Design and Implementation of an Encryption Algorithm for use in RFID System

M. B. Abdelhalim
AASTMT Cairo, Egypt

M. El-Mahallawy, M. Ayyad
AASTMT Cairo, Egypt

A. Elhennawy
ASU Cairo, Egypt

Abstract

The Tiny Encryption Algorithm (TEA) is a suitable lightweight cryptographic algorithm used in medium security systems such as RFID systems. The TEA is a fiestel structure used to satisfy the confusion and the diffusion properties to hide the statistical characteristics of the plaintext. However, TEA has few weaknesses, most notably from equivalent keys and related-key attacks. So, a Modified TEA algorithm (MTEA) is proposed which uses the Linear Feedback Shift Register (LFSR) to overcome the security weakness of the TEA algorithm against attacks. In this paper an implementation of MTEA algorithm is presented and benchmarked with the standard TEA algorithm considering the area and power consumption.

1. Introduction

Radio Frequency IDentification (RFID) is an automatic identification technology which consists of two main parts (Reader and tag for object) that uses radio waves to identify objects. The main benefits of RFID systems are that they can provide automated contactless identification of a range of physical entities, and can be used to track valuable objects. From the way in which RFID tags operate via a wireless radio communications channel, there is a concern about privacy and security, including the possibility of eavesdropping, snooping, cloning, counterfeiting and tracking of end users since information stored in tags can easily be retrieved by hidden readers, eventually leading to violation of user privacy and tracking of individuals by the tags they carry [1].

One of the best ways to provide security and privacy measures is through an authentication process. Authentication is an assurance of the identity of an entity at the other end of communication channel. There are various authentication schemes or protocols such as password protection which is an example of a weak authentication and strong authentication schemes such as those based on a challenge and response concept. Many RFID authentication protocols use cryptographic techniques to protect messages exchanged over a radio frequency interface from eavesdropping. Compared to asymmetric or public key alternatives, a symmetric key is generally less complicated, requires less number of operations, and can have the same security strength as its asymmetric equivalence using a key of smaller size. These facts make the symmetric key approach more suitable for limited resource RFID systems [2].

RFID tags have limited processing power and storage because of tight tag cost requirements. Symmetric cryptographic schemes, such as hash functions and symmetric encryption algorithms, are commonly used. As a result, authentication protocols for RFID systems should not only be designed to address privacy and security threats, but should also take into account the limited capabilities of RFID tags.

Tiny Encryption Algorithm (TEA) is a lightweight cryptographic algorithm which makes it suitable for embedded systems that require high performance; ease of implementation, high speed, low power consumption and low cost beside security [3]. All these requirements need a simple cryptographic algorithm such as TEA. TEA can meet these requirements and has a resistance towards differential cryptanalysis; however, it has a simple key schedule which makes it weak against the related key attacks [4].

A modification in the standard TEA (MTEA) is proposed to improve the security strength of the TEA against attacks by using a Pseudo Random Number Generator (PRNG) such as Linear Feedback Shift Register (LFSR) where the MTEA's key is frequently changed in each round instead of using one key only for all rounds in the standard TEA [5].

TEA and MTEA is a fiestel structure network which depends on the confusion & diffusion properties as a substitution & permutation properties respectively. In [5], these properties are tested using completeness & avalanche effect properties which

result in good performance for MTEA rather than TEA algorithm.

In this paper, the proposed MTEA is implemented and we will point to several issues that need to be taken care of when implementing the design such as limitation of silicon area and power consumption in RFID systems.

This paper is organized as follows: Sections 2 present an introduction of the fiestel cipher. Sections 3 and 4 provide brief introductions to TEA and MTEA block cipher, respectively. Section 5 evaluates MTEA implementation compared to the standard TEA implementation. Finally, in section 6 we conclude the paper and draws guidelines for future enhancements.

