15.4 A 28nm 68MOPS 0.18μJ/Op Paillier Homomorphic Encryption Processor with Bit-Serial Sparse Ciphertext Computing

Guiming Shi[1], Zhanhong Tan[1], Dapeng Cao[2], Jingwei Cai[1], Wuke Zhang[3], Yifu Wu[3], Kaisheng Ma[1]

[1]Tsinghua University, Beijing, China
[2]Xi'an JiaoTong University, Xi'an, China
[3]Polar Bear Tech, Xi'an, China

Cloud computing currently lies at the heart of tremendous emerging information applications, providing various reliable and high-performance services based on vast amounts of individual and organizational data. Paillier homomorphic encryption (PHE) [6] is one of the critical privacy computing techniques that enables the ciphertext to enjoy equivalent utility as the plaintext. In the PHE scheme shown on the upper left of Fig. 15.4.1, the client encrypts its plaintext into ciphertext and then sends it to the server, which performs the homomorphic evaluation and returns the encrypted results to the client. However, this process avoids the delivery of plaintext at the cost of several computing burdens, and we list three dominant challenges in the rest of Fig. 15.4.1. First, the ciphertext domain calculation needs expensive large integer modular arithmetic operations (e.g., modular multiplication (ModMul), modular inversion (ModInv), and modular exponentiation (ModExp)) with several orders of magnitude higher energy and latency. Second, the client performs encryption and decryption featuring independent vector operations, while the server is expected to perform evaluations with multiply-and-accumulation (MAC) operations. Third, the diversity of tasks requires computing scalability to meet the latency and throughput demands in the cloud.

This paper presents a high-performance Paillier homomorphic encryption processor unit (PH-EPU) that includes: 1) two efficient modular processing units (i.e., Montgomery unit and Stein unit) with dynamic bit-width for basic Paillier homomorphic operations; 2) efficient ciphertext processing elements featuring a reconfigurable dataflow using bit-serial computing for different task patterns; 3) a scalable computing flow using instruction-based control for various workload volumes. This 28nm processor consumes 42.96mm^2 and operates at 500MHz and 0.9V, which supports encryption, decryption, and homomorphic evaluation of Paillier. Our PH-EPU achieves up to 14.9× speedup compared to a desktop CPU Intel Core i9-9900 with 16 cores, and our system-level PCIe card with 8 chips takes up to 22.8× lower latency than a server CPU Intel Xeon Platinum 8260M with 192 cores.

The overall architecture shown in Fig. 15.4.2 comprises a top controller, a global memory system with three buffers, and 16 ciphertext processing elements with bit-serial reconfigurable dataflow (BSRD-PE). The top controller manages the data interaction between the input/output buffers and 16 BSRD-PEs via a 256-bit bus. Each BSRD-PE consists of a weight buffer of 9KB, an input buffer of 8KB, an output buffer of 1KB, a look-up-table of partial-sum terms for bit-serial processing, a Montgomery unit (MU), and a Stein unit (SU). The MU performs the Montgomery modular multiplication (Mont) that can be organized as ModMul and ModExp operations for the Paillier algorithm. The SU leverages the Stein modular inversion algorithm (Stein) for the ModInv operation. These two primitive execution units solve the first challenge of modular operations via specialized arithmetic hardware. However, such common ModExp dataflow is inefficient for performing the dependent MACs in the evaluation stage, so we further extend a plug-in partial-sum (Psum) term look-up-table (LUT) coupled with a bit-1 selector to leverage the bit-level sparsity. The overall data path helps to process diverse computing patterns in the second challenge with high efficiency. The scale-out computing for the third challenge is realized by an instruction-based control flow combined with an automatic scheduling flow at the system level.

Figure 15.4.3 describes the implementation of MU and SU in detail. The MU mainly comprises three 256-bit multipliers for the compute-intensive Mont. The two-stage Montgomery operation has several similar iterations, the number of which lies on the width of multiplier (B). We evaluate four alternatives of B and select the 256-bit counterpart as it obtains the highest cost-efficiency. Therefore, based on the 256-bit operands, the first stage is to update the q-value, a dependent intermediate variable for the second stage. The second stage executes large-integer multiplications (4096b×256b) for the intermediate Montgomery results, where it takes 16 cycles using the 256-bit multipliers. The SU contains a local register file, a lightweight branch detector (BD), and execution units (EU) for the control-intensive Stein. The BD is only fed with the critical bits for branch detection to eliminate redundant fan-out, while the EU contains three 4102b adders and one shifter unit to perform the operations of each branch in one cycle. To avoid the bubbles coming from the branch detection, we further forward the updated results before the pipeline registers to the BD stage in each loop, which guarantees the utilization of SU.

