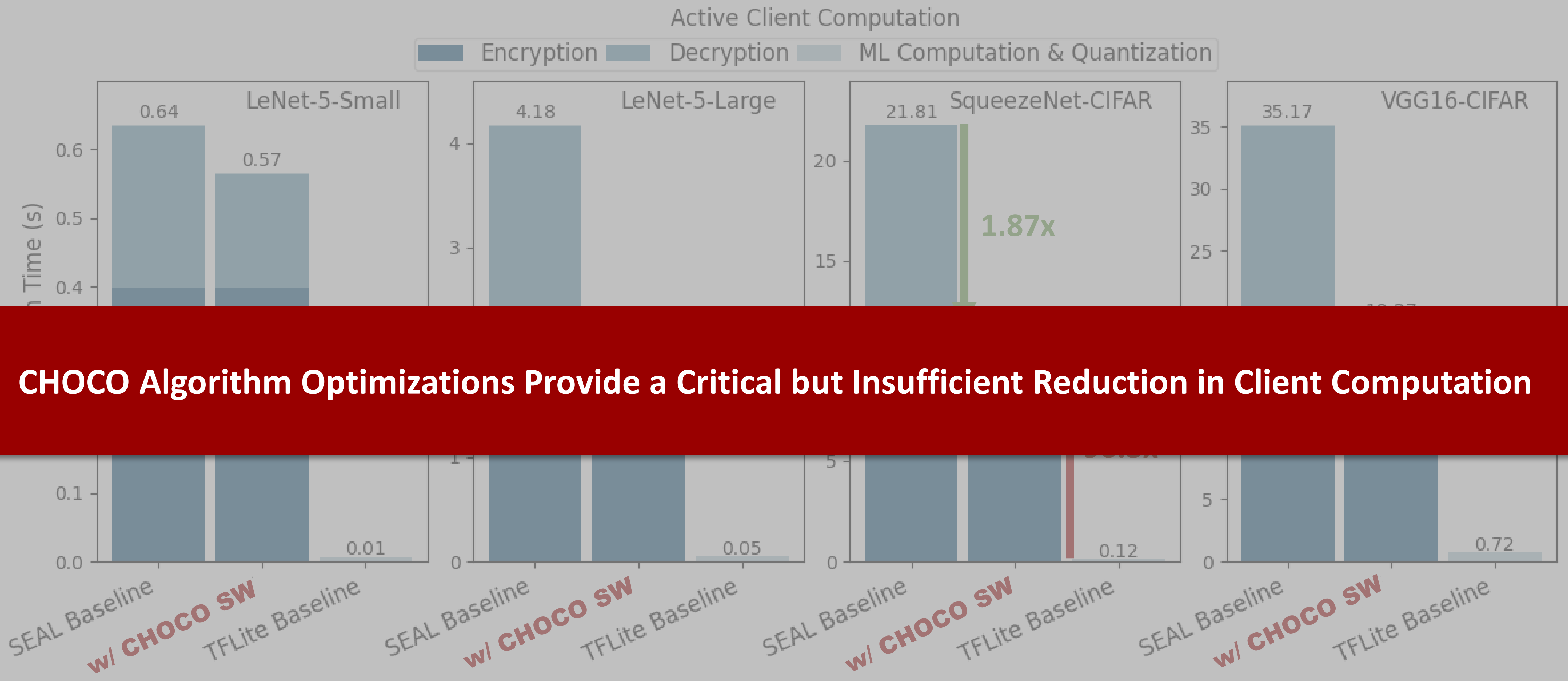


CHOCO Algorithms Reduce Client Computation



- **50%** Smaller Ciphertexts
- Average **1.7x** improvement over SEAL
- **CHOCO SW** = SEAL baseline + Rotational Redundancy
- Average **62.5x** remaining overhead vs TFLite

CHOCO Algorithms Reduce Client Computation

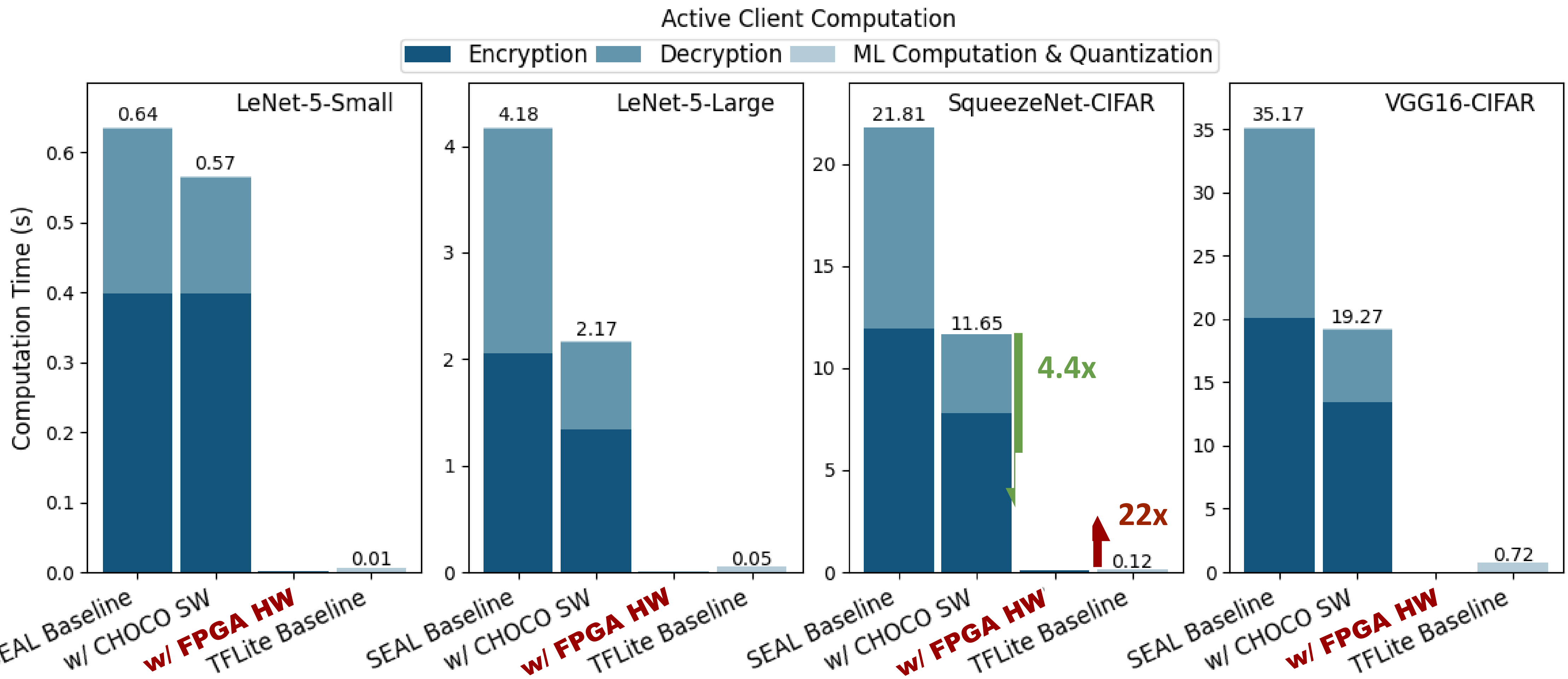


CHOCO Algorithm Optimizations Provide a Critical but Insufficient Reduction in Client Computation

- CHOCO SW = SEAL baseline + Rotational Redundancy
- 50% Smaller Ciphertexts
- Average 1.7x improvement over SEAL
- Average 62.5x remaining overhead vs TFLite

Hardware Acceleration

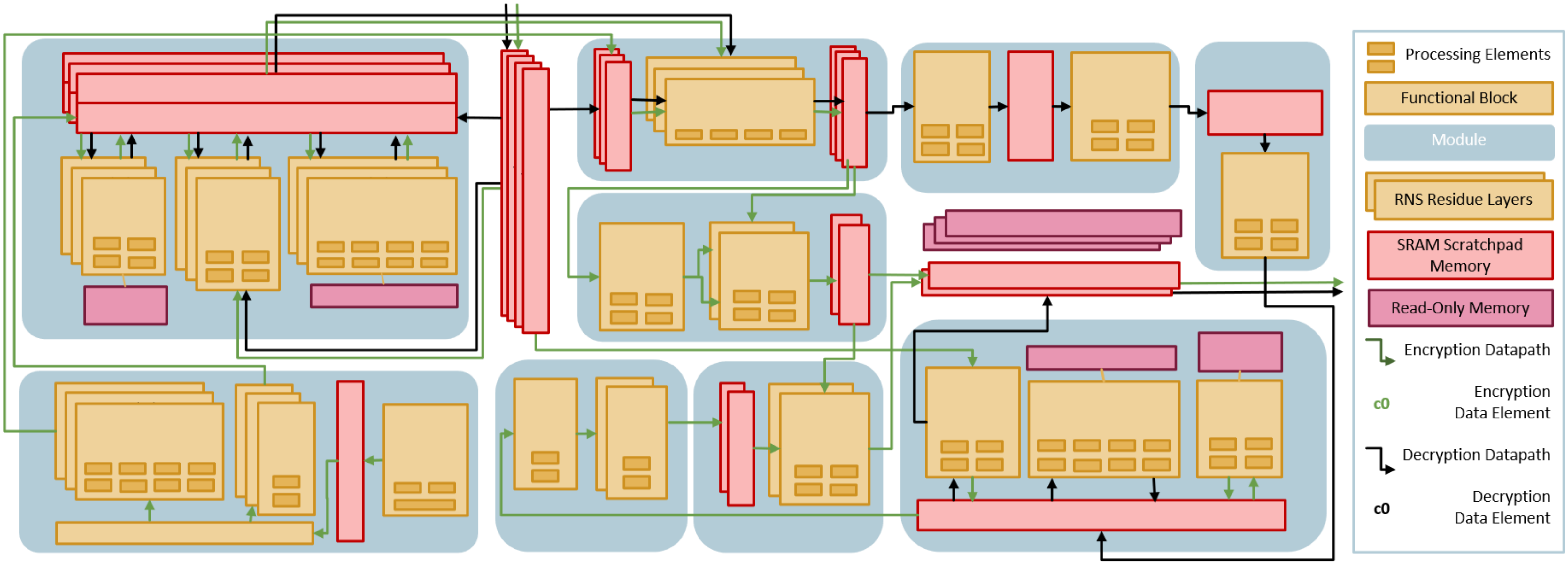
Existing FPGA Acceleration is Incomplete



