International Journal of Engineering Innovations in Advanced Technology
ISSN: 2582-1431 (Online), Volume-4 Issue-4, December 2022

Implementation Of Cryptography Using Rom Sub Modules And Exclusion Of Shift Rows

K Narsimha
MTech Student, Department Of ECE, VLSI System Design, Avanthi Institute of Engg. & Tech., India.
Email: prasannavemuganti@gmail.com

Dr.S. Kishore Reddy
Associate professor, HOD, Department Of ECE, VLSI System Design, Avanthi Institute of Engg. & Tech., India.
Email: kishorereddy416@gmail.com

G Srinivas
Assistant professor, Department Of ECE, VLSI System Design, Avanthi Institute of Engg. & Tech., India.
Email: sri414china@gmail.com

Abstract-In this article, we present a simple, yet energy- and area-efficient method for tolerating the stuck-at faults caused by an endurance issue in secure-resistive main memories. In the proposed method, by employing the random characteristics of the encrypted data encoded by the Advanced Encryption Standard (AES) as well as a rotational shift operation, a large number of memory locations with stuck-at faults could be employed for correctly storing the data. The technique may be employed along with other error correction methods, including the error correction code (ECC) and the error correction pointer (ECP). To assess the efficacy of the proposed method, it is implemented in a phase-change memory (PCM)- based main memory system and compared with three error tolerating methods. The results reveal that for a stuck at fault occurrence rate of 10−2 and with the method is similar to that of the state-of-the-art method.

***Keywords*- Advanced Encryption Standard, error correction code, phase-change memory**

Manuscript received Oct 10, 2022; Revised Oct 25, 2022; Accepted Nov 4, 2022

I. INTRODUCTION

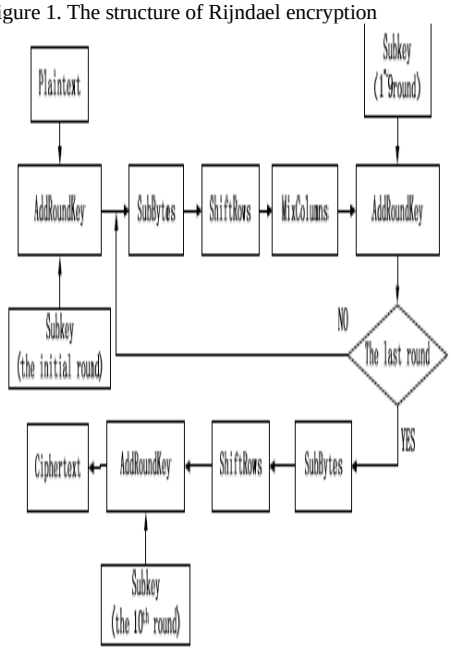
AES is short for Advanced Encryption Standard and is a United States encryption standard defined in Federal Information Processing Standard (FIPS) 192, published in November 2001. It was ratified as a federal standard in May 2002. AES is the most recent of the four current algorithms approved for federal us in the United States. One should not comparison with RSA, another standard algorithm, as RSA is a different category of algorithm. Bulk encryption of information itself is seldom performed with RSA.RSA is used to transfer other encryption keys for use by AES for example, and for digital signatures. AES is a symmetric encryption algorithm processing data in block of 128 bits. A bit can take the values zero and one, in effect a binary digit with two possible values as opposed to decimal digits, which can take one of 10 values. Under the influence of a key, a 128-bit block is encrypted by transforming it in a unique way into a new block of the same size.. Each additional bit in the key effectively doubles the strength of the algorithm, when defined as the time necessary for an attacker to stage a brute force attack, i.e., an exhaustive search of all possible key combinations in order to find the right one.

II. EXISTING METHOD

A. *Brief Description of Rijndael Algorithm*

Rijndael algorithm consists of encryption, decryption and key schedule algorithm. The main operations of the encryption algorithm among the three parts of Rijndael algorithm include: bytes substitution (Sub Bytes), the row shift (Shift Rows), column mixing (Mix Columns), and the round key adding (AddRoundKey). It is shown as Fig. 1.

figure 1. The structure of Rijndael encryption



algorithm

Encryption algorithm processes Nr+1 rounds of transformation of the plaintext for the ciphertext. The value of Nr in AES algorithm whose packet length is 128 bits should be 10, 12, or 14 respectively, corresponding to the key length of 128,192,256 bits. In this paper, only the (AES-128) encryption scheme with 128-bit keys is considered.

