ESSCIRC 2023- IEEE 49th European Solid State Circuits Conference (ESSCIRC) | 979-8-3503-0420-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/ESSCIRC59616.2023.10268762

A 10.33 µJ/encryption Homomorphic Encryption Engine in 28nm CMOS with 4096-degree 109-bit Polynomials for Resource-Constrained IoT Clients

Siddharth Das*, McKenzie van der Hagen*, Swarali Patil, Cagri Erbagci, Brandon Lucia, Ken Mai
Dept. of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, USA
Email: {sdas2, mckenziv}@andrew.cmu.edu

***Abstract*—Homomorphic Encryption (HE) is used to protect sensitive client data during offloaded compute on a potentially untrusted server. Unfortunately, the computational intensity of HE operations quickly depletes the limited resources on IoT clients. Thus, we present an energy-efficient silicon implementation of encryption/decryption in the Brakerski-Fan Vercauteren HE scheme. To support meaningful applications, including several machine learning workloads, we optimize for fixed parameters N=4096 and log_2q=109 through pipelining, multi-level parallelism, and efficient memory accesses. At an energy-optimal operating point of 60MHz and 0.64V our chip, fabricated in a 28nm bulk planar CMOS process with an accelerator core area of 1.69 mm^2, consumes 10.33 µJ/encryption. Ultimately, this work enables IoT clients to participate in privacy-preserving offloaded compute using client-aided Homomorphic Encryption.**

***Index Terms*—Homomorphic Encryption, Number Theoretic Transform, Accelerator**

I. Introduction

Internet of Things (IoT) clients increasingly offload computation to servers due to shrinking energy budgets and demand for higher performance. Privacy concerns that this engenders can be addressed with Homomorphic Encryption (HE) which allows for direct computation on encrypted ciphertexts, without decryption (Fig. 1). Unfortunately, traditional HE functionality is limited to a finite number of linear arithmetic operations. In contrast, client-aided HE allows for arbitrary computation by periodically sending ciphertexts back to the client for decryption and re-encryption [1]. Resource-constrained clients, such as those in IoT systems, must perform these computationally intensive decrypt/re-encrypt operations repeatedly on the critical path. Thus, we present an energy-optimized ASIC accelerator for client-side HE encryption and decryption using the Brakerski-Fan-Vercauteren (BFV) HE scheme [2] [3]. Our accelerator achieves an 11.20x improvement in energy efficiency for realistically large HE parameters over a previously published design using scaled results [4]. To our knowledge, this is the first silicon implementation of client-side HE primitives with energy consumption appropriate for low-resource IoT devices while using HE parameters large enough to enable real multi-operation workloads including several machine learning models [1].

*Equally-Credited Authors

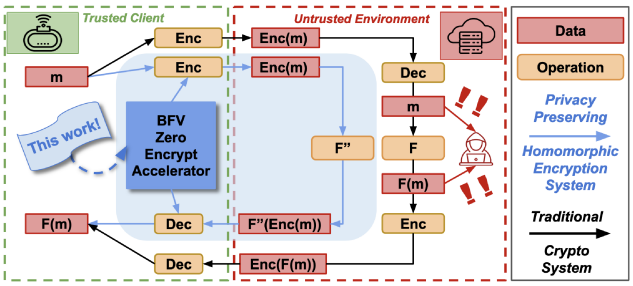


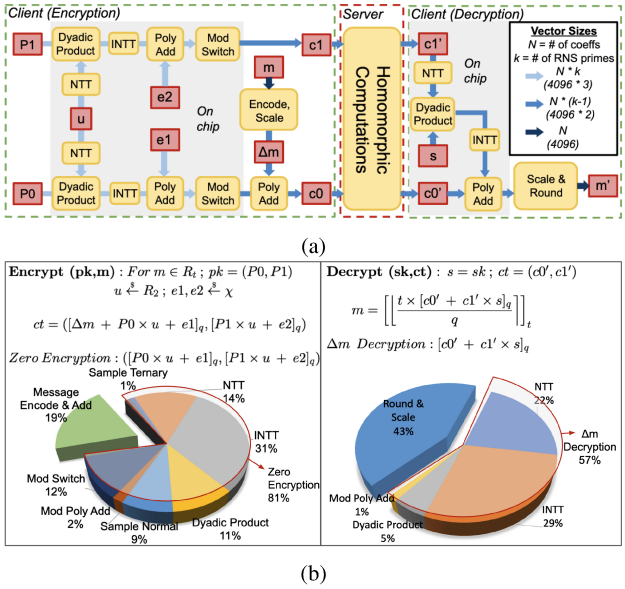
Fig. 1: Client-server interaction in an offloaded compute system with and without homomorphic encryption.