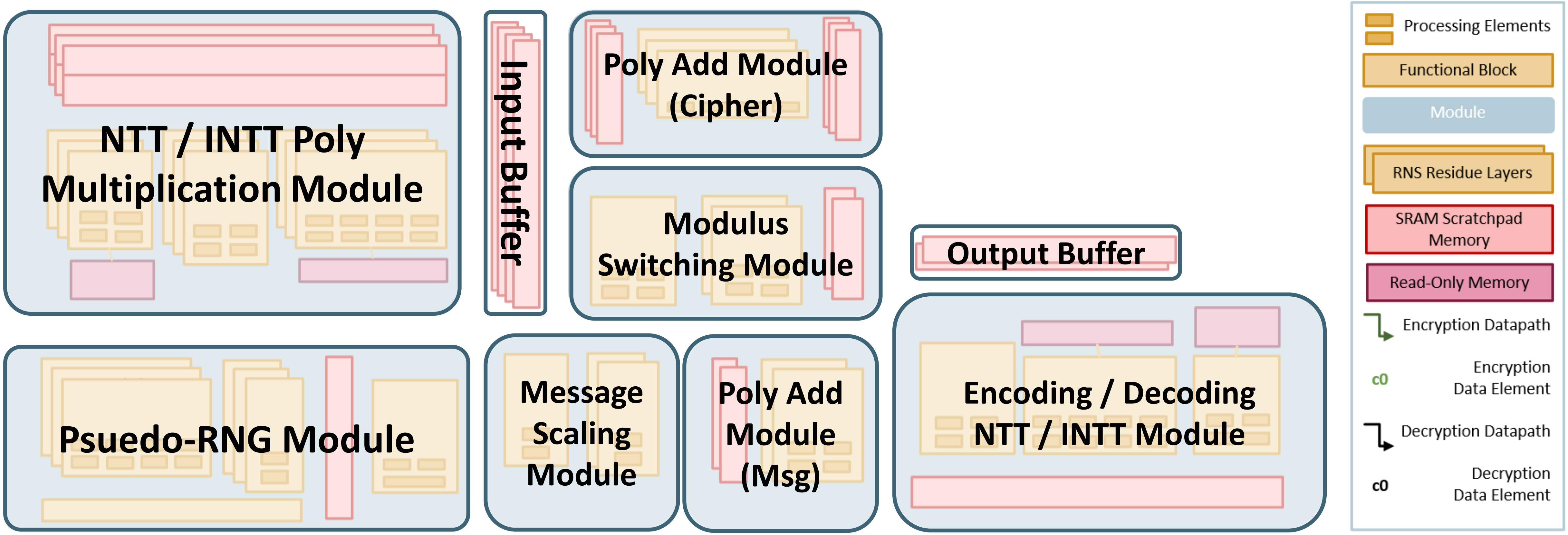
- **FPGA HW** = CHOCO SW + Encryption/Decryption FPGA
- Average **14.5x** remaining overhead vs TFLite

A. Mert, E. Ozturk, and E. Savas. 2020. *Design and Implementation of Encryption/Decryption Architectures for BFV Homomorphic Encryption Scheme.* IEEE Transactions on Very Large Scale Integration (VLSI) Systems 28, 02 (feb 2020), 353–362.

CHOCO – Through Accelerated Cryptographic Operations



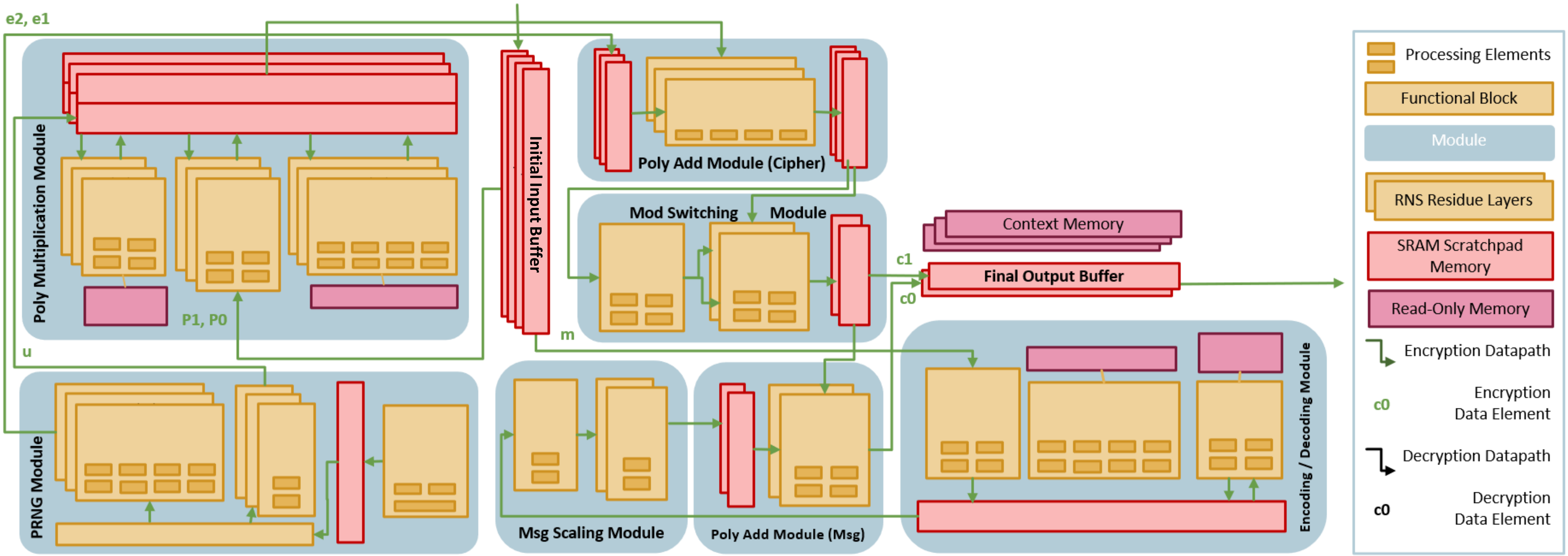
CHOCO-TACO Encryption & Decryption Hardware



- $\texttt{Encrypt}(\texttt{pk}, m)$: For $m \in R_t$, let $\texttt{pk} = (p_0, p_1)$. Sample $u \xleftarrow{\$} R_2$, and $e_1, e_2 \leftarrow \chi$. Compute