2. The Fiestel cipher:

The Fiestel cipher structure is proposed to hide the statistical characteristic of the plaintext by using an alternative of substitution & permutation. This idea first proposed by Claude Shannon to develop the block cipher that alternates confusion & diffusion [6]. Shannon was concern about cryptanalysis depend on the statistical analysis of the plaintext that some messages in different languages have the frequency distribution such as letters, words or phrases which may be transferred to the ciphertext at the same frequency distribution. If the attacker has some knowledge about these statistics, maybe he can get the encryption key used to encrypt the plaintext.

Shannon suggests two strategies to hide the statistical characteristics of the plaintext: diffusion & confusion.

- ***Diffusion:*** its target to make the statistical analysis relationship between the plaintext & ciphertext as complex as possible to thwart any attempts from attacker to notice the key.
- ***Confusion:*** its target to make the statistical analysis relationship between the ciphertext & the encryption key as complex as possible to thwart any attempts from attacker to find the key.

In the Fiestel Cipher Structure: the concept of confusion & diffusion done by using substitution & permutation in the Fiestel Cipher Structure – sometimes called Fiestel Network – as shown in Figure 1.

- ***Substitution:*** is done on the left half of the plaintext after the function F is applied to the right half then; the output of it is XORed with the left half. The function F is the same structure in each round.
- ***Permutation:*** is done by interchanging the two halves of the plaintext every round.

Decryption process is the same as the encryption process except that the sub – key used with the function F is in the reverse order. The Feistel cipher structure is used in many symmetric block ciphers such as Data Encryption Standard (DES) & Tiny Encryption Algorithm (TEA).

Some ciphers use arithmetic operations as a diffusion and confusion technique, but this can significantly increase the area and power consumption. On the other hand, bit permutations in hardware can be realized with wires and no transistors are involved. They are therefore a very efficient component [7].

3. Tiny Encryption Algorithm

TEA is a block cipher with block length of 64 bits and key lengths of 128 bits. It is a Feistel type cipher as shown in Figure 2 which uses operations from mixed (orthogonal) algebraic groups. A dual shift causes all bits of the data and key to be mixed repeatedly. The key schedule algorithm is simple; the 128-bit key K is split into four 32-bit blocks (K[0], K[1], K[2], K[3]) and the 64-bit plaintext is split into two 32-bit blocks (Y_i and Z_i).

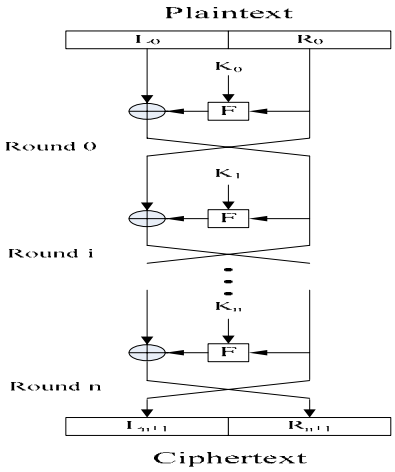

Figure 1. Feistel encryption network.

The TEA operations rely on the alternate use of XOR and ADD to provide nonlinearity. Addition and subtraction are used as the reversible operators for encryption and decryption rather than XOR. A dual shift causes all bits of the key and data to be mixed repeatedly. The number of rounds can be 16 cycles (32-iterations) or 32 cycles (64-iterations). The key is set at 128 bits as this is enough to prevent simple search techniques from finding the key [4].

The constant number, delta, is derived from the golden number ratio [4]:

$$\text{delta} = (\sqrt{5}-1) * 2^{31} = \text{9E3779B9h} \quad (1)$$

$$\text{delta}[i] = (i+1) * \text{delta}, \; i = 0,1,2,\ldots,31 \quad (2)$$

where delta is used to ensure that the sub keys are distinct and its precise value has no cryptographic significance.