B. *The Design of Improved AES-128 Encryption Algorithm:*

- Two main processes of AES encryption algorithm:

The AES encryption algorithm can be divided into two parts, the key schedule and round transformation. Key schedule consists of two modules: key expansion and round key selection. Key expansion means mapping Nk bits initial key to the so-called expanded key, while the round key selection selects Nb bits of round key from the expanded key module.

Round Transformation involves four modules by Byte Substitution, Byte Rotation, MixColumn and AddRoundKey.

C. *The Process of New algorithms*

From the above analysis, we can find that the process of AES encryption can be mainly divided into two parts: key schedule and round transformation. The improved structure is also divided into these two major processes. The initial key will be sent to the two modules: Key expansion and Key selection, while the plaintext is to be sent to the round transformation after the roundkey is selected. But the operand of data transmission is turned into a 32-bit unit. The process of new algorithm is shown as Fig.

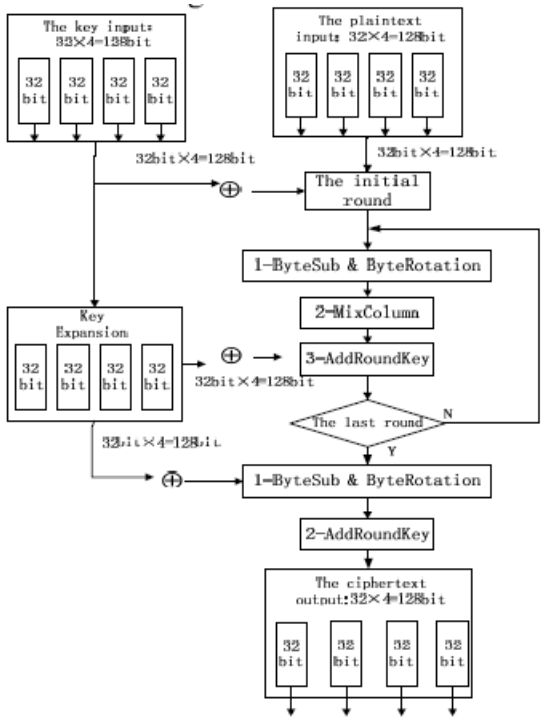


Figure 2. The new improved structure of AES algorithm

The functions of various parts of the structure shown above are described as follow:

- The initial round of encryption:

The four packets of consecutive 32-bit plaintext (128 bits) have been put into the corresponding registers. Meanwhile, another four packets of consecutive 32-bit initial key (128 bits) have been put into other registers by the control of the enable clock signal. Furthermore, this module should combine the plaintext and initial key by using the XOR operators.

D. Round Transformation in the intermediate steps:

A round transformation mainly realizes the function of SubBytes and MixColumns with 32-bit columns. Four packets of round transformation are processed independently. Then the results of MixColumns and the 32-bit keys sourced from Keyexpansion are combined by using XOR operators. Here, the round transformation is a module with 64 input ports (32-bit plaintext+32-bit key) and 32 output ports.

The function of SubByte is realized by Look-Up Table (LUT). It means that the operation is completed by the Find and Replace after all replacement units are stored in a memory (256×8bit = 1024 bit).

The implementation of MixColumn is mainly based on the mathematical analysis in the Galois field GF(28). Only the multiplication module and the 32-bit XOR module of each processing unit(one column) are needed to design, because the elements of the multiplication and addition in Galois field are commutative and associative. Then the function of MixColumn can be achieved. Fig.5 is a block diagram for the introduction of pipelining technology used in the round transformation.

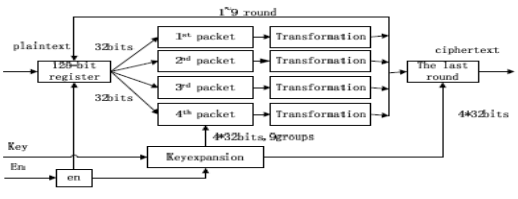


Figure 3. The round processing with pipeline technology

In the process of pipelining, the 128-bit data is divided into four consecutive 32-bit packets that take ound transformation independently.

