Design of 16-bit Countermeasure Circuit with AES Algorithm for AES Engine

M.Petchiammal ,
PG Scholar,(VLSI),Kalasalingam Academy of Research And Education.,
Dept.of Electronics &Communication Engg,

G.Ramyadevi,
Assistant Professor-I
Kalasalingam Academy of Research And Education,
Dept.of Electronics &Communication Engg,

ABSTRACT

The DPA attack can efficiently disclose the secret key of an AES Engine easily. To increase DPA Resistant of the AES engine by XORing the generated 16 bit from Pseudo Random Number Generator with the cipher text from the AES Engine. The cipher text is created by AES algorithm which is very Efficient Algorithm for data Securing. The 16 Bit Sequence Generator Circuit also provide Reduction in Area occupied by the Countermeasure circuit and Delay of propagation time by using pipelining process. The Speed of the DPA Countermeasure circuit also increased without degradation in throughput.

General Terms

Differential Power Analysis (DPA), ring oscillators, True random number generator (TRNG), Pseudo Random Number Generator (PRNG).

Keywords- AES (Advanced Encryption Standard)

1. INTRODUCTION

The differential power analysis (DPA) attack proposed by Kocher et al. in 1999 has become a serious issue when designing cryptographic circuits. The DPA attack can efficiently disclose the secret key by the power consumption information leaked from cryptographic devices. It has been proven that the secret key of an Advanced Encryption Standard (AES) chip can be disclosed within 10,000 measurements. Accordingly, the DPA resistance has become the most important consideration for hardware-based cryptographic devices. Several methods have been proposed to counteract the DPA attack, either in the algorithm or in the circuit level. Some of them use a data masking method to randomize the data processed in cryptographic circuits .The data being processed is changed by an internally generated random mask before the en-/decryption process. As a result, a corresponding mask should be used to recover the actual output data at the end of the process. In this way, the power consumption of cryptographic circuits will be independent of the predicted power consumption. Some proposals balance the power consumption of different operations by using new logic cells called sense amplify based logic or wave dynamical differential logic (WDDL).Standard cells are replaced by this new logic family and then the power consumption of different patterns would be almost the same. Some proposals isolate the power supply and cryptographic circuits by switching capacitors .The current is charged to a capacitor array, and the current consumed by cryptographic circuits is then supplied by the capacitor array instead of the power supply However, the increased security level results in extra hardware cost and throughput degradation. For example, the WDDL method can increase the security with 3 times larger silicon area and 75% throughput degradation The switching capacitor method can reduce the area overhead to 27%, but the performance is still degraded by 50% .A ring-oscillator-based DPA countermeasure circuit can effectively reduce the area overhead and throughput degradation .Details of the ring-oscillator-based same after the system is reset. Therefore, the additional power consumption added by the DPA countermeasure circuit in each cycle would be the same if the attacker resets the system before recording power traces. To solve problem in a different architecture that incorporates a true random number generator is proposed not only to counteract the DPA attack but also to self-generate a true random sequence. With the proposed architecture, the security level of AES engines can be further enhanced while the area overhead can be also reduced problem in a different architecture that incorporates a true random number generator is proposed not only to details of the ring-oscillator based DPA countermeasure circuit such as inversion stages.

2. DPA ATTACK

The DPA attack utilizes the statistical analysis to calculate the correlation between the leaked power information and the predicted power consumption. Irrelative noises can be eliminated by statistical analysis and therefore, the DPA attack can still be successfully conducted even in a noisy environment. The secret key of a cryptographic circuit can be disclosed from the correlation index of the analysis result. For the AES algorithm, the 128-bit secret key can be divided into 16 8-bit subkeys, and the attacker can disclose each 8-bit subkey at one time. As a result, the array would consist of 2^8 = 256 columns for all key hypotheses. After the measured and the predicted power arrays are available, the secret key can be disclosed by the statistical analysis.Each column of the predicted power array is used to find a correlation index with every column of the measured power array. If the key hypothesis matches the secret key used by the cryptographic circuit, the correlation index would be higher than that of other key hypotheses.

3. DPA COUNTERMEASURE CIRCUIT

The true random-based architecture is introduced first and then the improved architecture with self-generated random sequence is presented.