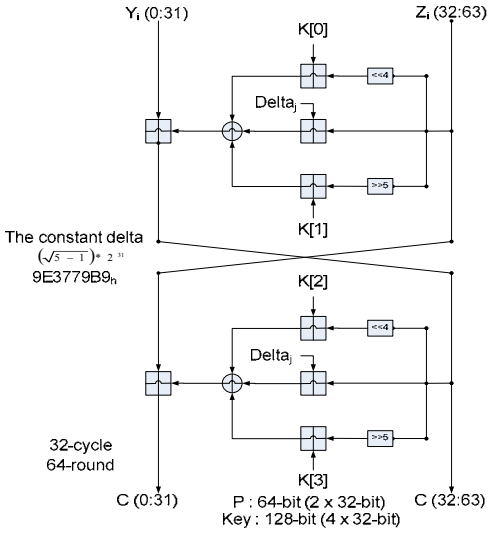

Figure 2. Tiny Encryption Algorithm block diagram

In Figure 4, using the input "Mohammad" which ASCII code is "4D 6F 68 61- 4D 4D 61 64" and the key "AbdallahOmarAyad" which ASCII code is "41 62 64 61 - 6C 6C 61 68 - 4F 4D 61 72 - 41 79 61 64"; the encrypted data is "43 64 A8 4A - 1E 55 9C 48" which will be used in the decryption side to recover the original plaintext "Mohammad"as sown in Figure 5.

4. The proposed modified TEA algorithm

In order to increase the TEA security against cryptanalysis, a PRNG is used to generate a new key every round. We propose to use LFSR as a PRNG as shown in Figure 3 for its ease of implementation using a software or hardware [5].

A feedback shift register is made up of two parts: a shift register and a feedback function. The shift register is a sequence of bits [8], the length of a shift register is figured in bits; if it is n bits long, it is called an n-bit shift register. Each time a bit is fed to LFSR, all of the bits in the shift register are shifted one bit to the right. The new left-most bit is computed as a function of the other bits in the register. The output of the shift register is one bit, often the least significant bit. The period of a shift register is the length of the output sequence before it starts repeating.

The simplest kind of feedback shift register is a linear feedback shift register. The feedback function is simply the XOR of certain bits in the register; the list of these bits is called a tap sequence.

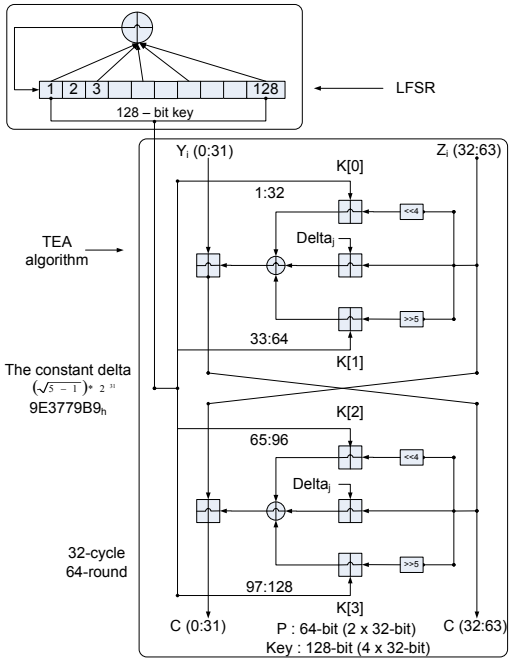

Figure 3. Modified Tiny Encryption Algorithm (MTEA) block diagram

An n-bit LFSR can be in one of 2^n - 1 internal state. This means that it can, in theory, generate a 2^n – 1 bit-long pseudo-random sequence before repeating. It is 2^n - 1 and not 2^n because a shift register filled with zeros will cause the LFSR to output a never ending stream of zeros—this is not particularly useful. Only LFSRs with certain tap sequences will cycle through all 2^n - 1 internal states; these are the maximal-period LFSRs. The resulting output sequence is called an m-sequence[8]

The characteristic polynomial of an LFSR generating a maximum-length sequence is a primitive polynomial where:

- number of 1s = number of 0s + 1
- same number of runs of consecutive 0s and 1s
- 1/2 of the runs have length 1
- 1/4 of the runs have length 2

It is proved that LFSRs achieve best performances by using one of the following polynomials [8-10]:

$$P_1(X) = 1+X^{64}+X^{78}+X^{123}+X^{128} \quad (3)$$

$$P_2(X) = 1+X^{32}+X^{47}+ X^{58}+X^{121}+X^{128} \quad (4)$$

$$P_3(X) = 1+X+X^2+X^7+X^{128} \quad (5)$$

One of the mentioned polynomials is used to generate 128-bit key in each round, so we have 32 * 128 – bits keys for each block.