Figure 15.4.4 depicts the dataflow inside the BSRD-PE for two typical computing patterns of Paillier: a 4-stage bit-serial mode for dependent MAC operations and a classical pipeline mode for regular modular operations. The dataflow between the two modes is reconfigured by the PE controller, reusing the same set of resources. Mode-1 employs the bit sparsity of weights via the bit-1 selector to improve the performance. In step 1, we decouple the partial sum (Psum) into eight terms corresponding to the accumulated results based on eight bits of weights. For each weight, the selector searches for the positions of bit-1 to indicate the update of the corresponding Psum terms. After finishing all the weights, since the native Psum data after step 1 does not consider the magnitude, the Psum terms are aligned in step 2 by performing Mont several times according to their bit positions. Step 3 further processes the negative sign term of Psum (i.e., $Psum_7$) through Stein once and Mont twice. Step 4 reduces all the Psum terms and gets the final result. Mode-2 is reconfigured to carry out the basic modular operations of Paillier, where ModMul and ModExp are implemented by repeatedly invoking the MU while SU implements the ModInv. In the case of ModExp, the corresponding weight is up to 4096b, so a fast exponentiation algorithm is introduced by multiplexing the bit-1 selector of mode-1 to skip the sparse bits in the same way.

Figure 15.4.5 shows an efficient task deployment flow across PEs and chips. The compiler executes task assignment, memory allocation, and instruction generation to generate the runtime and instructions for a full-height-full-length (FHFL) PCIe card integrated with a host FPGA and 8 PH-EPU chips. Typically for the convolution, since the bandwidth requirement of ciphertext is much larger than the plaintext weight, the task assignment at the chip level is prioritized in the output channel to broadcast the ciphertext and save bandwidth. For the PE level, we employ a search-optimized partition scheme to decide the parallelism of height ($Para_H$), width ($Para_W$), and channel ($Para_K$) of the 3D output feature maps across 16 PEs. We also establish a system-level experiment flow to simulate the complete process between the client and the server. The runtime library loads the input data and execution instructions to the cards through host CPUs. To demonstrate the scalability of our system, we evaluate a series of tasks with different workload volumes. The end-to-end evaluation results, including both software and hardware time except for the communication time between client and server, show that our PH-EPU chip and 8-chip card outperform the desktop and server CPUs significantly and present better scalability.

Figure 15.4.6 firstly demonstrates the efficiency of the critical primitive operations (i.e., Mont and ModInv) and the bit-serial optimization. The energy efficiency of our accelerator exceeds that of the CPU by several orders of magnitude. As for the bit-sparsity efficiency, we evaluate four data patterns of different sparsity rates. Results show an effective utilization of the zero-bit skipping with up to 3.3× speedup in 75% sparsity. This figure also shows the comparison table with prior work on custom hardware design for homomorphic encryption. Our PH-EPU supports all types of HE tasks (encryption, decryption, and evaluation) with high flexibility. It also supports dynamic bit-width for the ciphertext and weights while leveraging bit-serial sparsity to optimize the performance. The throughput of typical operations in Paillier achieves 23~68MOPS for Mont and 38KOPS for ModInv with 0.18~0.52μJ/Op and 105.3μJ/Op efficiency, respectively. Compared with a current Paillier processor [4], 115×~340× better throughput and 30.4×~87.8× improved energy efficiency are obtained for Mont. Figure 15.4.7 shows the chip micrograph and specifications for both the chip and PCIe card.

Acknowledgement:
This work was completed during the internship in the Polar Bear Tech. The corresponding author is Kaisheng Ma.

