2022 IEEE International Electron Devices Meeting (IEDM) | 978-1-6654-8959-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/IEDM45625.2022.10019409

First Demonstration of Homomorphic Encryption using Multi-Functional RRAM Arrays with a Novel Noise-Modulation Scheme

Xueqi Li[1], Bin Gao[1]*, Bohan Lin[1], Ruihua Yu[1], Han Zhao[1], Ze Wang[1], Qi Qin[1], Jianshi Tang[1], Qingtian Zhang[1], Xinyi Li[1], Zhenqi Hao[1], Xiaotao Li[2], Dequn Kong[2], Liqiu Ma[2], Ning Deng[1], He Qian[1], and Huaqiang Wu[1]*

[1]School of Integrated Circuits, BNRist, Tsinghua University, Beijing, China.
[2]Department of Internet of Things Technology and Application, China Mobile Research Institute, Beijing 100053, China.
*Email: gaob1@tsinghua.edu.cn; *Email: wuhq@tsinghua.edu.cn

Abstract— Homomorphic encryption (HE) is an encryption technology of which encryption and decryption process can be summarized as polynomials modulo multiplication computing with noise. In this paper, HE is firstly implemented on resistive random-access memory (RRAM) arrays, which are utilized as both matrix-vector multiplication (MVM) units and true random number generators (TRNG). Both high stability and good randomness are achieved for MVM and TRNG, respectively, by using different forming schemes, so that two distinct functions can be realized using the same device. Furthermore, the encryption-decryption process for privacy-preserving cloud computing is experimentally implemented on a hardware system with eight 144Kb RRAM arrays. For the whole RRAM array-based encryption-decryption process, small accuracy losses of 0.73% (for SVM) and 1.9% (for CNN) are achieved. This is the first demonstration of encryption computing acceleration with emerging device technology.

I. Introduction

Nowadays, information security and privacy protection are raising more and more concerns. Traditional encryption technologies allow data to be stored and transmitted in encrypted form; however, performing even simple analytics on the encrypted data is impossible. Homomorphic encryption (HE) is an important and mainstream encryption method that allows any third party to perform computation on the encrypted data without decrypting it (**Fig. 1**). This special property makes HE applicable in many fields, such as blockchain, federated learning, and multi-party computation (**Fig. 1**).

A key challenge that hinders HE from a wide range of practical applications is low efficiency. In brief, to encrypt or decrypt a batch of small integers, modulo multiplication of high-order large modulus polynomials are calculated, and random 'error' polynomials are required [1], which cause computational burden and large energy consumption on edge devices **(Fig. 1)** [2]. To accelerate the polynomial modulo multiplication process, a widely adopted approach is to use the number theoretic transform (NTT) and inverse NTT (INTT). Compared with the NTT, which does not need to be performed on each operand and the coefficients of the polynomials are either 0 or 1, the INTT are performed more often and the coefficients of the polynomials are up to tens of bits, thus, INTT need much more computation resources (**Fig. 2**).

Using emerging devices to process HE computing can solve the above issues. RRAM array is ideal for accelerating and reducing energy consumption of HE because RRAM has both computational capability and intrinsic randomness characteristics. However, the reliability required for computation and the randomness required for TRNG are somewhat contradictory: it is challenging to modulate RRAM device to exhibit certain characteristics and make it meet the requirements of HE process. In addition, the excessive parameters required for HE also place demand on the integration scale and array size of RRAM devices.

In this paper, we experimentally implement homomorphic encryption on RRAM arrays, using it as the MVM unit and TRNG at the same time by applying different forming schemes. To reduce the MVM error caused by intrinsic stochasticity in RRAM devices, we use a so-called RNS-wise (residue number system) mapping method. Two privacy-preserving cloud computing tasks are demonstrated. The RRAM array-based encryption-decryption process for cloud SVM inference is implemented on a fully-hardware system with eight 144Kb RRAM arrays.

II. Homomorphic Encryption

The flow chart of HE is illustrated in **Fig. 3**. Key generation is the initialization process before encryption, that only needs to be performed once. The encryption and decryption process can be summarized as calculating $\boldsymbol{a} \times \boldsymbol{s} + \boldsymbol{e}$, while '$\boldsymbol{a}$' and '$\boldsymbol{s}$' are both polynomials on different rings, '$\boldsymbol{e}$' refers to an error polynomial whose coefficients are consistent with a Gaussian distribution. Since three error polynomials are required for encrypt one message, random number generation accounts for a large overhead in hardware implemented HE scheme.