It is worthy to mention that the key of the final round will be used as the key of the first round in the decryption process; hence this is the key that will be given to the users. This feature adds another advantage to the proposed technique as the keys given to the users, i. e. decryption keys, are different compared to the original algorithm where the same key is used in the encryption and decryption processes; hence the key secrecy is highly preserved.

In [5], the security strength of the proposed modified encryption algorithm (MTEA) is tested using two important security analysis criteria for block cipher; Completeness and avalanche effect tests. These tests are evaluated, compared to that of the standard TEA algorithm and proved that MTEA provides better strength as shown in table 1 and table 2.

Completeness test was done by using matrix of (64 x 64) elements with 65 plaintexts of length 64 bits, the difference between each plaintext and the next one is only one bit. Each plaintext is encrypted and its output – ciphertext – is XORed with the previous output from the previous plaintext. Finally, these results are written in the matrix until the matrix is completed. The number of ones in this matrix is added and then the percentage average value of this matrix is taken, this percentage determine how an algorithm can achieve the percentage of completeness effect that each bit of the ciphertext needs to depend on many bits on the plaintext [11]. This test is performed for the standard TEA as well as MTEA which uses 3 different polynomials P1, P2 and P3 in equations (3), (4) and (5). The results are shown in table 1.

Table 1. Completeness effect test evaluation

TEA	*Modified TEA (P_1)*	*Modified TEA (P_2)*	*Modified TEA (P_3)*
0.4836	0.5046	0.4968	0.4885

Avalanche effect test is done by encrypting 20 plaintexts that results in 20 ciphertexts; after that, the same 20 plaintexts are encrypted again but with only one bit difference in the key that is used in the previous test. Each two ciphertexts resulted from the same plaintext using the key & its one bit different version are XORed together. The number of ones from this XORing is calculated and then the percentage average of each result (number of ones / 64- bit) is obtained. Finally, the percentage average for all 20 results is calculated for evaluating the avalanche effect for an algorithm when an input/key is changed slightly, the output changes significantly [11]. This test is done for the standard TEA and MTEA which uses 3 different polynomials P1, P2 and P3 in equations (3), (4) and (5). The results are shown in table 2.

Moreover, it was proven that the performance degradation of MTEA could be neglected when the algorithm is implemented using MATLAB tool [5].

The remaining metric that shows the superiority of MTEA over standard TEA is actual Hardware

Table 2. Avalanche Effect Test Evaluation

Block #	*TEA*	*M TEA (P_1)*	*M TEA (P_2)*	*MTEA (P_3)*
1	42.18	51.56	53.12	48.43
2	45.31	48.43	48.43	65.62
3	46.87	62.5	42.18	43.75
4	56.25	62.5	48.43	50
5	46.87	37.5	45.31	53.12
6	46.87	50	54.68	50
7	57.81	56.25	53.12	50
8	50	54.68	53.12	64.06
9	48.43	54.68	53.12	48.43
10	48.43	50	50	57.81
11	45.31	54.68	57.81	62.5
12	57.81	65.62	48.43	42.18
13	48.43	48.43	45.31	48.43
14	56.25	57.81	46.87	46.8
15	50	51.56	40.62	57.81
16	42.18	48.43	54.68	46.8
17	37.5	57.81	54.68	51.56
18	56.25	62.5	48.43	50
19	50	51.56	57.81	56.25
20	39.06	50	50	37.5
Ave.	***31.1***	***34.45***	***32.2***	***33***

5. Hardware implementation of MTEA

Implementing cryptographic functionality on RFID tags is limited not only by the available silicon area, but also by the available power for computing cryptographic operations. Transponders basically operate as active or passive devices. The functionality of both types is similar; the main difference is the increased performance in view of communication distance and computation capabilities of the active vs. the lower cost of the passive transponders. The integrated battery increases the cost of the transponder, limits the tag's life time, causes environmental issues over disposal, and limits the form factor and thickness of the tag.