References:

[1] S. Song et al., "LEIA: A 2.05mm^2 140mW Lattice Encryption Instruction Accelerator in 40nm CMOS," *IEEE CICC*, 2018.
[2] I. Yoon et al., "A 55nm 50nJ/encode 13nJ/decode Homomorphic Encryption Crypto-Engine for IoT Nodes to Enable Secure Computation on Encrypted Data," *IEEE CICC*, 2019.
[3] M. S. Riazi et al., "HEAX: An Architecture for Computing on Encrypted Data," *ACM ASPLOS*, pp. 1295-1309, 2020.
[4] M. Bahadori et al., "A Programmable SoC-Based Accelerator for Privacy-Enhancing Technologies and Functional Encryption," *IEEE TVLSI*, vol. 28, no. 10, pp. 2182-2195, 2020.
[5] Y. Zhu et al., "A 28nm 48KOPS 3.4μJ/Op Agile Crypto-Processor for Post-Quantum Cryptography on Multi-Mathematical Problems," *ISSCC*, pp. 514-515, 2022.
[6] P. Paillier, "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes," *EUROCRYPT*, vol. 1592, pp. 223-238, 1999.

2023 IEEE International Solid-State Circuits Conference (ISSCC) | 978-1-6654-9016-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/ISSCC42615.2023.10067522

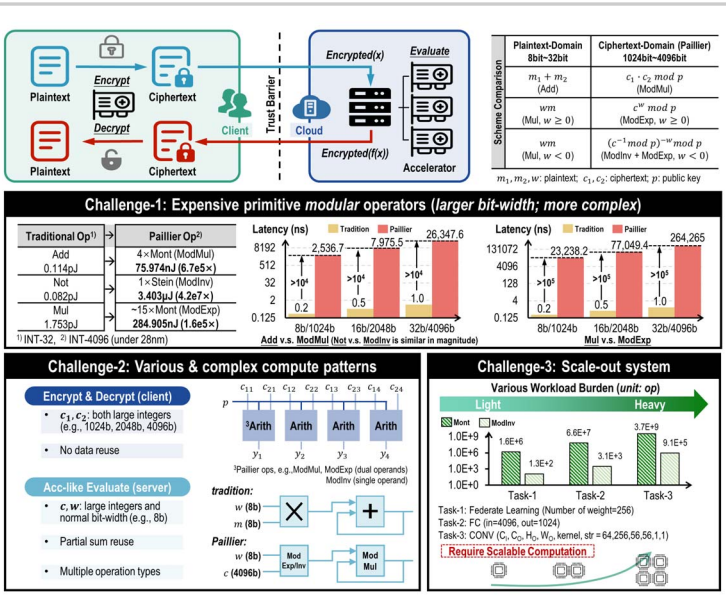


Figure 15.4.1: Paillier homomorphic encryption (PHE) challenges.

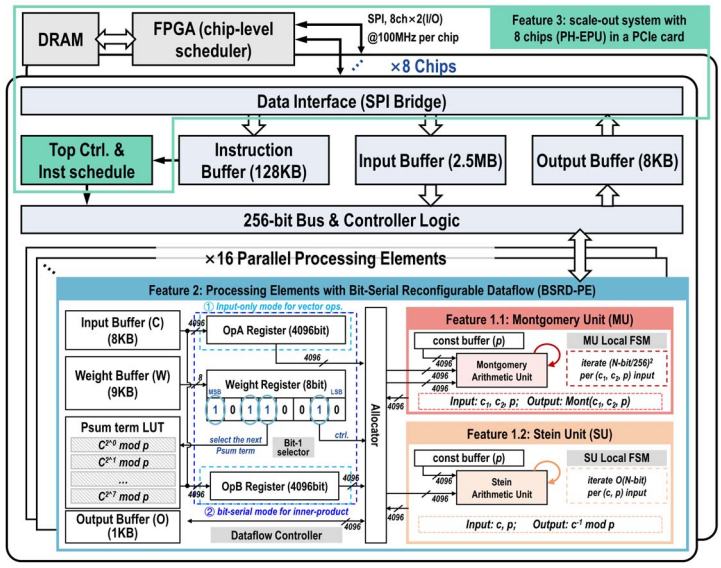


Figure 15.4.2: System architecture of the Paillier homomorphic encryption accelerator (PH-EPU).