III. Implementation using RRAM array

A. Test platform and characteristics of RRAM device

A 144K-bit 2T2R RRAM array platform is fabricated with a material stack of TiN/TEL/HfO_2/TiN, in which TEL is the thermal enhanced layer (TEL) to improve the analog switching characteristics, and HfO_2 serves as the resistive switching layer [5]. The DC I-V curve of RRAM device shows good cycle-to-cycle uniformity (**Fig. 5**). **Fig. 6** illustrates an HRS/LRS ratio of ~8.3×, which is large enough for our HE demonstration.

B. RRAM-based MVM unit

The INTT algorithm is essentially a matrix-vector multiplication (MVM), and the transform matrix remains unchanged, so it can be naturally implemented on RRAM array for acceleration. RRAM MVM unit is based on analog computing, which means that the weight conductance is written as a number in an interval rather than a definite value, so the number of rows opened at the same time directly affects the bit error rate (BER) of the computation. Also, the ADC precision determines the range in which the device conductance is limited, which greatly affects the BER. In our implementation, a 3-bits ADC is used to get higher energy efficiency, and the relationship between the number of open rows and BER is illustrated in **Fig. 7**. As we can see, when opening more than or equal to 3 rows simultaneously, the resistance distribution between different levels begin to overlap and the BER increases dramatically.

Unlike most researches that use RRAM MVM for low bit (or low precision) calculation, our demonstration shows the possibility of RRAM MVM to do large number calculation (up to 29 bits). For digital calculations, large numbers are usually expanded bit-wise and calculated; however, for analog calculation, a small error in the high-bit multiplied by a weighting factor will invalidate all results in the low-bit. Here, we use the so-called RNS-wise expansion method [6] which can self-detect errors. Experiment results show that after 7th write cycles, only 1 result out of 128 is wrong (**Fig. 8**).

C. RRAM-based TRNG

The forming scheme greatly affects the shape and quantity of conductive filaments (CFs) of RRAM, and further affects the resistive switching characteristics. Here we propose a noise-enhanced forming method to modulate the noise amplitude, as shown in **Fig. 9**. Low forming current and high target resistance makes CFs of noise-enhanced forming device more susceptible. The measured current waveforms show that the read noise after noise-enhanced forming is larger (**Fig. 10**). In addition, during 100 read cycles, standard variation of all noise-enhanced forming devices is larger than 10, and the overall distribution trend is better (**Fig. 11**). In sum, devices with noise-enhanced forming method show larger read noise and is a better random resource for TRNG.

IV. Privacy-preserving cloud Computing demostration

A. Overall structure

We construct a HE system to accomplish privacy-preserving cloud inference and the flowchart is shown in **Fig.12**. The key point in the whole HE process is that the cloud cannot access to user's private data, and the user cannot access to cloud's private algorithm model and weights. HE can only handle polynomials, so all the private data is quantified as integers and encoded as coefficient of polynomials (**Fig.13**). This encode method makes full use of each coefficient of the polynomial and greatly improves the efficiency of data processing. Moreover, the mutual irrelevance of the coefficients improves the tolerance of errors after RRAM-based HE system.

The proposed schematics of HE system is consisted of multiple RRAM arrays and digital circuits (**Fig.14**). Secret keys and public keys are generated by the RRAM-based physical unclonable function (PUF), which uses device-to-device variation of RRAM resistance [7]. TRNG unit is based on cycle-to-cycle read noise of RRAM and the INTT module also plays the role of MVM compute unit.

B. Hardware implementation of SVM network

The encryption-decryption process for heart disease prediction using support vector machine (SVM) network is implemented on hardware (**Fig. 15**). The mapping weight and mapping error in the 144K RRAM array is illustrated (**Fig. 16**). Compared with 84.49% inference accuracy on original data, accuracy of RRAM HE slightly decreases to 83.72% (**Fig. 17**).

C. Simulation of CNN network

Furthermore, the encryption-decryption process for the demonstration of Fashion MNIST using 4 layers CNN network is simulated and hardware framework and peripheral circuit modules for RRAM MVM unit are shown (**Fig. 18**). By controlling the size of the '***error***' in $\boldsymbol{a} \times \boldsymbol{s} + \boldsymbol{e}$, the decryption result can carry a small noise, so it is impossible to back-calculate the weights in the cloud (**Fig. 19**). Meanwhile, as the neural network is insensitive to small errors, there is no significant loss of inference accuracy (**Fig. 20**).