Most relevant for implementing cryptographic circuits are HF systems working on 13.56MHz and UHF systems operating at 900 MHz [12]. Both systems have in common that only a small fraction of the energy emitted at the interrogator's antenna is

received by the RFID tag. Both systems induce a voltage in the antenna of the RFID tag. The characteristics of HF and UHF fields define the constraints for energy and power consumption of RFID tags. When implementing cryptographic circuits on RFID tags it is desired that the cryptographic functionality does not limit the operating range. Thus, the tags have to satisfy with the limited power and energy budget. Before discussing these limits it is necessary to clarify whether power consumption or energy consumption is more important for passive RFID tags [13].

Hardware specifications of an encryption core for low-cost RFID tag are:

- Area Less than 0.25 milli-meter square (or 4,000 gates)
- Power Less than 30 micro watt
- Execution time 2.5 milli-second for encryption

- For a typical 0.35-µm CMOS technology, the maximum layout areas allowed translate to equivalent numbers of gates from 2,000 to 4,000 gates [14].

The proposed design (MTEA) is implemented using VHDL description language and evaluated with respect to the parallel architecture for the standard TEA in [2] as a suitable lightweight cryptographic algorithm for passive RFIID tag. Figure 6 and Figure 7 show the encryption and decryption results for MTEA.

In Figure 6, using the input "Mohammad" which ASCII code is "4D 6F 68 61- 4D 4D 61 64" and the key "AbdallahOmarAyad" which ASCII code is "41 62 64 61 - 6C 6C 61 68 - 4F 4D 61 72 - 41 79 61 64"; the encrypted data is "94 04 11 F8 - 2B 07 4A 4A". This output is provided to the decryption circuit but with using the last key generated from the encryption circuit, which is "D3 A2 6C 6A -20 B1 32 30 - B6 36 30 B4 – 27 A6 B0 B9", and the decrypted output is similar to the original data as shown in Figure 7.

We implemented MTEA and standard TEA using 0.35-µm CMOS technology which is the same technology used for the standard TEA in [2] using the same parallel architecture with the difference of adding LFSR in key generation. For the sake of fair comparison, we used the same synthesis tool as in [2], namely, Lenoardo Spectrum from Mentor Graphics Inc. the output of the synthesis tool is a pre-layout implementation using library primitives.

Table 3 shows the comparison between MTEA and standard TEA with respect to area, performance, throughput and power. The second column shows the results of the standard TEA implementation in [2]. The results are for post-layout implementation where no details are given for the post layout step. Therefore we compared the two algorithms using pre-layout implementation.

The power measurements shown in Table 3, with 3.3V supply [14], are for 51.2 kbps throughput, which is the requirement for 2.5-milli-second process time.

Table 3 shows that the area increase of MTEA over standard TEA is not significant, the maximum clock speed for the two algorithms is so close due to the fact that the added LFSR, in MTEA, runs in parallel and will not affect the critical path delay. Throughputs of the two algorithms are similar as MTEA is build upon the parallel architecture of the standard TEA; therefore they have the same number of clock cycles per iteration.

The throughput can be calculated from the equation (6) [2].

$$\text{Throughput} = \frac{\text{Data size} * \text{max. clock speed}}{\text{no. of clocks per cycle} * \text{no. of cycles}} \quad (6)$$

where data size is 64 bits, no. of clocks per cycle is 1 and no. of cycles is 32 cycles.

Regarding the power consumption, Prime Time tool from Synopsys Inc. is used to calculate the power as in [2]. The power consumption is divided into two main components: internal power and net switching power. The authors of [2] did not mention the nature of their power calculation; therefore we provided, in Table III, the total power as well as the net switching power. The Table shows that there is an increase in the power consumption in MTEA over the standard TEA. However, this increase is acceptable as the power still within acceptable range.