3.1 True Random-Based DPA Countermeasure Circuit

To solve the security weakness in the pseudo random-based architecture, a true random sequence for the DPA

countermeasure circuit is required. However, most true random number generators are analog circuits with much higher power consumption. Goli proposed a digital method to generate random data by using ring oscillators in Fibonacci and Galois configurations .The Fibonacci and the Galois ring oscillator consists of a series of inverters connected with feedback polynomial. The proposed architecture incorporates a true random number generator into the DPA countermeasure circuit to resist the DPA attack and the reset problem mentioned earlier. The combination combination of two FiROs and two GaROs is used as the random source to generate one **random sequence. In order to generate eight independent random bits for each databyte, a** total of 32 ring oscillators (including Fibonacci and Galois ring oscillators) are required in the DPA countermeasure circuit. These sixteen random sources are sampled by flip-flops for further post processing input is obtained by XORing one data byte with arandom mask, and 16 ring oscillators are directly controlled by this16-bit input. The random mask can be After postprocessing, these 16 random bits are XORed with data bytes from the cryptographic circuit to dynamically enable oscillators in the FiRO and GaRO. The FiRO and GaRO now work not only as random sources into generate random data but also as the digitally controlled ring oscillators in to counteract the DPA attack. The FiRO will not have a fixed point if and only if $f(x) = (1 + x)h(x)$ and $h(1) = 1$, where $f(x)$ is the polynomial presentation of the feedback configuration for FiRO, and $h(x)$ is a primitive polynomial. Note that a fixed point is a state that the output vector of inverters is an alternating string of 1 and 0 ({ 01010 · · ·} or { 10101 · · ·}). Since each random source is from the combination of four different ring oscillators, at least four different $h(x)$ are required. To have four different forms of $h(x)$, the minimum degree of $f(x)$ for the FiRO is 6. Similarly, the condition for the GaRO, having no fixed point, is $f(1) = 0$, and the degree of $f(x)$ must be odd Again, in order to have four different configurations, the minimum degree of $f(x)$ for GaROs must be 7.The post processing circuits are composed of LFSRs with different initial seeds. The purpose of the postprocessing circuit. is to remove the bias of the random source. In each postprocessing circuit, the feedback value is XORed with that from the random source. In this way, even the postprocessing circuit starts from a deterministic state after the system is reset, the generated random

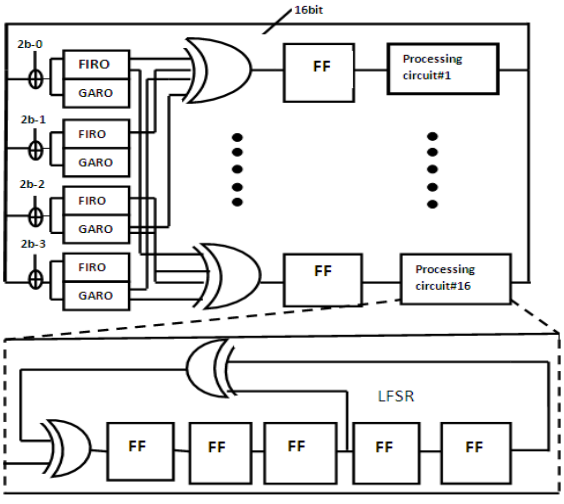

Fig. 1 Architecture of the DPA countermeasure circuit with self-generatedtrue random sequence.

random sequence would not be the same because the random source is added into the feedback of the LFSR. The means show that the random sequence would be VDD/2, and the standard deviations. The countermeasure circuit consists of12 ring oscillators, each of which can be enabled or disabled in-dependently. When a ring oscillator is generated by an internally designed random number generator, whose randomness dominates the DPA resistance of our proposed countermeasure circuit.The remaining four oscillators are controlled by pairsof these 16 inputs.The postprocessing circuits are composed of LFSRs with different initial seeds. The purpose of the postprocessing circuit is to remove the bias of the random source. In each postprocessing circuit, the feedback value is XORed with that from the random source. In this way, even the postprocessing circuit starts from a deterministic state after the system is reset, the generated random sequence to remove the bias of the random source. In each postprocessing circuit, the feedback value is XORed with that from the random source. In this way, even the postprocessing circuit starts from a deterministic state after the system is reset, the generated random sequence sequences generated with the proposed architecture, although the standard deviations are zero in the first few cycles, which means the generated bits in these cycles would be always the same after the system is reset.

3.2 .AES Algorithm

A symmetric block cipher that can process data blocks of 128 bits, using cipher keys with lengths of 128, 192, and 256 bits. The input and output for the AES algorithm each consist of sequences of 128 bits (digits with values of 0 or 1). The Cipher is described in the pseudo code in The individualtransformations,SubBytes,ShiftRows,MixColumns,and AddRoundKey.