D. Benchmark

To complete a MVM calculation, RRAM array achieves an energy efficiency of 4.96 TOPS/W, showing >49.6× advantage over GPU and ~553× advantage over CPU (**Fig. 21**). Adding all other calculations and TRNG (3.51pJ/bit) [4], the energy consumption of RRAM array achieves 2μJ, showing ~8× advantage over GPU and ~39× advantage over CPU (**Fig. 22**). Compared with Microsoft SEAL library [8], RRAM-based HE reduces the time consumption by ~8.5× (with RRAM TRNG throughput = 230 Mbit/s) (**Fig. 23**) [4].

V. Conclusion

Key innovations of this work: 1) Reliable MVM computing and good randomness source for TRNG are realized on the same RRAM device by using a novel noise modulation scheme; 2) Based on the RRAM device, we propose a method for implementing homomorphic encryption and decryption; 3) The RRAM array based-encryption decryption process for cloud privacy-preserving computing is demonstrated on a multiple 144Kb RRAM arrays hardware platform, and achieves an accuracy loss of only 0.73% but with time and energy consumption reduced significantly.

Acknowledgment

This work was supported in part by the MOST of China (2019YFB2205104), the NSFC (62025111, 61874169), ICFC and the IoT Intelligent Microsystem Center of Tsinghua University - China Mobile Joint Research Institute.

References

[1] Fan J *et al.*, *IACR,* 2012. [2] Banerjee U *et al.*, *ISSCC,* 2019 [3] P. Yao *et al.*, *Nature,* 2020 [4] B. Lin *et al.*, *IEDM,* 2020. [5] W. Wu *et al.*, *VLSI,* 2018. [6] C. -H. Chang *et al.*, *MCAS,* 2015 [7] B. Lin *et al.*, *JSSC,* 2021 [8] A. C. Mert *et al.*, *TVLSI,* 2020.

The homomorphic encryption and implementation on RRAM array

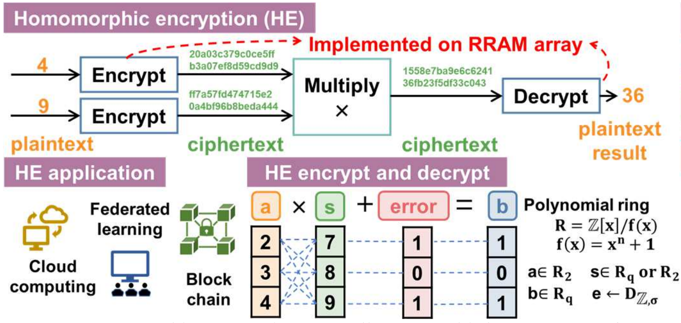


Fig. 1 Homomorphic encryption can allow any third party to perform computation on the encrypted data without decrypting it. This special property allow it has wide range of applications in many fields. HE encryption and decryption can be summarized as calculating $\boldsymbol{a} \times \boldsymbol{s} + \boldsymbol{e}$, '$\boldsymbol{a}$' and '$\boldsymbol{s}$' are on polynomial rings and '$\boldsymbol{e}$' is an error polynomial whose coefficients are Gaussian distributed.

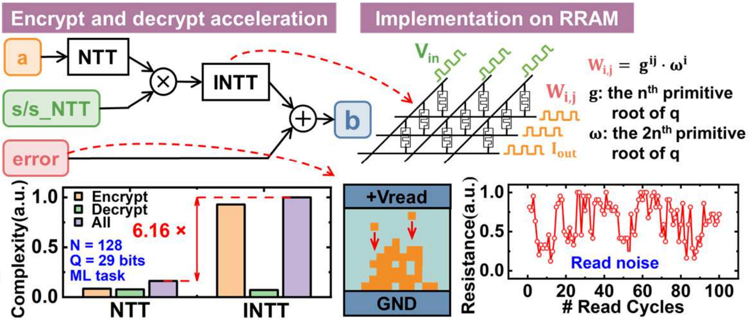
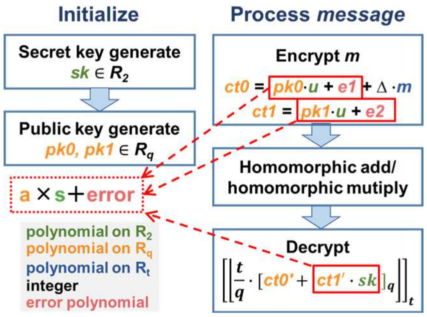
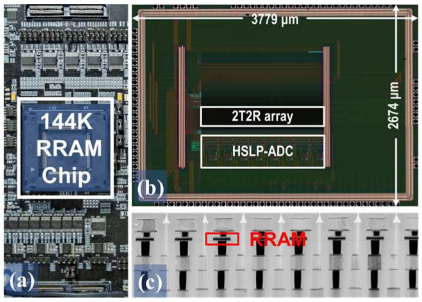
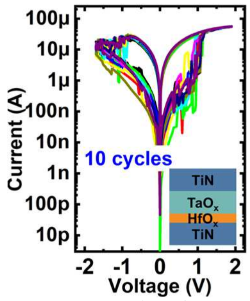