$$\texttt{ct} = ([\Delta m + p_0 u + e_1]_q, [p_1 u + e_2]_q) \; .$$

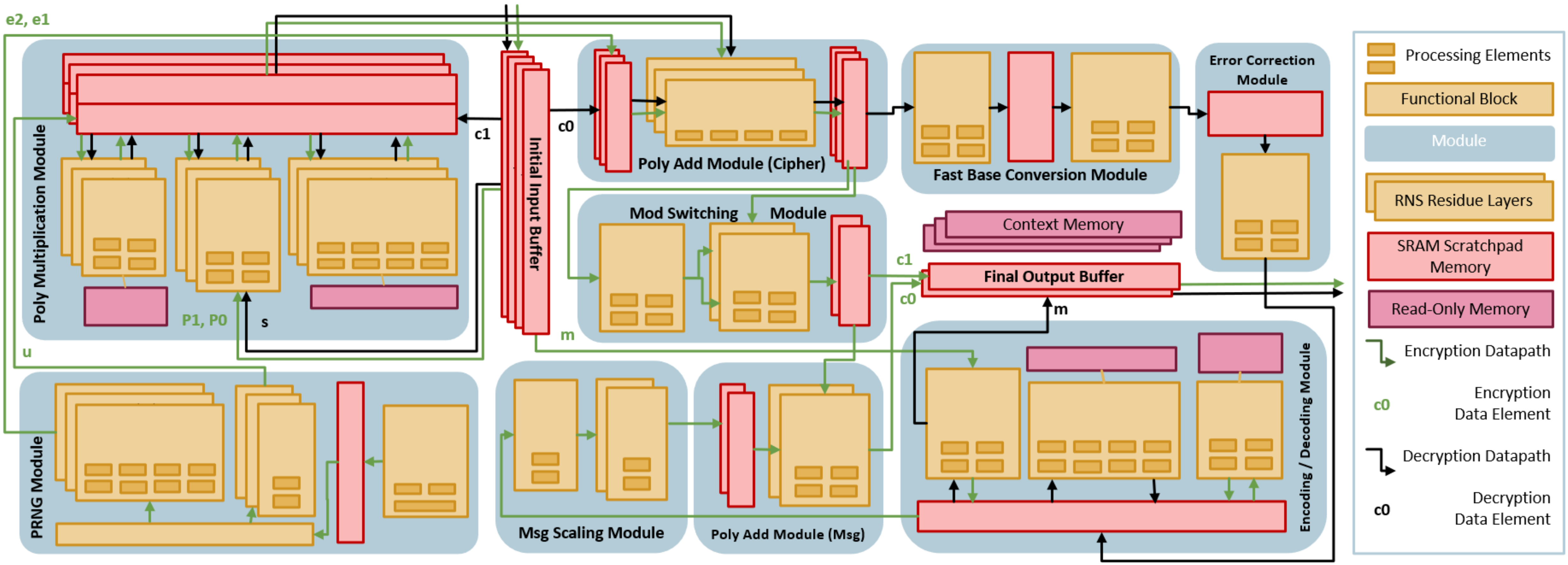
CHOCO-TACO Encryption & Decryption Hardware



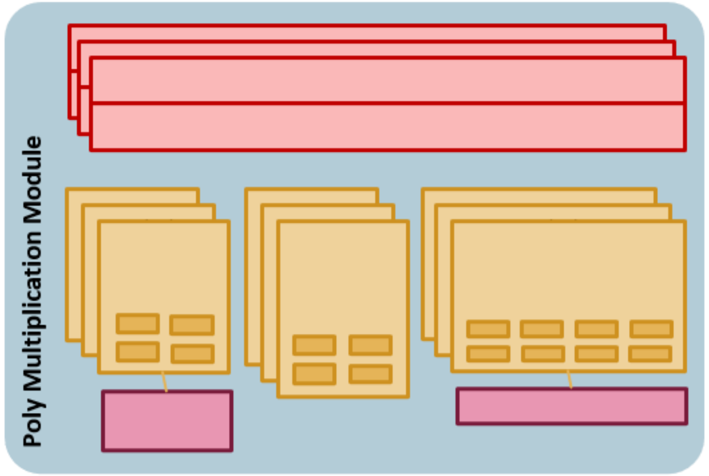
- $\mathtt{Encrypt}(\mathsf{pk}, m)$: For $m \in R_t$, let $\mathsf{pk} = (p_0, p_1)$. Sample $u \overset{\$}{\leftarrow} R_2$, and $e_1, e_2 \leftarrow \chi$. Compute

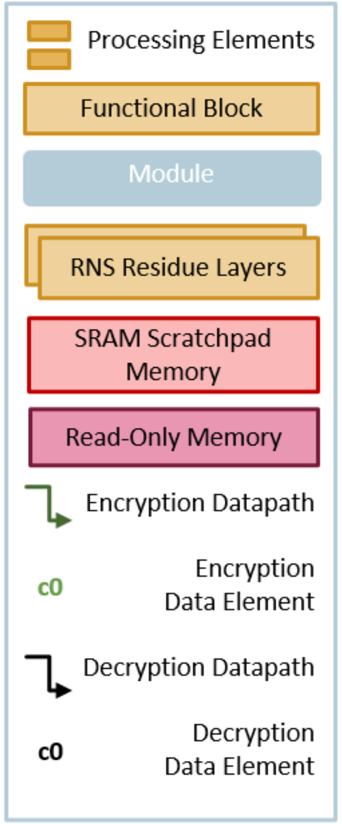
$$\mathtt{ct} = ([\Delta m + p_0 u + e_1]_q, [p_1 u + e_2]_q) \ .$$