1.Encryption

1. 1 SubBytes Transformation

The SubBytes transformation is a non-linear byte substitution that operates independently on each byte of the State using a substitution table (S-box).

1.2 ShiftRows Transformation

ShiftRows transformation, the bytes in the last three rows of the State are cyclically shifted over different numbers of bytes.

1.3 MixColumns Transformation

The MixColumns transformation operates on the State column-by-column, treating each column as a four-term polynomial. The columns are considered as polynomials.

1.4AddRoundKeyTransformation

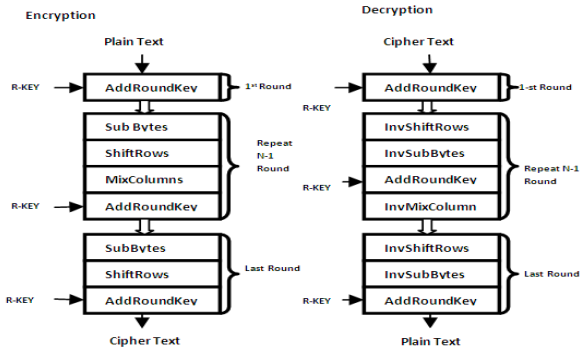

Fig.3 .AES algorithm

AddRoundKey transformation, a Round Key is added to the State by a simple bitwise XOR operation.enabled, it will consume additional power to change the power consumption characteristic. An16-bit sequence

2.Decryption

2.1 InvShiftRows Transformation

InvShiftRow) is the inverse of the ShiftRows transformation. The bytes in the last three rows of the State are cyclically shifted over different numbers of bytes (offsets). The first row,r = 0, is not shifted.

2.2 InvSubBytes Transformation

InvSubBytes is the inverse of the byte substitution transformation, in which the inverse S-box is applied to each byte of the State.

2.3 AddRoundKey Transformation

AddRoundKey transformation, a Round Key is added to the State by a simple bitwise XOR operation.

2.4 InvMixColumns Transformation

InvMixColumns is to he inverse of the MixColumns transformation. InvMixColumns operates. on the State column-by-column, treating each column as a four term polynomial.

4.The Subpipelined Architecture

Three architectural optimization approaches can be used to speed up the AES algorithm in non-feedback modes by duplicating hardware for implementing each round, which is also called round unit. These architectures are based on pipelining, subpipelining and loop-unrolling. The pipelined architecture is realized by inserting rows of registers between each round unit. Similar to the pipelining, subpipelining also inserts rows of registers among combinational logic, but registers are inserted both between and inside each round unit. In pipelining and subpipelining, multiple blocks of data are processed simultaneously. Comparatively, loop unrolled or unfolded architectures can process only one block of data at a time, but multiple rounds are processed in each clock cycle. Among these architectural optimization approaches

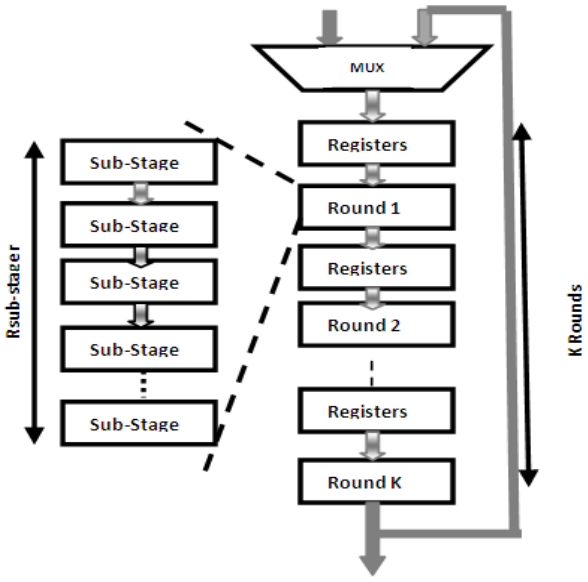

Fig.3 The architecture of subpipelining.