II. BFV Homomorphic Encryption

BFV is a second generation HE scheme with security properties based on the Ring Learning With Errors (RLWE) problem. The scheme includes key generation, encryption, decryption, and encrypted linear operations. Of these, the client performs the first three. While key generation happens only once, client-side encryption and decryption happen repeatedly within a single client-aided application, i.e. a single neural network inference [1]. Fig. 2(a) shows the arithmetic operations and dataflow of encryption and decryption. All operands and results are degree-N polynomials with coefficients in the field of the modulus q represented with k coprimes using the Residue Number System (RNS) [5]. While prior accelerators target infeasibly small N, q, and k values unfit for any meaningful HE application [4] [6] [7] [8], we target a parameter set of N=4096, log_2q=109, and k=3 to support real-world workloads [1]. Our chip implements Zero Encryption, the portion of encryption before the plaintext is added, which comprises 81% of the single-thread execution time on an ARM Cortex-A7 CPU (Fig. 2). It includes a polynomial multiplication, in the Number Theoretic Transform (NTT) domain, with the public keys and the addition of random error polynomials. With the resources on our custom ASIC, we can also support 57% of decryption, shown in Fig. 2 as Δm Decryption.

III. Architecture

We employ both pipelining and data parallelism for increased energy efficiency. Our design, shown in Fig. 3(a), has



(a)

(b)

Fig. 2: (a) BFV HE performs polynomial multiplication, in NTT form, with public keys followed by the addition of random error polynomials. Decryption includes multiplication with the private key. (b) Zero Encryption and Δm Decryption account for 81% and 57%, respectively, of the single-thread execution time on an ARM Cortex-A7 CPU

three parallel "layers," each corresponding to one of the k=3 RNS primes. Layers share global control signals, keeping them in lockstep. Datapaths and memories are duplicated across all layers to operate on independent polynomials.

A. Datapath

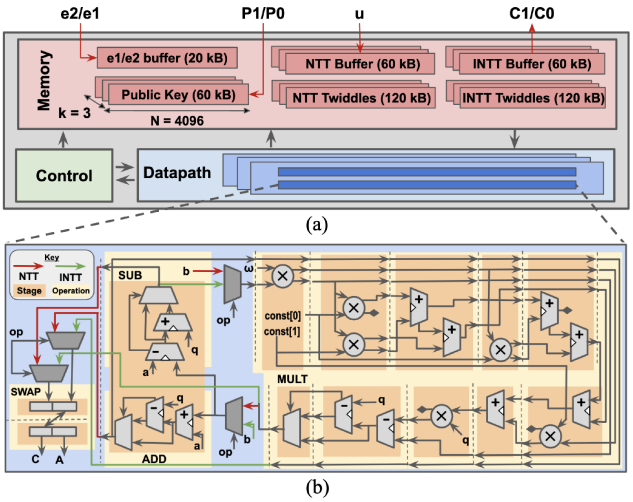
To accelerate polynomial operations, we use two parallel datapaths per layer. Each datapath operates simultaneously on two coefficients. The 13-stage pipelined datapaths support multiple 40-bit modular operations including modular addition/subtraction and modular multiplication using Barrett Reduction (Fig. 3(b)). These primitive coefficient operations are sufficient to support all polynomial operations including addition, multiplication and NTT/INTT operations.

B. Memories

The pipelines are fed from the twelve (four per layer) 1024x80b dual port SRAMs. Three additional (one per layer) 2048x80b single port SRAMs store the public keys, and one 2048x80b single port SRAM contains the HE error polynomials. Since our implementation has a fixed parameter set, the twiddle factors used by NTT/INTT do not change. Therefore they are stored in twelve (four per layer) 4096x40b ROMs.

C. NTT Optimization

Forward and Inverse Number Theoretic Transforms (NTT/INTT) are widely utilized to accelerate polynomial multiplication. Unfortunately, they remain the primary bottlenecks



(a)

(b)

Fig. 3: (a) Overall system architecture. (b) Pipelined datapath supporting all polynomial operations, including NTT (red) and INTT (green), composed of 40-bit modular operations (yellow), across 13 pipeline stages (orange).

in RLWE schemes. Together, they account for over 40% of the computation in BFV Encryption (Fig. 2(b)). Recognizing this, we design an efficient NTT for large-degree polynomials using an optimized memory access pattern. Fig. 4 shows this operation for four coefficients a, b, c, and d. We store two 40-bit interacting coefficients in the same memory location with a word size of 80 bits. In the first stage of the butterfly, a and b (c and d) interact to produce A and B (C and D). In the second stage of the butterfly, opposite pairs interact (A with C and B with D). To accommodate these future interactions a swap stage rearranges coefficients to store interacting 40-bit intermediate results in the same 80-bit memory word. Storing interacting values together allows them to be accessed with a single read/write operation and prevents unnecessary data movement across butterfly stages.