CHOCO-TACO Encryption & Decryption Hardware

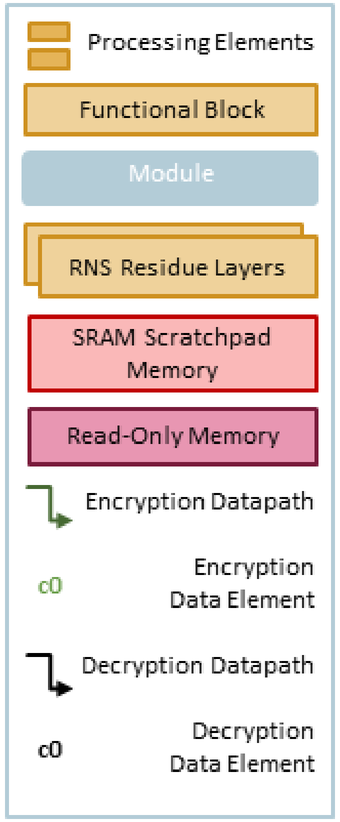


CHOCO-TACO Hardware Optimization





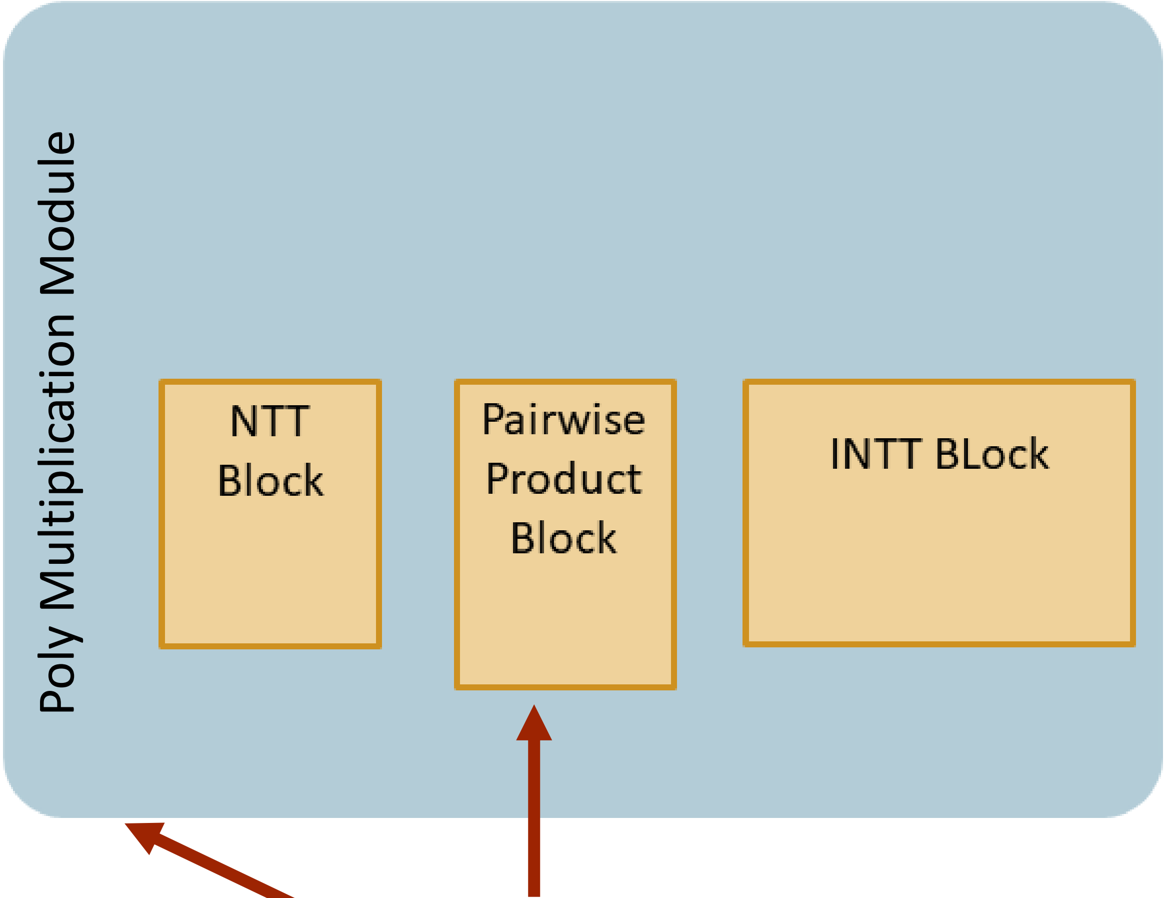
CHOCO-TACO Hardware Optimization

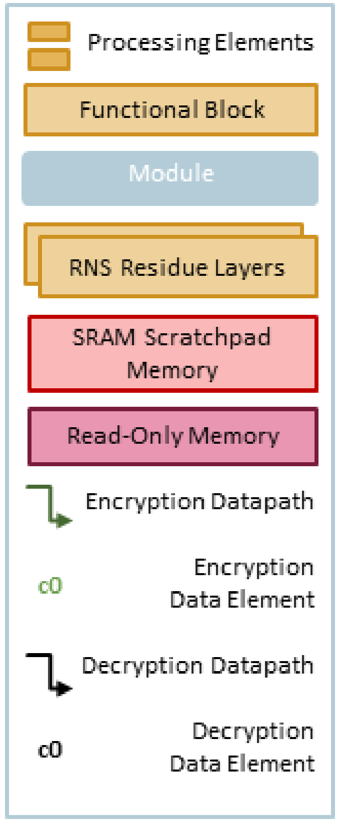


Modules & Functional Blocks

Specialized & Comprehensive

CHOCO-TACO Hardware Optimization

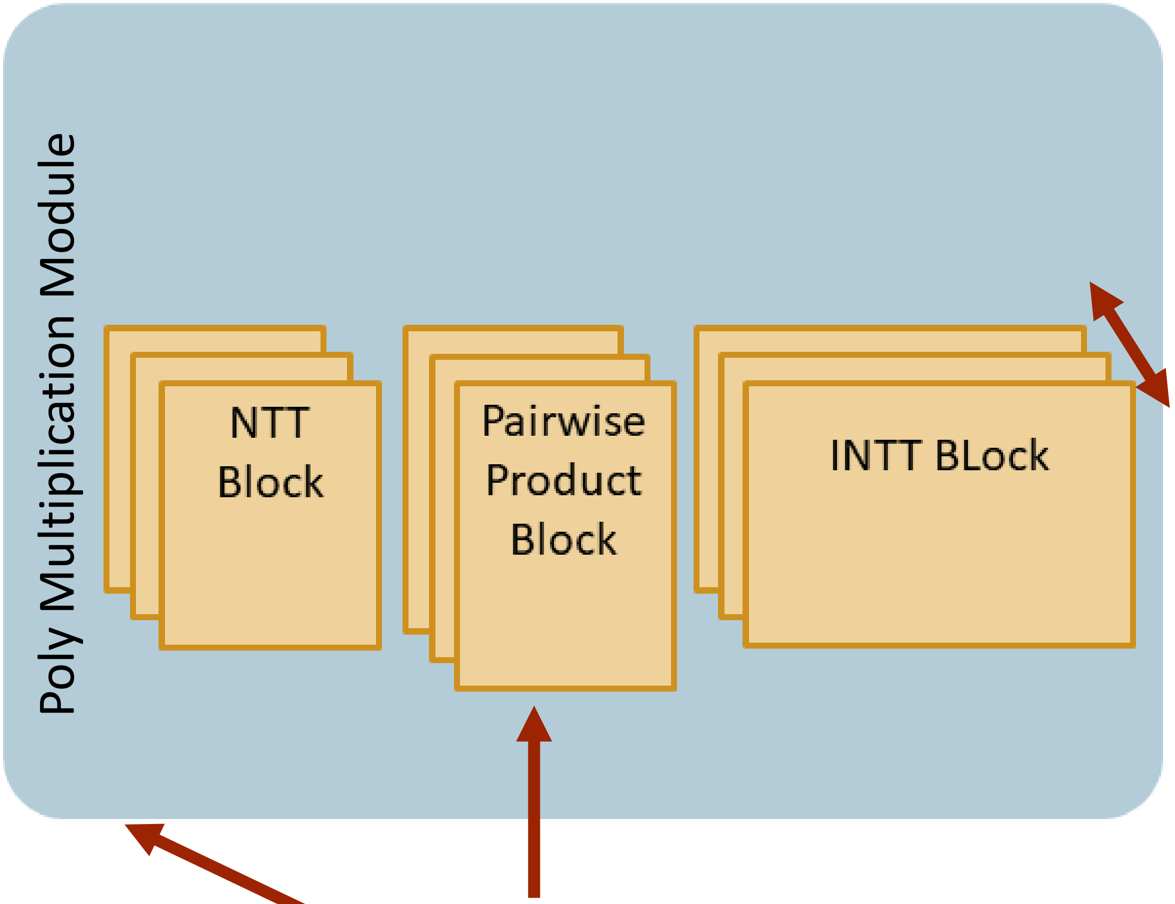




Modules & Functional Blocks

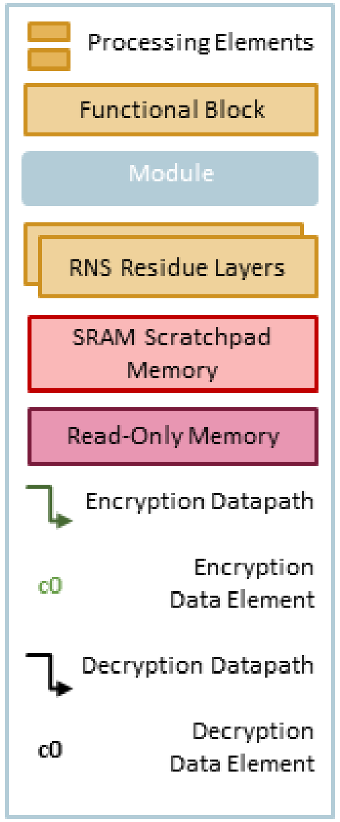
Pipelined & Comprehensive

CHOCO-TACO Hardware Optimization

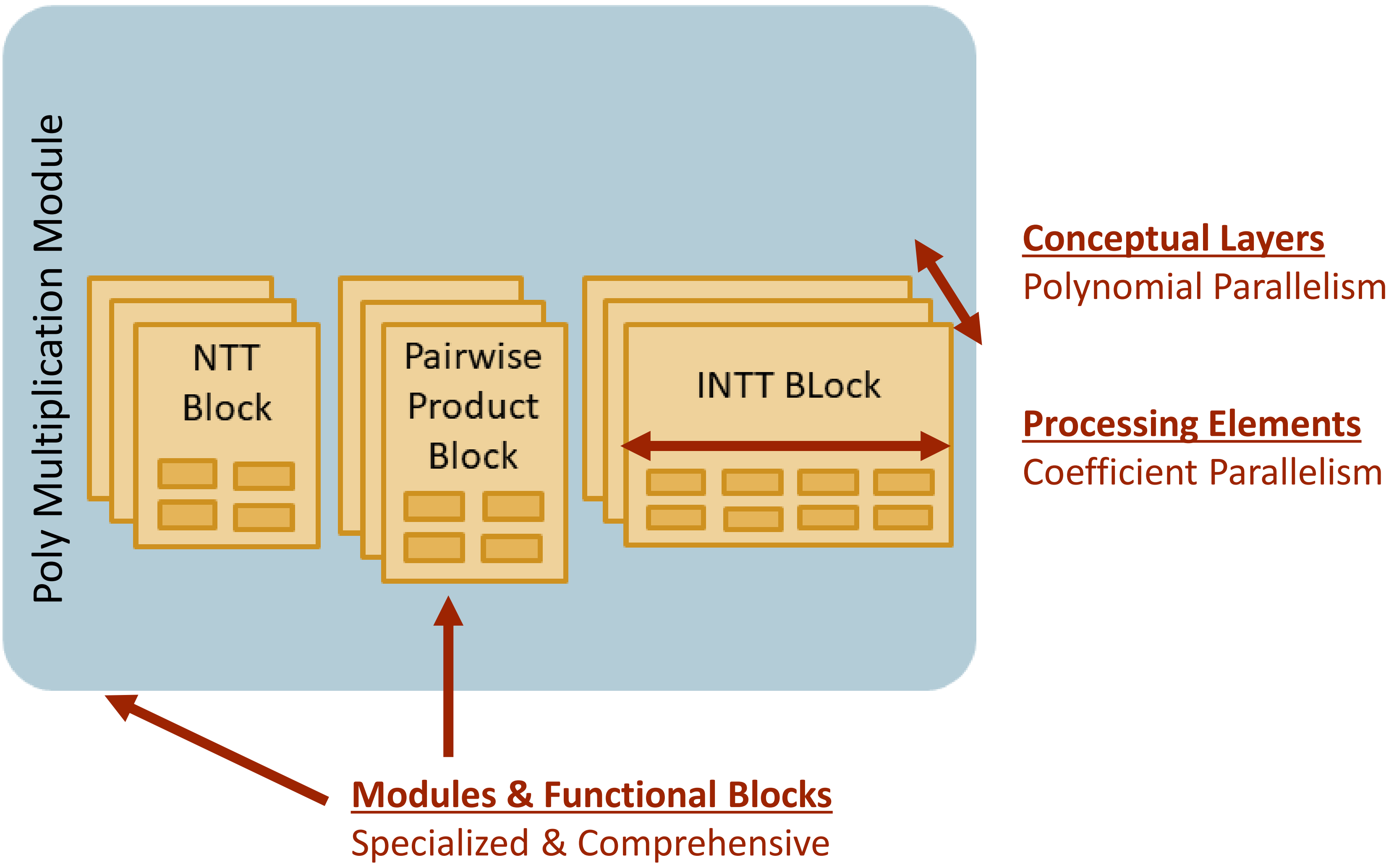


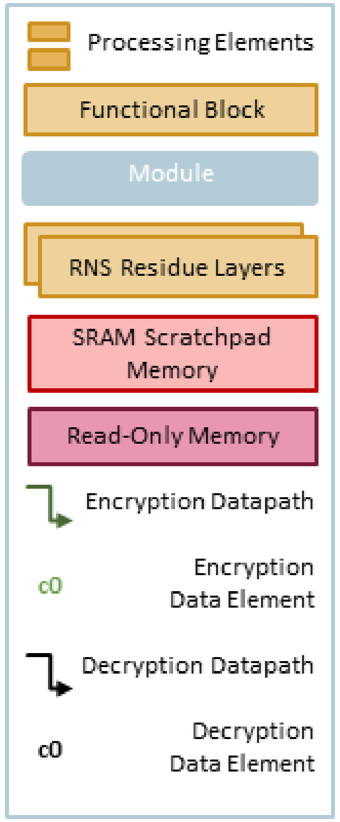
Conceptual Layers
Polynomial Parallelism