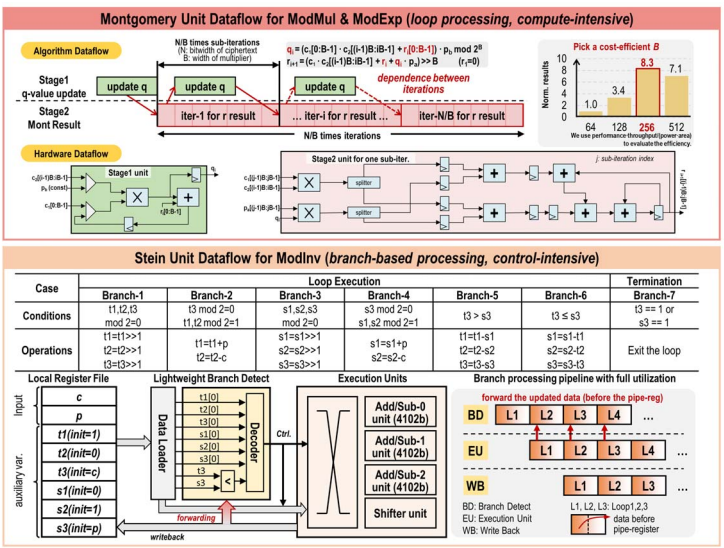


Figure 15.4.3: Illustration of the Montgomery unit (MU) and the Stein unit (SU).

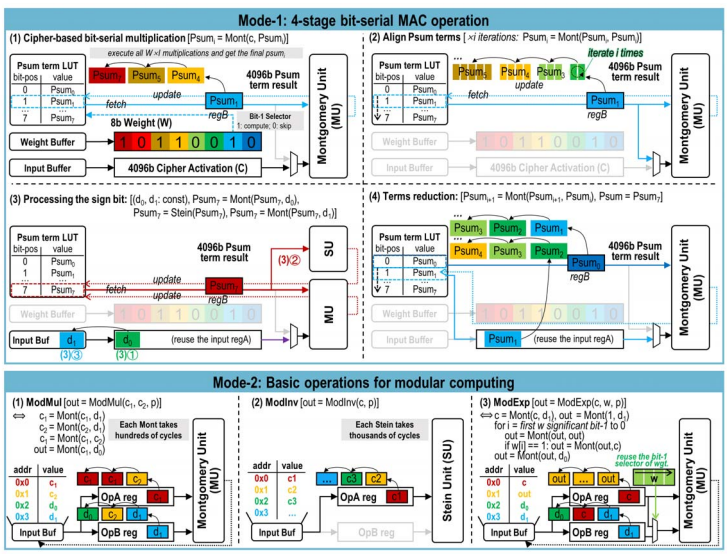


Figure 15.4.4: Details of the reconfigurable dataflow to enable bit-serial MACs and basic modular operations.

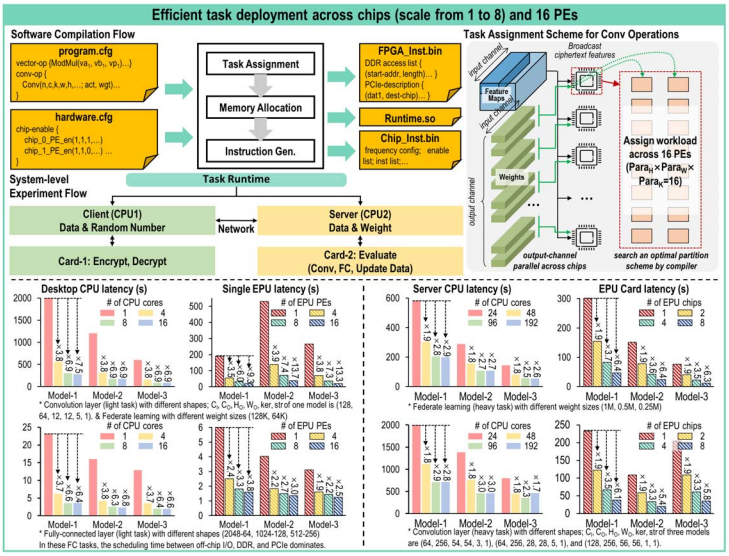
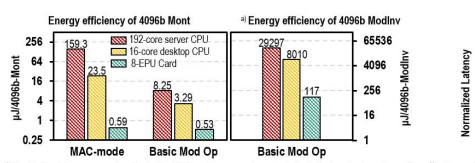


Figure 15.4.5: Task deployment flow and performance comparison with software implementations.