Comparatively, loop unrolled or unfolded architectures can process only one block of data at a time, but multiple rounds are processed in each clock cycle. Among these architectural optimization approaches, subpipelining can achieve maximum speedup and optimum speed/area ratio in non-feedback modes. However, dividing each round unit into arbitrary number of substages does not always bring speedup. Since the minimum clock period is determined by the indivisible component with the longest delay, dividing the rest of the round unit into more substages with shorter delay does not reduce the minimum clock period. Although more blocks of data are being processed simultaneously, the average number of clock cycles to process one block of data does not change. Therefore, the overall speed does not improve despite increased area caused by the additional registers. In a LUT-based implementation, it can be observed that nearly half the delay of a round unit is attributed to the LUTs, and thus, each round unit can be divided into only two substages to achieve some speedup without wasting any area. On the contrary, the longest unbreakable delay in the non-LUT-based approaches is the delay of individual logic gates. Accordingly, each round unit can be divided into multiple substages with approximately equal delay. LFSR is a linear feedback shift register used to generate various bit Except all zerovalues.LFSR is good for pseudo random number generator.If the reset is enable the input is stored as output at the register where else the variety of bit pattern can be generated.The initial value of the LFSRis called as seed value because the operation of the register is deterministic, the stream of values produced by the register is completely determined by its current (or previous) state.. A **linear feedback shift register** (LFSR) is a shift register whose input bit is a linear function of its previous state. The most commonly used linear function of single bits is XOR. Thus, an LFSR is most often a shift register whose input bit is driven by the exclusive-or (XOR) of some bits of the overall shift register value. In the Galois configuration, when the system is clocked, bits that are not taps are shifted one position to the right unchanged. The taps, on the other hand, are XOR'd with the output bit before they are stored in the next position. The new output bit is the next input bit. The effect of this is that when the output bit is zero all the bits in the register shift to the right unchanged, and the input bit becomes zero. When the output bit is one, the bits in the tap positions all flip (if they are 0, they become 1, and if they are 1, they become 0), and then the entire register is shifted to the right and the input bit becomes 1. To generate the same output stream, the order of the taps is the *counterpart* (see above) of the order for the conventional LFSR, otherwise the stream will be in reverse. Note that the internal state of the LFSR is not necessarily the same. Galois LFSRs do not concatenate every tap to produce the new input (the XOR'ing is done within the LFSR and no XOR gates are run in serial, therefore the propagation times are reduced to that of one XOR rather than a whole chain), thus it is possible

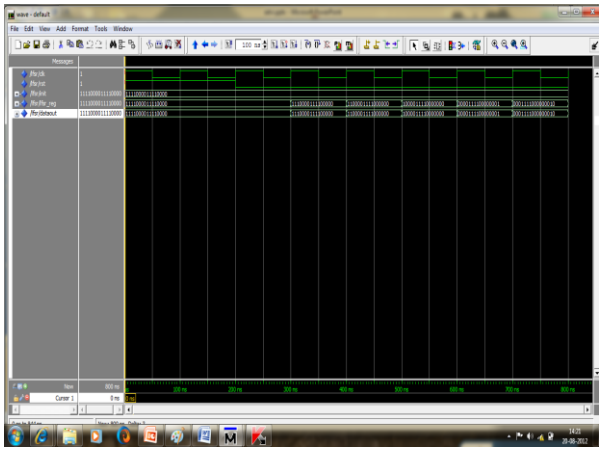

Fig.4 Simulation Result of LFSR

for each tap to be computed in parallel, increasing the speed of execution.In a software implementation of an LFSR, the Galois form is more efficient as the XOR operations can be implemented a word at a time: only the output bit must be examined individually.The taps are XOR'd sequentially with the output bit and then fed back into the leftmost bit. The sequence of bits in the rightmost position is called the output stream. The bits in the LFSR state which influence the input are called *taps.*

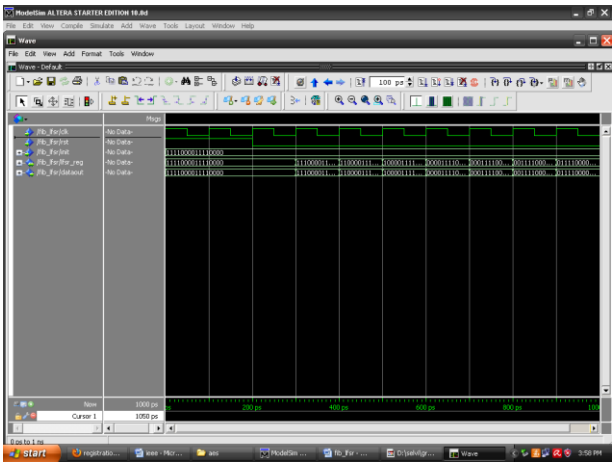

Fig.5 Simulation Result of FIRO

for each tap to be computed in parallel, increasing the speed of execution.In a software implementation of an LFSR, the Galois form is more efficient as the XOR operations can be implemented a word at a time: only the output bit must be examined individually.The taps are XOR'd sequentially with the output bit and then fed back into the leftmost bit. The sequence of bits in the rightmost position is called the output stream. The bits in the LFSR state which influence the input are called *taps.*