Modules & Functional Blocks
Specialized & Comprehensive

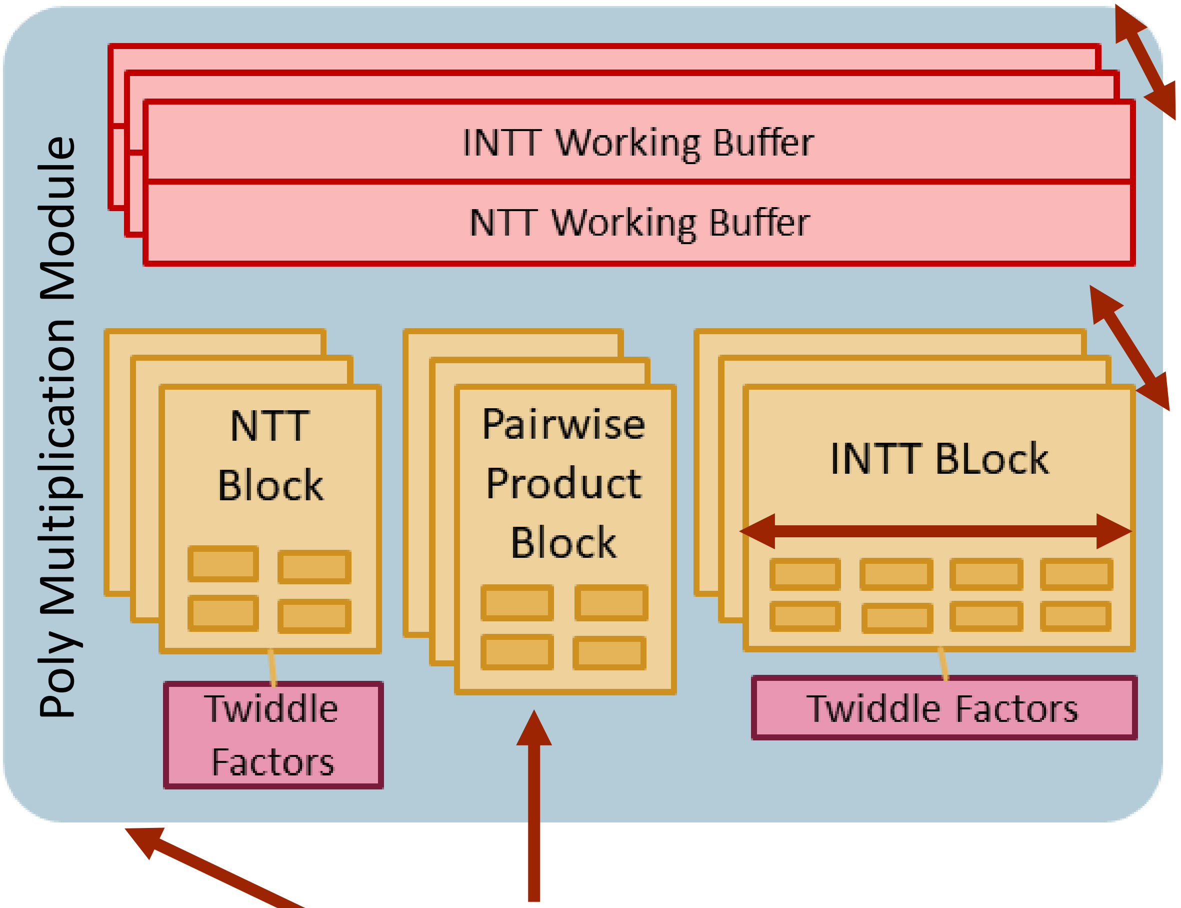


CHOCO-TACO Hardware Optimization





CHOCO-TACO Hardware Optimization

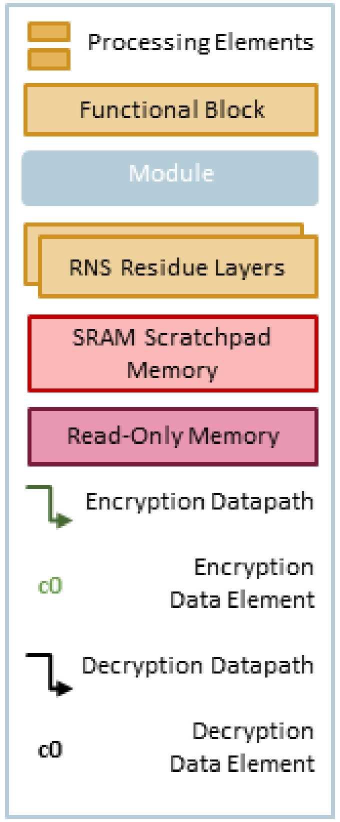


SRAM Buffers
Parallelism, Pipelining, Minimal Data Movement

Conceptual Layers
Polynomial Parallelism

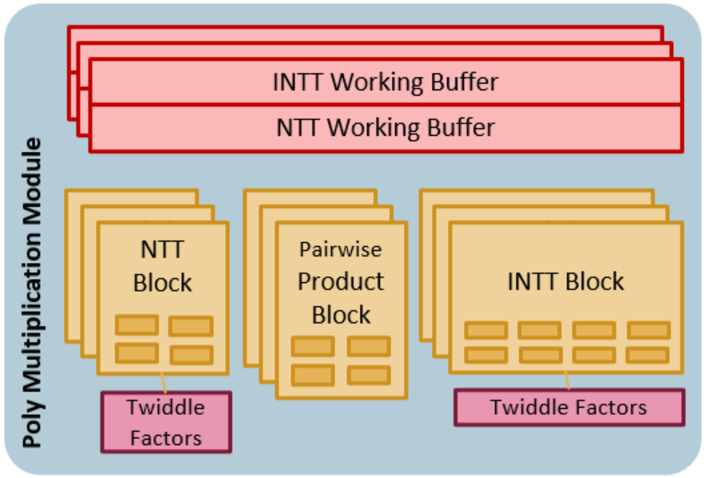
Processing Elements
Coefficient Parallelism

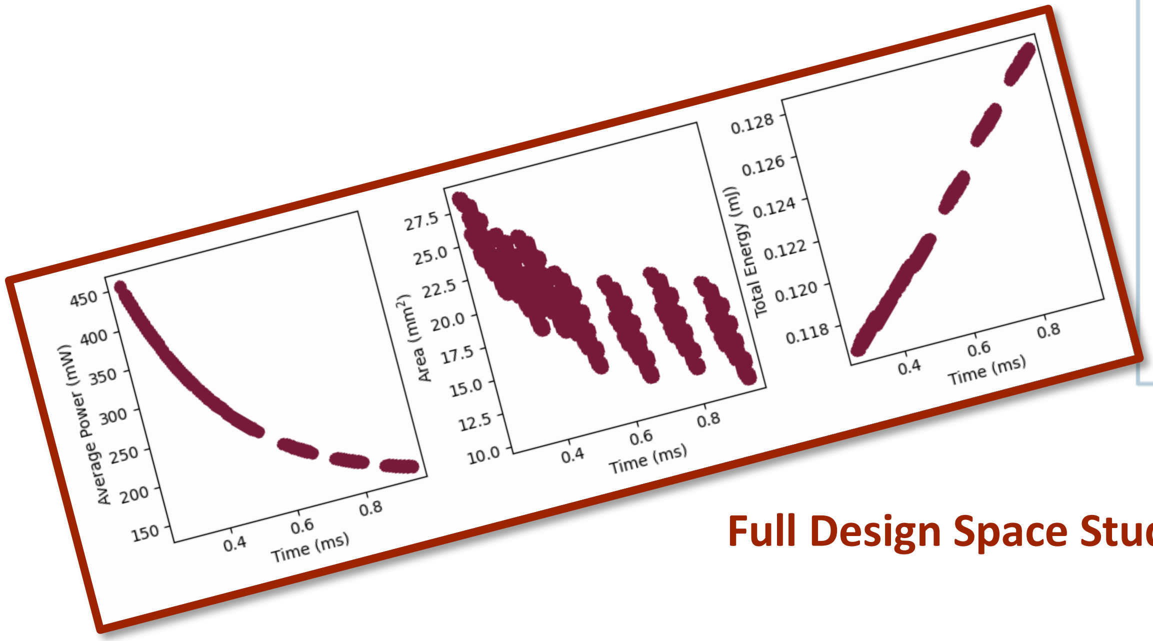
Modules & Functional Blocks
Specialized & Comprehensive