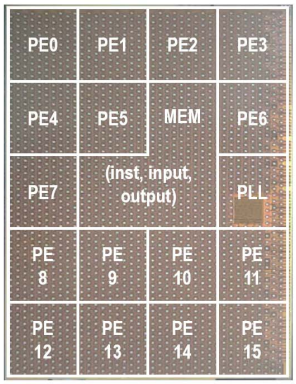


a) The ModInv takes very small portion in the MAC mode, so we only evaluate the typical vector-based operations. b) Latency without the cost of off-chip I/O.

	CICC 2018 [1]	CICC 2019 [2]	ASPLOS 2020 [3]	TVLSI 2020 [4]	ISSCC 2022 [5]	This Work
Technology	40nm	55nm	20nm (FPGA)	28nm (FPGA)	28nm	28nm
Chip Area	2.05mm²	3.13mm²	-	-	3.60mm²	42.96mm²
Frequency	300MHz	60MHz	275MHz	122MHz	500MHz	500MHz
Voltage	0.90V	1.20V	1.21V	1.00V	0.90V	0.90V
Power Consumption	216.5mW	4.8mW	-	3.2W	368mW	4W/12W[1]
Functionality						
HE Scheme/ Mathematical Problem	Lattice (MP)	BGV (HE)	CKKS (HE)	Paillier (HE)	Lattice (MP)	Paillier (HE)
Supported HE Tasks	Encrypt, Decrypt	Encrypt, Decrypt	Evaluate	Encrypt, Decrypt, Evaluate	Encrypt, Decrypt	Encrypt, Decrypt, Evaluate
Sparsity Type	-	-	-	Bit sparsity		Bit serial sparsity for dynamic weight bit-width
Paillier Scheme						
Throughput (4096b) Mont	-	-	-	0.2[3] MOPS	-	68[2]/23[3] MOPS[4]
Throughput (4096b) ModInv	-	-	-	-	-	38[3] KOPS[4]
Energy eff. (4096b) Mont	-	-	-	15.8[3] μJ/Op	-	0.18[2]/0.52[3] μJ/Op
Energy eff. (4096b) ModInv	-	-	-	-	-	105.3[3] μJ/Op

1) 4W of basic ModInv Op and 12W of others 2) Best throughput or energy efficiency of MAC-mode 3) Throughput or energy efficiency of basic Mod Op
4) The reported results of the Mont/ModInv throughput include the cost of off-chip I/O, DDR, and PCIe.
*The performance of basic Mod Mont is tested using ModExp.

Figure 15.4.6: The energy/bit-sparsity efficiency summaries and comparison table.



Items	Chip Specifications	
Technology	UMC CMOS 28nm HPC+ 1P8M	
Die Area	42.96mm²	
SRAM	2968 KB	
Standard Supply Voltage	Core	0.81V ~ 0.99V
	IO	1.8V
	SRAM	0.81V ~ 0.99V
Frequency	0.5MHz~500MHz	
Data Width	Ciphertext: 4096/2048/1024b	Weight: 2/4/8b (MAC-mode) 256b to 4096b (Basic-mode)
Power Consumption	4W~12W @0.9V, 500MHz[1]	
Maximal Performance	68[2]/23[3] MOPS[4]	
HE Scheme	Paillier	

Items	PCIe Card Specifications
Chip Number	8 chips
Power Supply	120W
Maximal Performance	355[2]/181[3] MOPS[4]
DDR3 Bandwidth	100Gb/s
Off-chip Bandwidth	12.5Gb/s
PCIe Bandwidth	32Gb/s

[1] Not include the external memory access; [2] MAC-mode; [3] Basic Mod Op;
[4] Performance of 4096b Mont (considering the cost of off-chip I/O, DDR and PCIe);
* All evaluation is under 25°C room temperature.

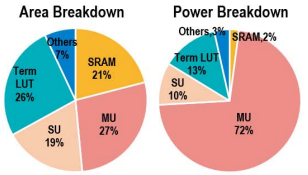


Figure 15.4.7: Chip micrograph combined with specification tables for the chip and the 8-chip PCIe card.

978-1-6654-9016-0/23/$31.00