The operation of the above four groups of data can be realized in pipelining technology. In brief, it can be described as follow: store the unprocessed data in the 128-bit register, and

control the clock for re-starting the 128-bit register to read the new data when the four groups' operations have been overcome. Thus the 128-bit round-operating unit has been transformed into four 32-bit round-operating elements. The internal pipelining processing should be implemented during the whole nine intermediate Round Transformations of the four packets before achieving the 128-bit ciphertext.

- The process of the last round

The final round is a 128-bit processor. After nine rounds of operations included Shiftrows, SubByte and Mixclumns, the 128-bit intermediate encrypted data will be used in XOR

operation with the final expanded key(4*32bit), which is provided by the key expansion module. The output of final round in the processor is the desired 128-bit ciphertext. Similarly, the ciphertext is divided into four packets of 32- bit data by an external enable signal.

- Key expansion and Key extraction

This module is implemented basically the same with the traditional way as another part of the AES encryption algorithm. The only difference lies on the mode of data transmission. The

initial key and expanded keys are divided into four 32-bit data before being extracted.

All of the above modules can be decomposed into basic operations of seeking and XOR if the AES algorithm is implemented on FPGA. So the basic processing unit (look-uptable)of FPGA can be used. The operation of AddRoundKey is taken first in each round. When the plaintext and initial key are input, the encryption module starts running, and the expanded keys are stored into the registers at the

same time. This implementation method is independent on a specific FPGA.

III. VLSI

Very-large-scale integration (VLSI) is the process of creating integrated circuits by combining thousands of transistor-based circuits into a single chip. VLSI began in the 1970s when complex semiconductor and communication technologies were being developed. The microprocessor is a VLSI device. The term is no longer as common as it once was, as chips have increased in complexity into the hundreds of millions of transistors.

A. VLSI and systems

These advantages of integrated circuits translate into advantages at the system level:

Smaller physical size. Smallness is often an advantage in itself-consider portable televisions or handheld cellular telephones.

Lower power consumption. Replacing a handful of standard parts with a single chip reduces total power consumption. Reducing power consumption has a ripple effect on the rest of the system: a smaller, cheaper power supply can be used; since less power consumption means less heat, a fan may no longer be necessary; a simpler cabinet with less shielding for electromagnetic shielding may be feasible, too.

Reduced cost. Reducing the number of components, the power supply requirements, cabinet costs, and so on, will inevitably reduce system cost. The ripple effect of integration is such that the cost of a system built from custom ICs can be less, even though the individual ICs cost more than the standard parts they replace.

A. ASIC

An Application-Specific Integrated Circuit (ASIC) is an integrated circuit (IC) customized for a particular use, rather than intended for general-purpose use. For example, a chip designed solely to run a cell phone is an ASIC. Intermediate between ASICs and industry standard integrated circuits, like the 7400 or the 4000 series, are application specific standard products (ASSPs).

As feature sizes have shrunk and design tools improved over the years, the maximum complexity (and hence functionality) possible in an ASIC has grown from 5,000 gates to over 100 million. Modern ASICs often include entire 32-bit processors, memory blocks including ROM, RAM, EEPROM, Flash and other large building blocks. Such an ASIC is often termed a SoC (system-on-a-chip). Designers of digital ASICs use a hardware description language (HDL), such as Verilog or VHDL, to describe the functionality of ASICs.

Field-programmable gate arrays (FPGA) are the modern-day technology for building a breadboard or prototype from standard parts; programmable logic blocks and programmable interconnects allow the same FPGA to be used in many different applications. For smaller designs and/or lower production volumes, FPGAs may be more cost effective than an ASIC design even in production.