Table 3. Pre-layout synthesis results comparison

Algorithm	*Original TEA ENC [2]*	*TEA ENC*	*TEA DEC*	*MTEA ENC*	*MTEA DEC*
Pre-routed area (mm2)	0.207	0.262	0.3	0.337	0.346
Maximum clock speed (MHz)	53	66.8	69.1	67.1	64.3
Maximum throughput rate (Mbps)	106	133.6	138.2	134.2	128.6
Power (µW) at 51.2 (kbps) at 100 (KHz)	7.37	546.1	519.9	903.6	969.8
Net switching power (µW)	-	119.5	102.8	68.69	116.4
Power (µW) at 51.2 (kbps) at 1 (MHz)	-	650	626.7	1478	1072
Net switching power (µW)	-	176.3	161.3	375.2	171.6

It is worthy to mention that all the results are based on pre-layout implementations. We believe that better results would be obtained after performing the layout step.

6. Conclusion and future work

In this paper, a hardware implementation of the Modified TEA algorithm (MTEA) is proposed which uses the Linear Feedback Shift Register (LFSR) to overcome the security weakness of the standard TEA algorithm against attacks. The implementation of MTEA algorithm is benchmarked with the standard TEA algorithm considering the area, throughput and power consumption. The pre-layout synthesis results show that there is no significant degradation in the considered metrics due to using MTEA over standard TEA; hence MTEA is a good security candidate to be implemented in RFID systems.

In the future, we plan to perform the layout step for the proposed MTEA design to meet the limitations for silicon area, throughput and power consumption for RFID system.

7. References

[1] D. Tassos, "RFID Security and Privacy", *RFID Security: Techniques, Protocols and System-On-Chip Design*, Springer US, 2009.

[2] P. Israsena, "Design and Implementation of Low Power Hardware Encryption for Low Cost Secure RFID Using TEA", Fifth International Conference in Information, Communications and Signal Processing, Bangkok, Thailand, 2005.

[3] D. Issam, H. Samer, D. Hassan. "Efficient Tiny Hardware Cipher Under Verilog", Proceedings of the 2008 High Performance Computing and Simulation Conference, Nicosia, Cyprus, 2008.

[4] A. V. Reddy, "A Cryptanalysis of the Tiny Encryption Algorithm", Master of Science, Department of Computer Science in the Graduate School, The university of Alabama, 2003.

[5] M. Elmahallawy, M. B. Abdelhalim, M. Ayyad, A. Elhennawy. "Security analysis for a lightweight modified cryptographic TEA algorithm", 4th International Conference on Computer and Electrical Engineering ICCEE, Singapore, Malaysia, 2011.

[6] Shannon, C.E., " Communication Theory of Secrecy Systems ". Bell System Technical Journal, v. 28, n. 4, 1949, p. 656–715.

[7] C. Paar, A.Posshmann, and M.J.B. Robshaw, "New Design in Lightweight Symmetric Encryption", *RFID Security: Techniques, Protocols and System-On-Chip Design*, Springer US, 2009.

[8] S. Bruce, *Applied Cryptography*. 2nd edition, John Wiley and Sons, Inc, 1996.

[9] L. Cedric, "From Hardware to Software Synthesis of Linear Feedback Shift Registers", IEEE Parallel and Distributed Processing Symposium, Long Beach, CA, USA, 2007, pp. 1-8.

[10]H. Martin, J. Thomas, M. Alexender, M. Willi, "A Stream Cipher Proposal: Grain-128", IEEE International Symposium on Information Theory, Seattle, WA, USA,2006, pp.1614-1618.

[11]B. Kam, I. DAVIDA. "Structured Design of Substitution-Permutation Encryption Network", IEEE Transactions on Computers, vol. 28, no.10, 1979 , pp.747-753.

[12] C. Paar, A. Poschmann, M.J.B. Robshaw, "New Designs in Lightweight Symmetric Encryption", *RFID Security: Techniques, Protocols and System-On-Chip Design*. Springer US, 2009.

[13] S. E. Sarma, "Towards the 5 /c Tag", Technical Report MIT-AUTOID-WH-006. MIT, Auto-ID Center, 2001.

[14] Austria Micro Systems, 0.35µm CMOS Digital Standard Cell Databook, 2003.