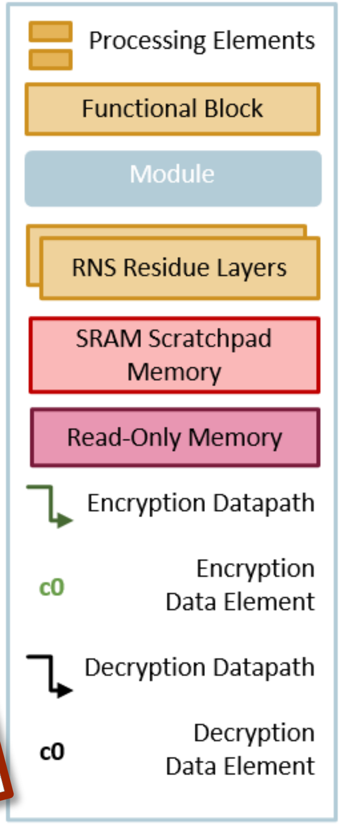


Electrical & Computer ENGINEERING

CHOCO-TACO Hardware Optimization

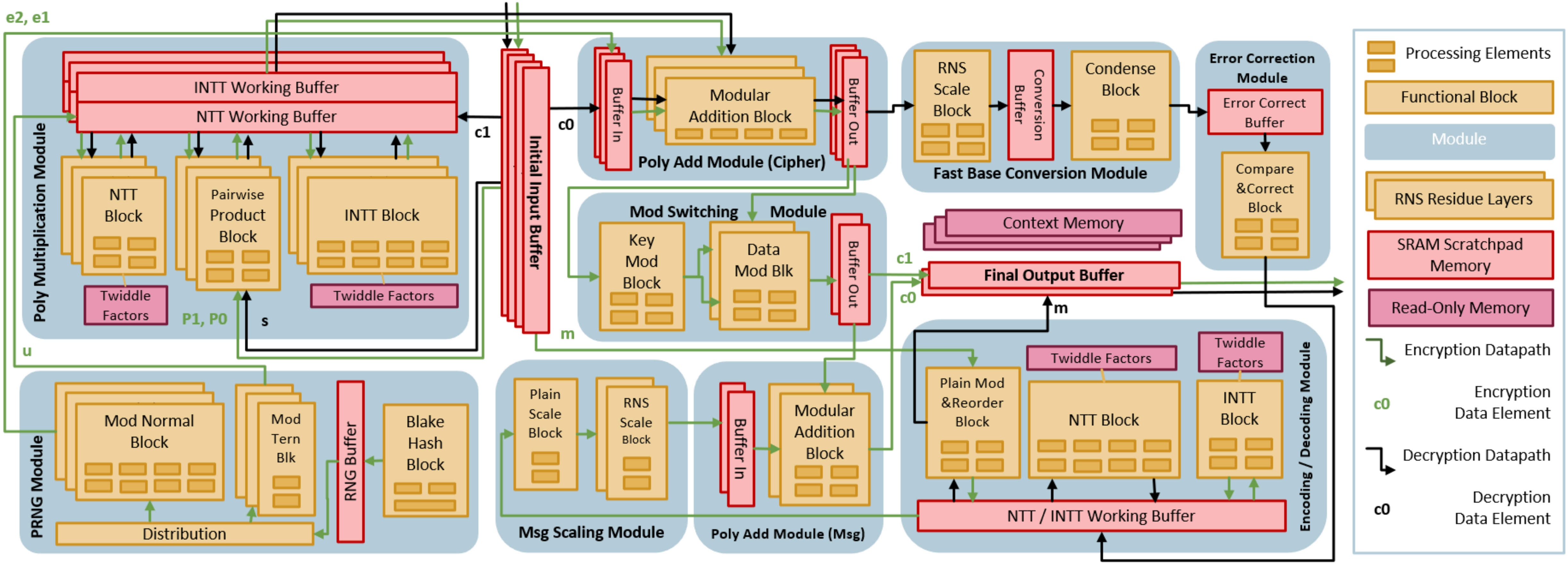






Full Design Space Study in Paper!

CHOCO-TACO Encryption & Decryption Hardware



19.3 mm^2 area. Consumes 200 mW power, .1228 mJ to perform a single encryption in .66 ms.

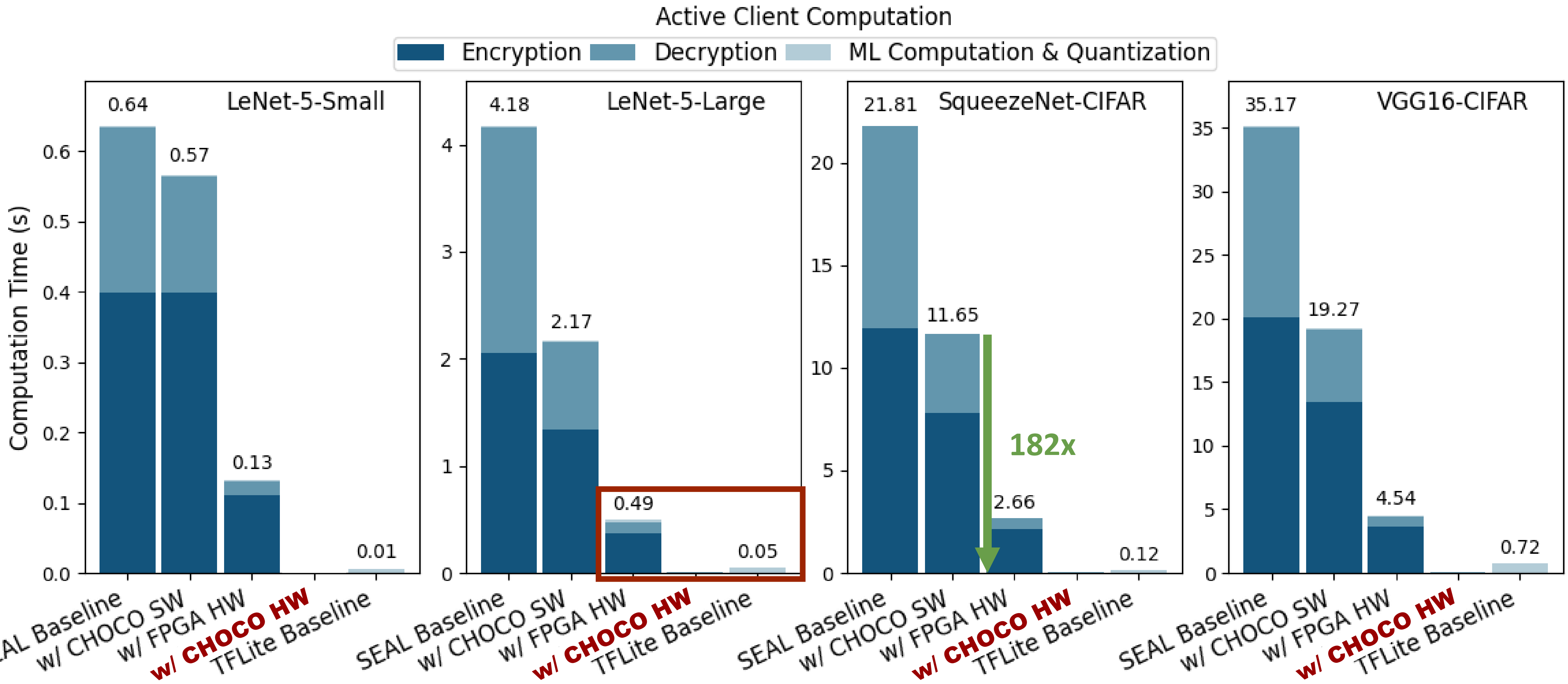
Outline

- Introduction
- Encrypted Computing Background
- **CHOCO System Implementation**
- **Algorithm Optimizations & Hardware Acceleration**
- Results & Conclusions