- An application-specific integrated circuit (ASIC) is an integrated circuit (IC) customized for a particular use, rather than intended for general-purpose use.
- A Structured ASIC falls between an FPGA and a Standard Cell-based ASIC

Structured ASIC's are used mainly for mid-volume level design. The design task for structured ASIC's is to map the circuit into a

IV. PROPOSED METHOD

A. Introduction

Computational processing has increased in cloud servers, requiring larger core counts and higher memory densities. The number of processor cores doubles every two years, while the DRAM DIMM capacities double every three years [1]. This causes a large gap between the core count and the memory density. On the other hand, traditional DRAM chips consume more than 40% of power of the servers [2]. Also, DRAM scaling to reach a higher memory density has some challenges such as high leakage current, reduced memory cell reliability, and more

complex fabrication processes [3]. Emerging memory technologies, which are categorized into volatile (DRAM-based) and nonvolatile (resistive-based (have been introduced to solve scaling and power consumption problems [4]. Some of the volatile DRAM-based memories include reduced latency and tiered latency DRAM (RL-DRAM and TL-DRAM) and low power DDR DRAM (e.g., LPDDR3, LPDDR4) architecture. While they have lower latency and power consumption, they suffer from higher costs of the fabrication process [5]. algorithms increase by enlarging the number of errors imposing considerable overheads on the system .

B. Randshift Method

As mentioned before, owing to the limited write endurance of the PCM, some of the cells are worn-out, permanently become stuck-at "1" or "0" value. The idea of fault coverage based on "ST-R" and "ST-W" is demonstrated by a memory word shown in Fig. 1, where 4-bit positions have stuck-at faults (i.e., the bit positions of 0, 4, 10, and 13). For example, in this memory word, storing "0xB3A8" leads to ST-W at the 4th- and 13th-bit positions. In this case, a 1-bit circular shift to the right causes the value to become "0x59D4" giving rise to ST-R at the 4th- and 13th-bits without inducing any other ST-W. As stated previously, the correlation of any two AES encrypted data is almost zero, and thus one may consider the output of the AES encryption as a random number.

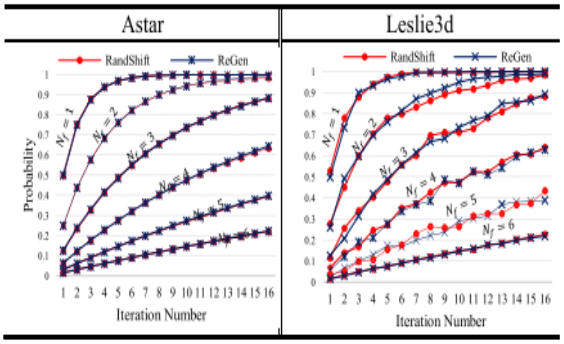


Figure 4 Probability of fault coverage.

In Fig, the average correlation coefficient of AES-128 encryption for 50K 128-bit encrypted memory data in the "astar" benchmark from CPU2006 is shown [22]. As Fig. indicates, the illustrated average is very close to zero implying a low relationship between the output data in each iteration of the AES encryption method. The randomness feature of the output value in the AES may be employed as a solution to tolerate stuck-at faults in PCM cells. However, in the case of raw data (data that are not encrypted), due to the existence of spatial/temporal correlations among the data blocks, the use of manipulating methods (e.g., circular shifting or inversing) may not be appropriate for overcoming the problem of the stuck-at faults. If the encrypted data fail to fully match with the stuck-at faults (the number of "ST-W" is zero), one may use the technique suggested in [12] to regenerate the encrypted data.

Otherwise, a data write failure exception signal is issued to inform, e.g., the processor of the write failure (lines 14–19).

As mentioned before, due to the possibility of having to do a large number of regenerations, this technique is a power-hungry approach. Each mismatch between the writing data and the stuck-at fault values causes to add ten rounds of add-Round-Key, substitute, shift-Rows, and mix-Column in AES-128. Although data shifting approach, as well as the generation of new encrypted data may not completely keep the randomness of the data, it may increase the number of "ST-R" in the case when there are not many stuck-at fault bits in the memory block. Obviously, the former approach is considerably faster and more energy efficient. Since the RandShift approach is applied on the encrypted data whose bits enjoy a high degree of randomness, the disturbance in the randomness in the RandShift approach is similar to that in the regeneration approach. To evaluate the efficacy of the proposed RandShift method compared to the one proposed in [12] (which is denoted ReGen in the rest of this article), probabilities of the stuck-at fault coverages of the two methods should be compared. Because of the randomness property of AES encryption, there is no dependency between the previous and the next data generations at each time in the ReGen method [12]. Because of the data dependency between the shifted bits in the RandShift method, extracting a close-form formula to calculate the fault coverage probability is not possible.