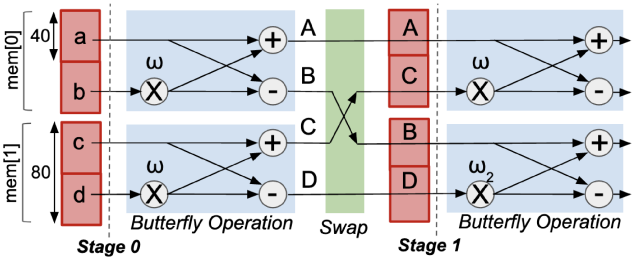


Fig. 4: NTT operation for four coefficients a, b, c and d. Interacting values for the next stage are swapped and stored together to be accessed with a single read/write operation

We use a Cooley-Tukey butterfly pattern and a Gentleman-Sande butterfly pattern for forward and inverse NTTs respectively (Fig. 5(a)). The control flow to support both of these operations within a single datapath is depicted in Fig. 3(b)

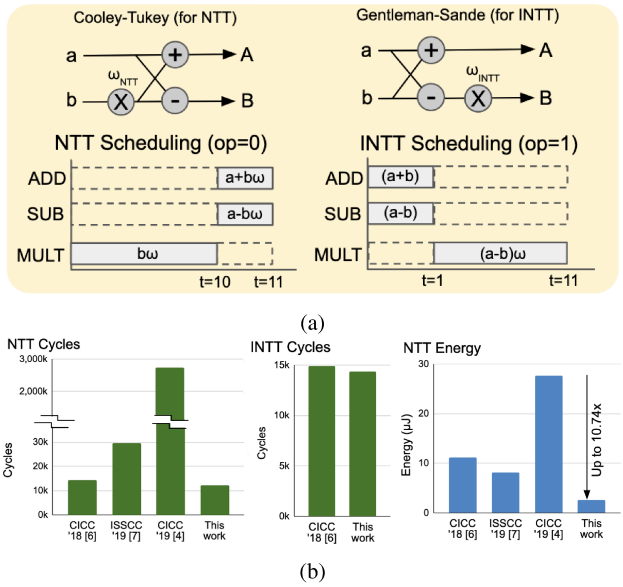


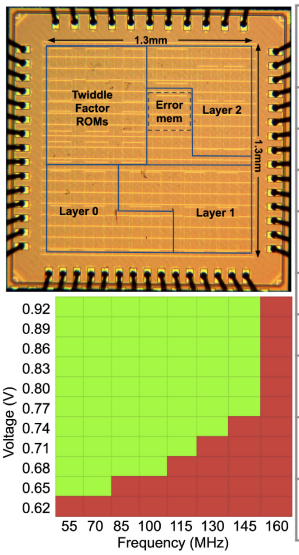
Fig. 5: (a) Dataflow and timing diagrams to support butterfly patterns for both forward and inverse NTT (b) Scaled cycle count and energy comparisons with previous implementations of NTT on hardware

using *op* select muxes. Corresponding timing diagrams are also included in Fig. 5(a).

Fig. 5(b) shows a cycle count per NTT operation comparison with prevailing literature using scaled parameters as described in the following sections. One previously published silicon NTT implementation explored a hierarchical architecture with local processing elements and vector processing for larger values of N, but with rapidly diminishing returns [6]. Other NTT architectures using only single-port SRAMs reduce area but require increased read and write operations performed per cycle [7]. An implementation using Bluestein NTT prioritizes functionality by accommodating non-power-of-2 polynomial transformations [4]. In contrast, performance-targeted designs with high levels of parallelism in the NTT core have also been explored, but they do not scale well to HE applications with large parameter sets [8]. Compared to these prior works, our design achieves up to a 10.74x energy reduction for a single NTT operation.

IV. Evaluation

Our prototype testchip was fabricated in a 28nm 9-metal bulk planar CMOS technology on a 2.56 mm^2 die with an accelerator core area of 1.69 mm^2 (Fig. 6). The on-die memories (200kB SRAM, 240kB ROM) occupy 48% of the core area and are enough to store all values for a single ciphertext component (*c1* or *c0* in Fig. 2(a)). Between computation for each component, the 28 I/O pads (out of 56 total) are utilized to exchange data off-chip via a custom handshake protocol. For decryption we use the existing encryption infrastructure while making necessary changes to the data inputs.