Outline

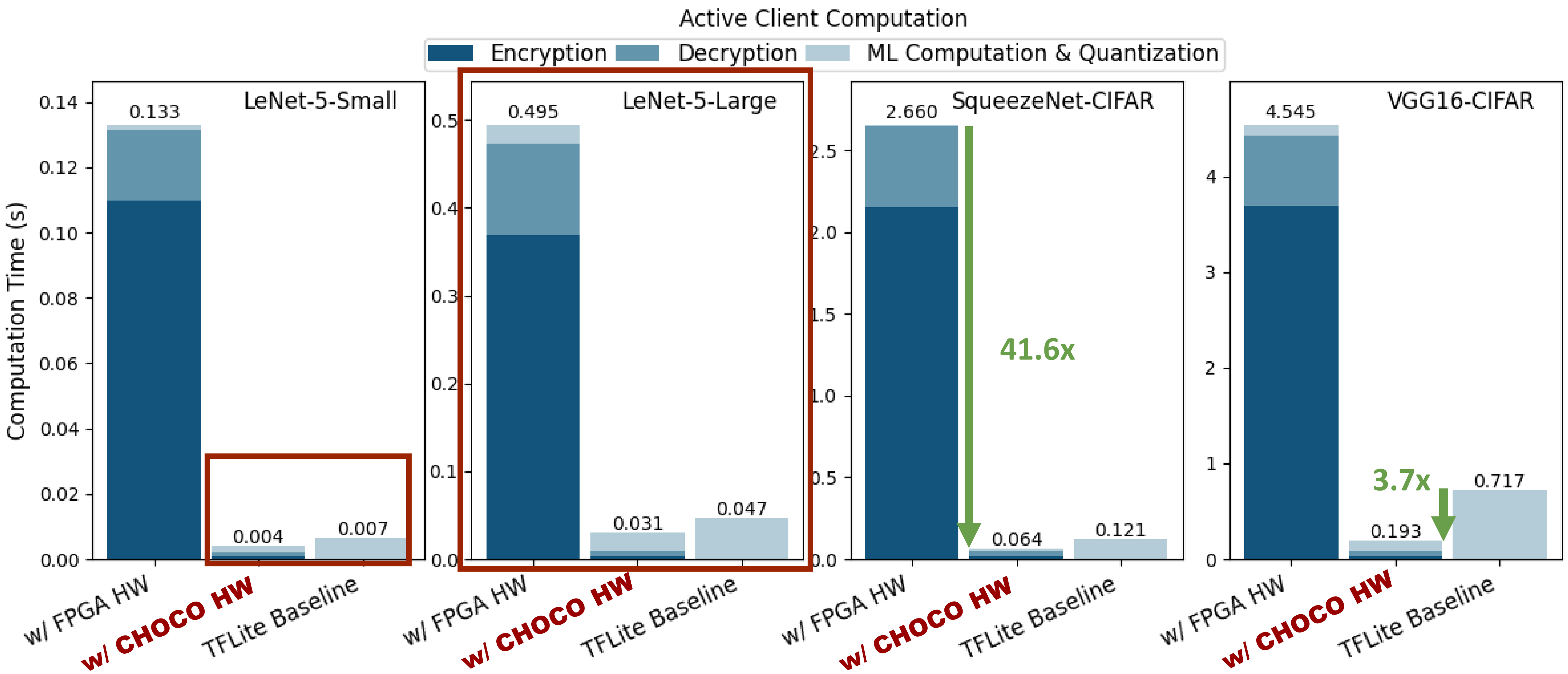
- Introduction
- Encrypted Computing Background
- CHOCO System Implementation
- **Results & Conclusions**
- **Active Client Computation, Communication & Applications**

CHOCO-TACO Accelerates Client Compute



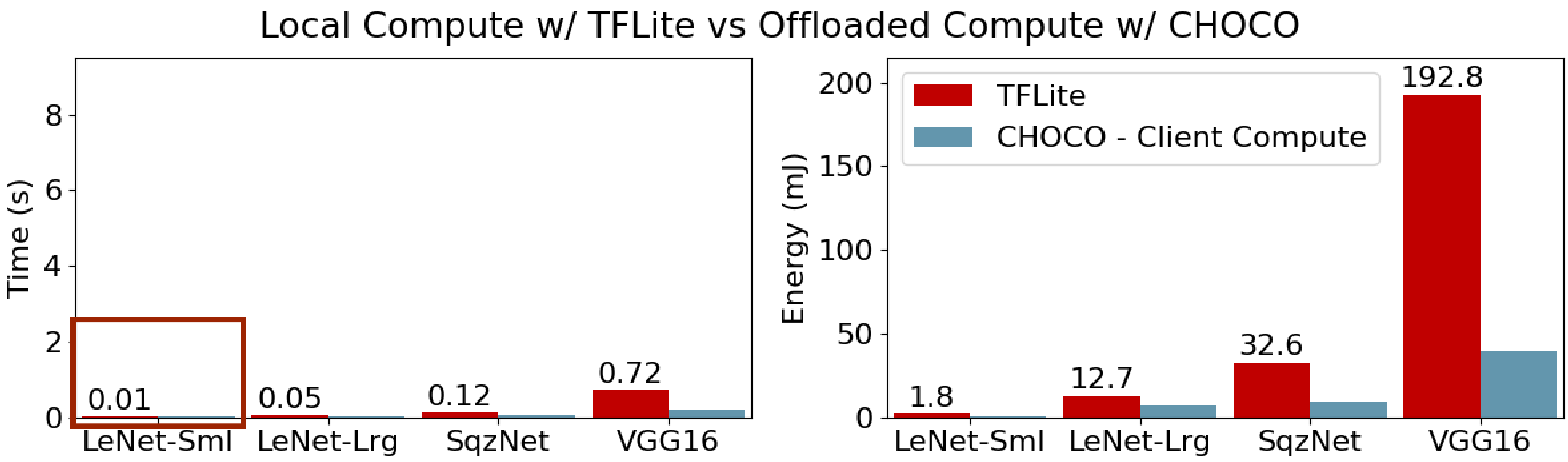
- **CHOCO HW** = CHOCO SW + CHOCO-TACO Encryption/Decryption Simulated ASIC
- Average **123.3x** Improvement over CHOCO software alone

CHOCO-TACO Accelerates Client Compute

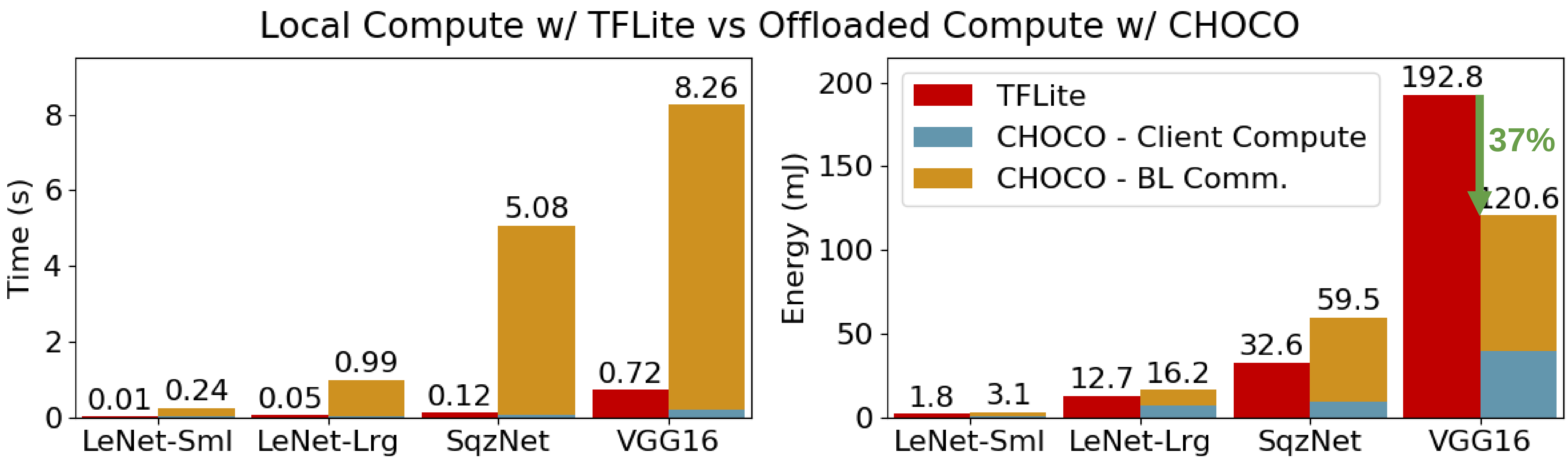


- Average **29x** better than FPGA accelerators
- Average **2.2x** better than local compute via