The coverage probabilities of RandShift and ReGen were obtained by applying one million encrypted data as stimuli. In this article, the number of faults in 128 bits of data does not exceed six for the assumed distribution. In Fig, the parameter Diff indicates the average absolute difference between the two cases of RandShift and ReGen. When a data write request arrives from the LLC, the memory controller encrypts the data block (lines 1 and 2). Then, the RowVerifier checks the stuck-at faults value and their positions by storing and reading the pattern of all 1s and 0s in the memory [6] (line 3). Next, the RandShift method iterates until the shifting process hides all stuck-at faults (lines 4–13) or the number of iterations reaches a predefined value. Finally, if all faults are matched, the data write process will be performed.

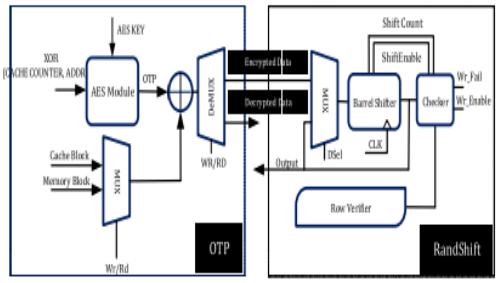


Figure 5. Full architecture of RandShift.

V. CONCLUSION

In this work, we proposed a method employing the randomness feature of AES encryption as well as rotational shift operation to tolerate hard faults in nonvolatile memory cells. This method, which was called RandShift, enjoyed the simple hardware implementation and low energy consumption. It limited the need for exploiting powerful error correction methods, such as ECC and ECP. The results of our comparative study showed up to 82% lower energy consumption for RandShift when obtaining about the same fault coverage as that of the state-of-the-art technique.

VI. FUTURE SCOPE

A FPGA implementation of area-optimized AES algorithm which meets the actual application is proposed in this paper. After being coded with Verilog Hardware Description Language, the waveform simulation of the new algorithm was taken in platform. Ultimately, a synthesis simulation of the new algorithm has been done. The result shows that the design with the pipelining technology and special data transmission mode can optimize the chip area effectively.

REFERENCES

[1]. Madakam, Somayya, R. Ramaswamy, and Siddharth Tripathi. “Internet of Things (IoT): A literature review.” Journal of Computer and Communications 3, no. 05 (2015): p.164.
[2]. Wang, Yong, Garhan Attebury, and Byrav Ramamurthy. “A survey of security issues in wireless sensor networks.” IEEE Communications Surveys Tutorial (2006).
[3]. Veeramallu, B., S. Sahitya, and Ch LavanyaSusanna. Veeramallu, B., S. Sahitya, and Ch LavanyaSusanna. “Confidentiality in Wireless sensor Networks.” International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-6, January 2013.
[4]. Eisenbarth, Thomas, and Sandeep Kumar. “A survey of lightweightcryptography implementations.” IEEE Design & Test of Computers 24.6 (2007).
[5]. Banik, Subhadeep, Andrey Bogdanov, and Francesco Regazzoni. “Exploring energy efficiency of lightweight block ciphers.” International Conference on Selected Areas in Cryptography. Springer, Cham, 2015.
[6]. Bogdanov, Andrey, et al. “PRESENT: An ultra-lightweight block cipher.” CHES. Vol. 4727. 2007.
[7]. Borghoff, Julia, et al. “PRINCEa low-latency block cipher for pervasive computing applications.” International Conference on the Theory and Application of Cryptology and Information Security. Springer, Berlin, Heidelberg, 2012.
[8]. Beaulieu, Ray, et al. “The SIMON and SPECK lightweight block ciphers.” Design Automation Conference (DAC), 52nd ACM/EDAC/IEEE. IEEE, 2015.
[9]. Suzaki, Tomoyasu, et al. “TWINE: A Lightweight Block Cipher for Multiple Platforms.” Selected Areas in Cryptography. Vol. 7707. 2012.