Fig. 2 Polynomial modulo multiplication with error is very suitable for RRAM array to implement. On one hand, the modulo multiplication operation can be accelerated by INTT, which is essentially a matrix-vector multiplication (MVM) and can be accelerate by RRAM array. On the other hand, the small 'error' can be easily generated by read noise within RRAM devices.



Characteristics of RRAM devices and array





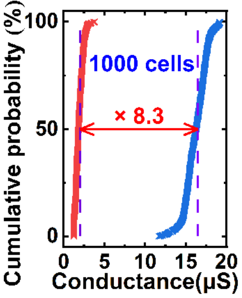
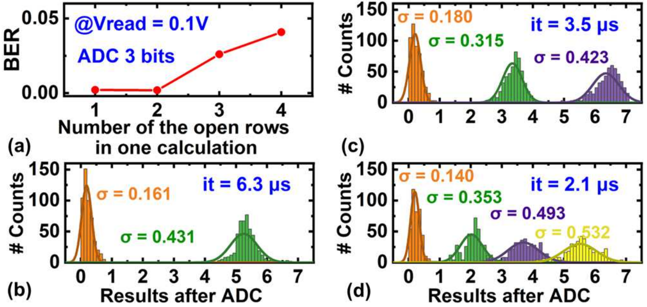


Fig. 3 Flow chart of homomorphic encryption. To encrypt or decrypt a massage, we need to calculate modular multiplication of two polynomials on different rings and add a small error.

Fig. 4 (a) Photography of the hardware test platform. (b) Die micrograph of the 144kb RRAM chip in 130-nm process. (c) Transmission electron microscope (TEM) image of RRAM.

Fig. 5 10 cycles of DC I-V sweep of one 1T1R RRAM device.

Fig. 6 The conductance distribution of 1000 1T1R cells, showing an HRS/LRS ratio near 8.3.



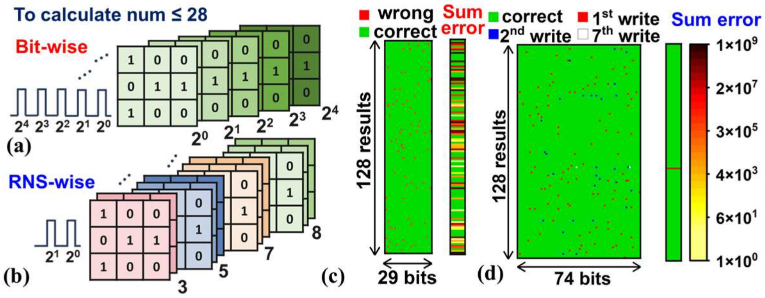
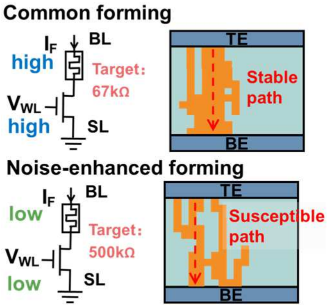
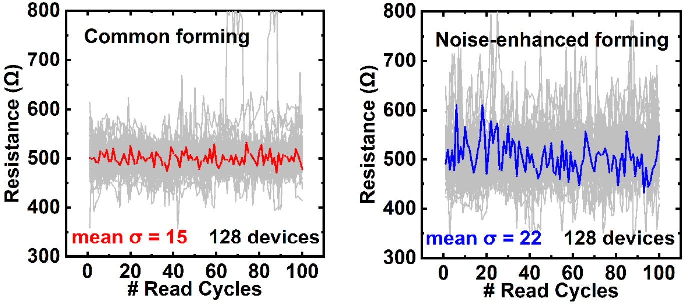


Fig. 7 The influence of the number of open rows on BER with 3-bits ADC. From figure (b)(c)(d) we can see, when opening more than or equal to 3 rows simultaneously, the resistances distribution between different levels begin to overlap and the BER increases dramatically.