Technology	28nm 9-metal planar bulk CMOS Nominal Voltage: 0.9V
Supply Voltage	0.64 V to 1.1 V
Die size	1.6x1.6 mm^2
# of pads	56
Operating Frequency	0.5 MHz to 157 MHz
Memory	440 kB
Application	BFV Zero Encryption
Lattice Parameters	N: 4096 q: 109-bit modulus
Average Power	12.58 mW @ 0.64V, 60 MHz, room temp
Energy per encryption	10.33 μJ @ 0.64V, 60 MHz, room temp

Fig. 6: Chip micrograph for the 2.56 mm^2 die with specifications and shmoo plot

V. Results

Timing diagrams for encryption and decryption operations are shown in Fig. 7. At a max frequency of 157MHz, the chip performs a single Zero Encryption (Δm Decryption) in 314 (196) μs. The chip operates from 0.64V up to 1.1V, as shown by the Shmoo Plot in Fig. 6. At the empirically optimal energy point, it runs with a 0.64V supply voltage at 60 MHz and consumes 10.33 μJ/Zero Encryption and 6.45 μJ/Δm Decryption. This equates to 11.20x (4.58x) lower encryption (decryption) energy over scaled results for an existing silicon implementation of a BGV Homomorphic Encryption Accelerator [4] (Fig. 8(b)). To match our implementation and parameter set, we isolate these comparisons to only Zero Encryption and Δm Decryption and scale them using $Nlog_2\ N$ scaling and linear log_2q scaling. The BGV accelerator performs 3 NTT (1 INTT) operation per encryption (decryption), compared to the 1 NTT and 2 INTT (1 NTT and 1 INTT) performed by our BFV accelerator. We isolate the energy of an individual NTT operation and make further scaled comparisons with previous work (Fig. 5(b)). For the RLWE crypto accelerator supported by efficient NTT architectures [7], we use the same scaling factor of $Nlog_2\ N$. For the dedicated NTT accelerator ASIC [6], we use a factor of $N^{1.62}$, extrapolated from their reported results for N=256 and N=512. We observe that our NTT implementation consumes 10.74x lower energy compared to scaled results for that of a similar HE implementation [4]. Further comparisons are included in Table I, as well as an energy/power trend analysis in Fig. 8(a).

VI. Conclusion

Without considering real-world systems and applications, previous work often targets infeasibly small HE parameters that severely limit the depth and type of operations that

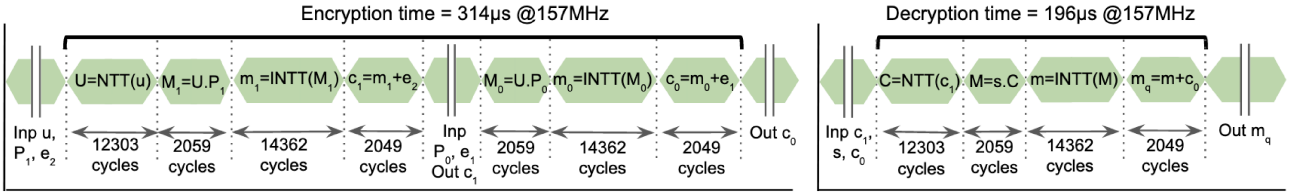


Fig. 7: Timing diagram of operations within encryption and decryption

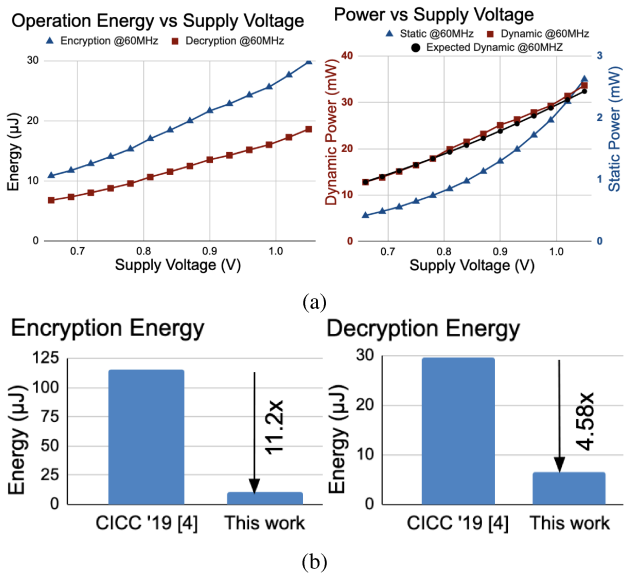


Fig. 8: (a) Energy and power plots vs supply voltage. Dynamic power follows the expected square dependency on supply voltage (b) Scaled encryption and decryption energy comparison with a previous implementation