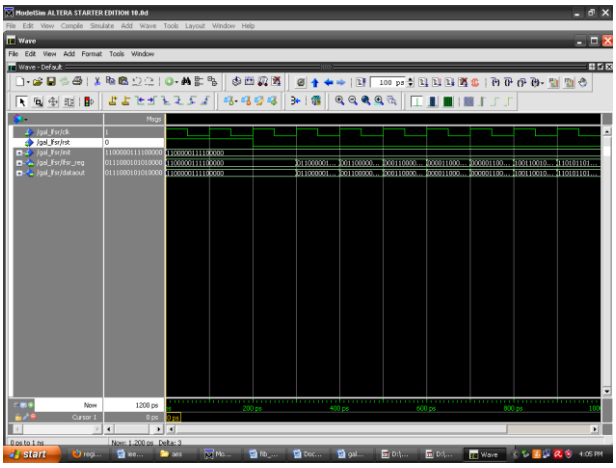

Fig.6 Simulation Result of GARO

5. ACKNOWLEDGMENTS

Our thanks to the experts Ms.G.Ramyadevi have contributed towards development of the template

Table .I

Key- Block Round Combinations

	Key length(N_k)	Block Size(N_b)	Number of Rounds(N_r)
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

6. CONCLUSION

There is currently no evidence that AES has any weaknesses making any attack other than exhaustive search, i.e. brute force, possible. Even AES-128 offers a sufficiently large number of possible keys, making an exhaustive search impractical for many decades, provided no technological breakthrough causes the computational power available to increase dramatically and that theoretical research does not find a short cut to bypass the need for exhaustive search.. The DPA attack utilizing the statistical analysis can efficiently disclose secret keys of cryptographic devices. Although the pseudo random-based method has the advantage of easy implementation, the DPA resistance is largely reduced if the system is reset before recording the power trace Accordingly, a true random-based architecture utilizing ring oscillators is proposed to resolve the reset problem by the self-generated true random sequence. The major contribution is that the security level of an AES engine can be improved by the proposed DPA countermeasure circuit. In addition, another minor improvement is that the area overhead can be reduced due to hardware sharing of ring oscillators for generating random power and random sources.

7. REFERENCES

[1] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in Proc. 19th Annu. Int. Cryptology Conf. Adv. Cryptology, 1999, pp. 388–397.

[2] D. Hwang, K. Tiri, A. Hodjat, B.-C. Lai, S. Yang, P. Schaumont, and I. Verbauwhede, "AES-based security coprocessor IC in 0.18- μm CMOS with resistance to differential power analysis side-channel attacks," IEEEJ. Solid-State Circuits, vol. 41, no. 4, pp. 781–792, Apr. 2006.

[3] C. Tokunaga and D. Blaauw, "Securing encryption systems with a switched capacitor current equalizer," IEEE J. Solid-State Circuits, vol. 45, no. 1, pp. 23–31, Jan. 2010.

[4] M.-L. Akkar and C. Giraud, "An implementation of DES and AES, secure against some attacks," in Proc. CHES, 2001, pp. 309–318.

[5] D. Suzuki, M. Saeki, and T. Ichikawa, "Random switching logic: A countermeasure against DPA based on transition probability," Cryptology ePrint Archive, Rep. 2004/346, 2004. [Online]. Available:http://eprint.iacr.org

[6] E. Oswald, S. Mangard, N. Pramstaller, and V. Rijmen, "A side-channel analysis resistant description of the AES S-Box," in Proc. 12th Int. Work-shop FSE, 2005, pp. 413–423.

[7] E. Trichina, T. Korkishkoand, and K. H. Lee, "Small size, low power, side channel-immune AES synthesis results," in Proc.AES, vol. 3373, Lecture Notes in Computer Science, 2005, pp. 113–127.

[8]. Mohammad Musa, Edward Schaefer, and Stephen Wedig, A simplified AES algorithm and its linear and differentialcryptanalyses, Cryptologia 27 (April 2003), no. 2, 148–177.

[9]. A. Lee, NIST Special Publication 800-21, Guideline for Implementing Cryptography in the Federal Government, National Institute of Standards and Technology, November 19

[10]. FIPS PUB 197, Advanced Encryption Standard (AES), National Institute of Standards and Technology, U.S. Department of Commerce, November 2001(http://csrc.nist.gov/ publications/fips/fips197/fips-197.pdf).

[11]. Stalling, W., "The Advanced Encryption Standard",Cryptologia, vol. 26, 2002, pp. 165-188.