CHOCO Makes Client End-to-End Costs Feasible

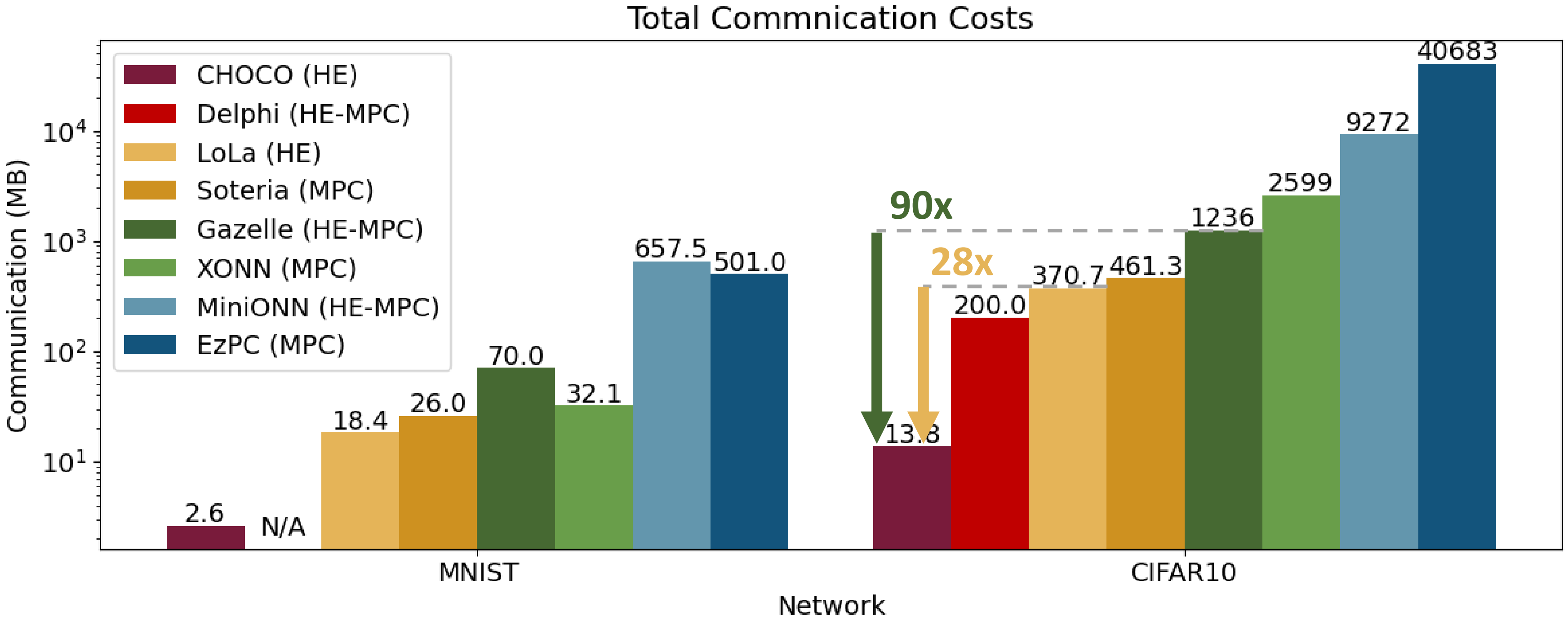


CHOCO Makes End-to-End Client Costs Feasible



- Privacy-Preserving Offload can be **Competitive with Local Compute**
- **37%** decrease in energy consumption for VGG16
- Up to **66% communication reduction** from SEAL baseline

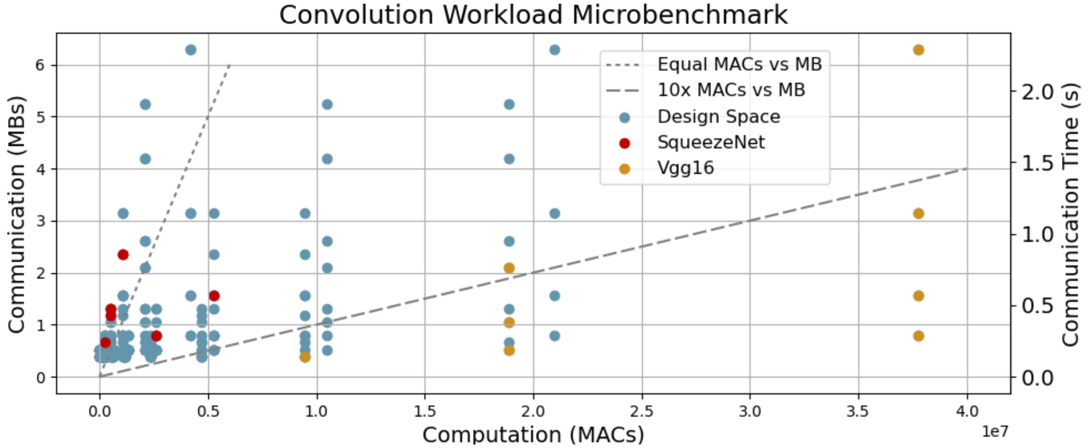
CHOCO Algorithms Reduce Communication



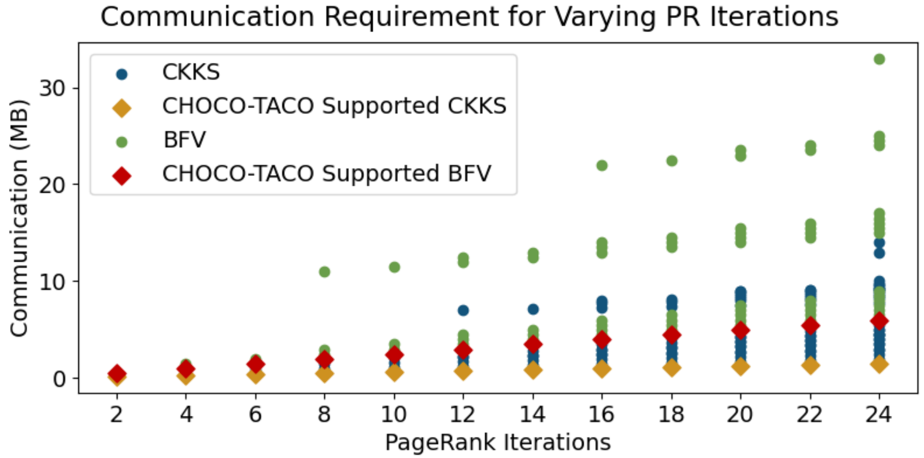
- Up to three orders of magnitude improvement in communication
- Nearly **90x** improvement over Gazelle [Juvekar `18]
- **28x** better than LoLa (*not* client-aided) [Brutzkus `19]

- C. Juvekar, V. Vaikuntanathan, and A. Chandrakasan. 2018. *GAZELLE: A Low Latency Framework for Secure Neural Network Inference.* In Proceedings of the 27th USENIX Conference on Security Symposium (Baltimore, MD, USA) (SEC'18). USENIX Association, USA, 1651–1668.
- Alon Brutzkus, Ran Gilad-Bachrach, and Oren Elisha. 2019. *Low Latency Privacy Preserving Inference*. In Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97), Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, Long Beach, California, USA, 812–821.

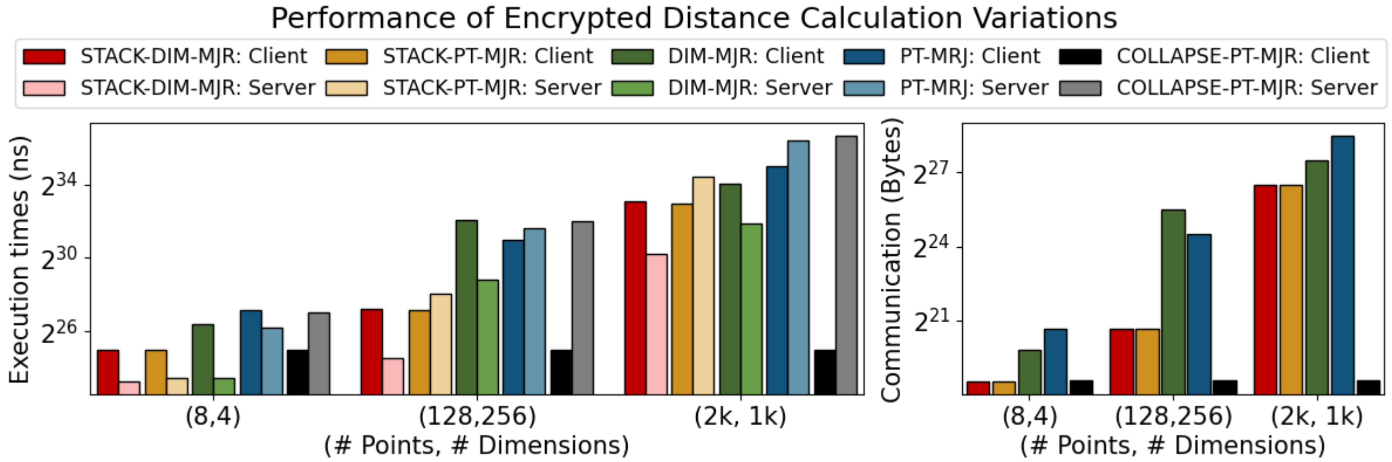
See Paper for More Applications & Results



Unmodified DNN Networks



PageRank



Encrypted Distance Calculations (K-Means & KNN)

Electrical & Computer ENGINEERING

Conclusions

- CHOCO motivates and prioritizes **client-aware optimizations**
- CHOCO alogorithm optimizations **reduce communication by orders of magnitude** over prior work
- CHOCO-TACO hardware **comprehensively accelerates client-side cryptographic primitives**
- CHOCO **enables participation from resource-constrained devices** in client-aided encrypted computation
- CHOCO makes client responsibility **competitive with local compute**
- CHOCO benefits **generalize to diverse applications**

Client-Optimized Algorithms & Acceleration for Encrypted Compute Offloading

Thank You! Questions?

McKenzie van der Hagen – mckenziv@andrew.cmu.edu

Brandon Lucia – blucia@andrew.cmu.edu