TABLE I: Comparison

	CICC '18 [6]	ISSCC '19 [7][a]	CICC '19 [4]	ISSCC '22 [8]	This work
Technology	40nm	40nm	55nm	28nm	28nm
Supply Voltage (V)	0.9	0.68-1.1	0.28-1.2	0.9	0.64-1.1
Frequency (MHz)	300	12-72	0.01-60	500	0.5-157
Area (mm^2)	2.05	0.28	3.125	3.6	1.69
Supported Scheme	RLWE	RLWE	FHE (BGV)	RLWE	FHE (BFV)
log_2q	32	24	34	24	109[b]
Poly. Size	512	1024	16	256	4096[c]
NTT Performance					
NTT Cycles	492	6155	3560	32	12303
NTT Energy(µJ)	0.096	0.342	0.012	-	2.58
Encryption/Decryption Performance					
Encryption Cycles[d]	-	-	14240	-	49243
Encryption Energy(µJ)[d]	-	-	0.050	-	10.33
Decryption Cycles[e]	-	-	4041	-	30773
Decryption Energy(µJ)[e]	-	-	0.013	-	6.45

[a]Results partially obtained from [9]
[b]Modulus split into 3 RNS primes with size [36, 36, 37]
[c]Chosen for practical usability in IoT applications
[d]Results for zero encryption only
[e]Results for Δm Decryption only

can be performed. The fixed parameter set targeted by this work, N=4096 and log_2q=109, is the minimum parameter set that allows for meaningful multi-operation homomorphic computation on an encrypted ciphertext without exceeding the noise budget [1]. This work demonstrates that such a parameter set can be efficiently supported within the area and energy constraints of an IoT device. At 60MHz and a supply voltage of 0.64V, our 1.69 mm^2 accelerator core performs Zero Encryption (Δm Decryption) using 10.33 (6.45) µJ, reducing energy consumption by 11.20x (4.58x) over the closest comparable published design [4]. These improvements enable real-world client-aided HE applications with resource-constrained clients.

ACKNOWLEDGEMENTS

We would like to thank Apple's New Silicon Initiative for supporting this project through accessible academic tapeouts.

REFERENCES

[1] M. van der Hagen and B. Lucia, "Client-optimized algorithms and acceleration for encrypted compute offloading," in *ASPLOS '22*, 2022, p. 683–696.
[2] Z. Brakerski, "Fully homomorphic encryption without modulus switching from classical gapsvp," in *CRYPTO 2012 - Volume 7417*, p. 868–886.
[3] J. Fan and F. Vercauteren, "Somewhat practical fully homomorphic encryption," Cryptology ePrint Archive, Report 2012/144, 2012.
[4] I. Yoon, N. Cao, A. Amaravati, and A. Raychowdhury, "A 55nm 50nj/encode 13nj/decode homomorphic encryption crypto-engine for iot nodes to enable secure computation on encrypted data," in *CICC 2019*, pp. 1–4.
[5] N. Samardzic, A. Feldmann, A. Krastev, S. Devadas, R. Dreslinski, C. Peikert, and D. Sanchez, "F1: A fast and programmable accelerator for fully homomorphic encryption," in *MICRO 2021*, 2021, p. 238–252.
[6] S. Song, W. Tang, T. Chen, and Z. Zhang, "Leia: A 2.05mm2 140mw lattice encryption instruction accelerator in 40nm cmos," in *CICC 2018*, pp. 1–4.
[7] U. Banerjee, A. Pathak, and A. P. Chandrakasan, "2.3 an energy-efficient configurable lattice cryptography processor for the quantum-secure internet of things," in *ISSCC 2019*, Feb 2019, pp. 46–48.
[8] Y. Zhu, W. Zhu, M. Zhu, C. Li, C. Deng, C. Chen, S. Yin, S. Yin, S. Wei, and L. Liu, "A 28nm 48kops 3.4µj/op agile crypto-processor for post-quantum cryptography on multi-mathematical problems," in *ISSCC 2022*, Feb 2022, pp. 514–516.
[9] U. Banerjee, T. S. Ukyab, and A. P. Chandrakasan, "Sapphire: A configurable crypto-processor for post-quantum lattice-based protocols (extended version)," in *IACR Trans. On CHES*, Oct 2019, pp. 17–61.