Fig. 8 Different mapping methods to calculate big integer. For bit-wise expansion method (a), a small error in the high-bit multiplied by a weighting factor will invalidate all results in the low-bit (c). The RNS-wise expansion method (b) can self-detecting error, and after 7th write only 1 result out of 128 is wrong (d).





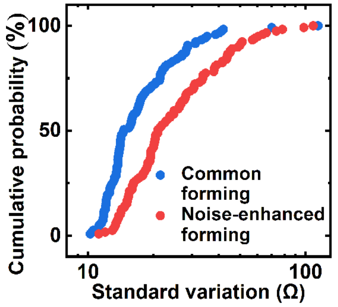


Fig. 9 The scheme and device characteristic of noise-enhanced forming method.

Fig. 10 Measured current waveforms of read noise under common forming and noise-enhanced forming. The read noise of noise-enhanced forming method is larger.

Fig. 11 STD of all the noise-enhanced forming devices is larger than 10, and the overall distribution trend is higher.

Demonstrations of privacy-preserving machine learning inference

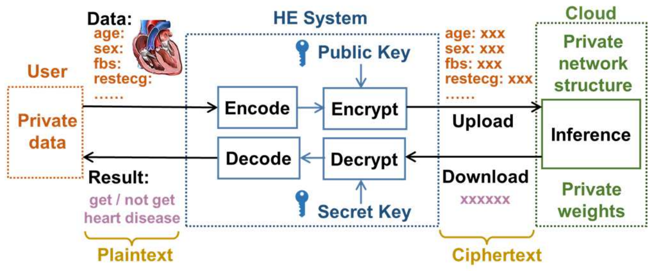
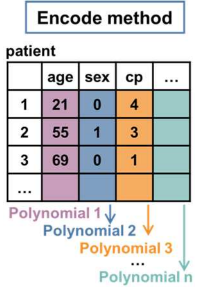


Fig. 12 Illustration of privacy-preserving cloud inference demonstration. The key point in the whole HE process is that the cloud cannot get user's private data, and the user cannot get cloud's private network structure and weights.



	age	sex	cp	…
1	21	0	4	
2	55	1	3	
3	69	0	1	
…				

Fig. 13 All the private data are encoded as coefficients of polynomials.

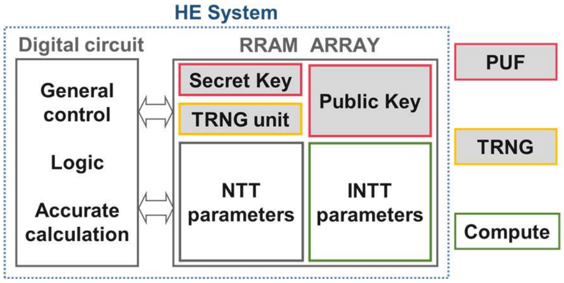


Fig. 14 The proposed schematics of HE system, which consist of RRAM array and digital circuit. Secret key and public key are generated by physical unclonable function (PUF). The RRAM array stored INTT can also do MVM calculation.

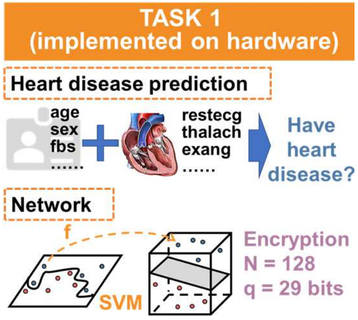


Fig. 15 Demonstration of heart disease prediction. The encryption-decryption process is implemented on RRAM hardware.

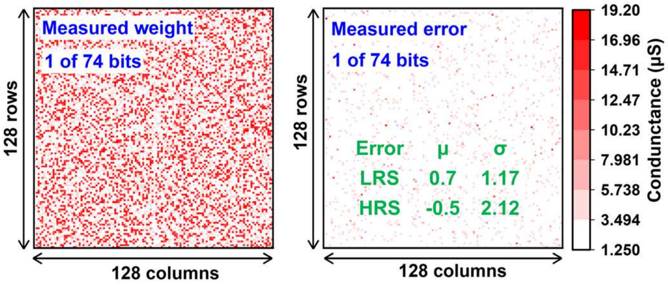


Fig. 16 The mapping weights and mapping errors of 1 layer measured in the 144K RRAM array. Two conductance levels are used and the target are 1.28μS and 16.6μS.