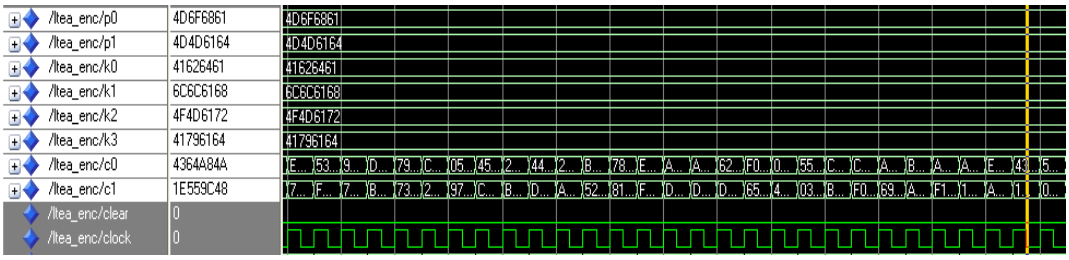

Figure 4. TEA encryption results

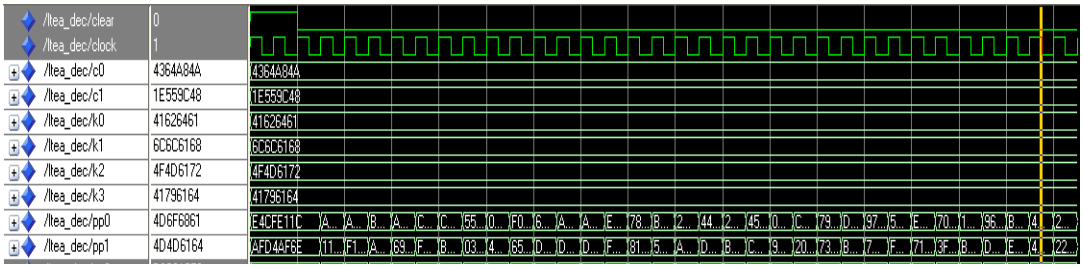

Figure 5. TEA decryption results

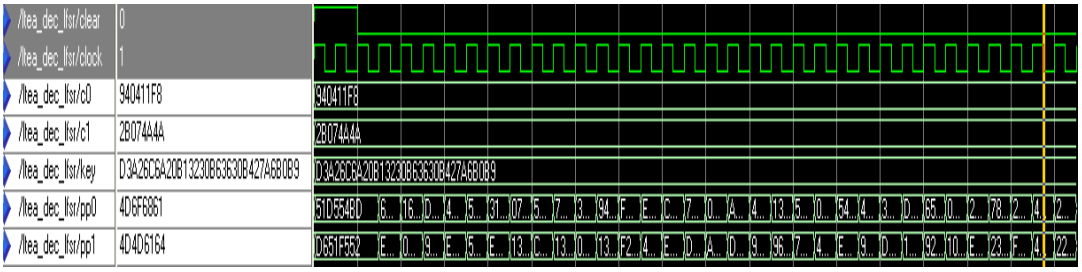

Figure 6. MTEA encryption results

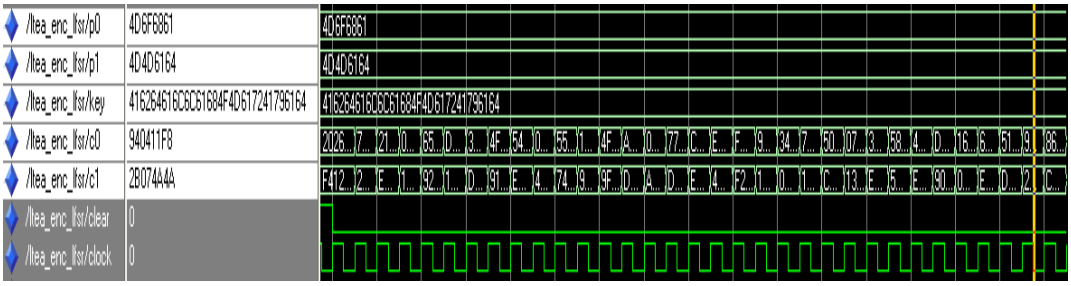

Figure 7. MTEA decryption results