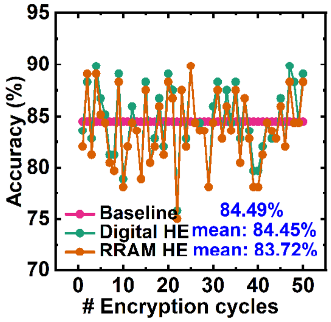


Fig. 17 The comparison of inference accuracy. Digital HE quantizes and encodes the input, moreover, RRAM HE carries its device noise.

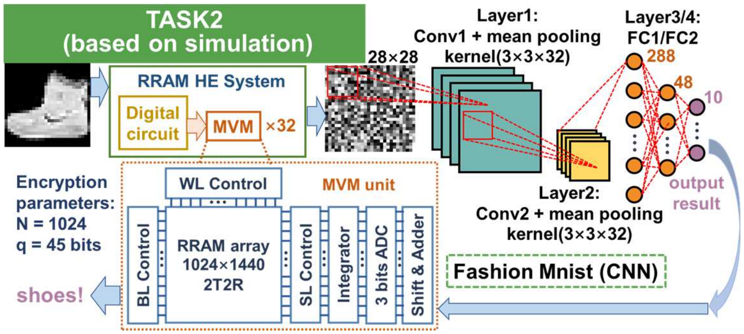


Fig. 18 Demonstration of Fashion MNIST using 4 layers CNN network. The encryption parameters are too large to be supported on hardware, so the result is based on computer simulation. Hardware framework and peripheral circuit modules for RRAM MVM unit is also shown.

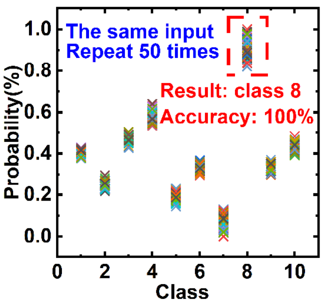


Fig. 19 By controlling the size of the 'error', the decryption result can carry a small noise, so it is impossible to back-calculating the weights.

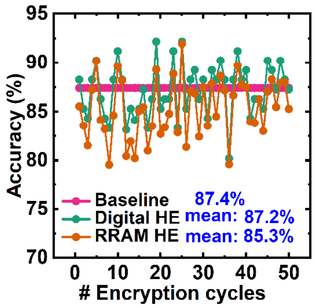


Fig. 20 The comparison of CNN inference accuracy with baseline, digital HE and RRAM HE respectively.

Experimental results and benchmark

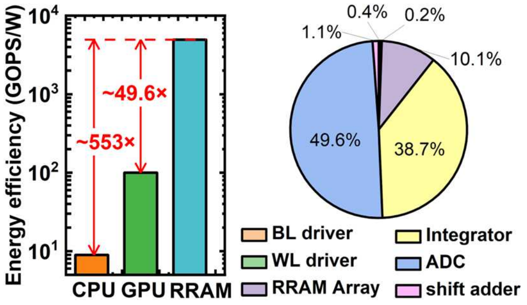


Fig. 21 Comparisons of energy efficiency between GPU, CPU and RRAM to complete a MVM calculation, and percentage of energy consumption of different circuit modules.

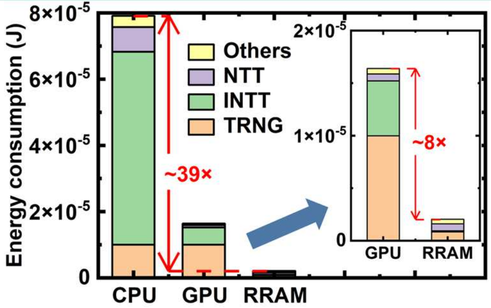


Fig. 22 Comparisons of energy consumption between GPU, CPU and RRAM to complete an encryption-decryption process for cloud inference.

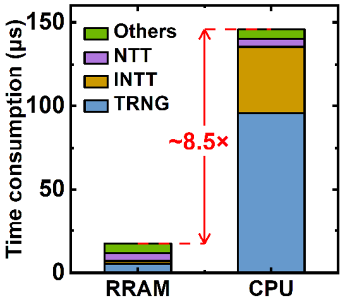


Fig. 23 Time consumption to complete an encryption-decryption process for cloud inference. And the values of CPU are based on the SEAL software library.