**International Journal of Engineering Research and Development**
**e**-ISSN: 2278-067X, **p**-ISSN: 2278-800X, www.ijerd.com
Volume 6, Issue 11 (April 2013), PP. 88-91

# Behavioral Modification in Cellular Message Encryption Algorithm

Pawan Pandey[1], Prof. (Dr.) Ajay Rana[2]
[1]Lecturer, Raj Kmar Goel Institute of Technology, Ghaziabad (INDIA)
[2]Professor, Amity School of Engineering and Technology, Amity University Uttar Pradesh (INDIA)

**Abstract:-** This paper, first we describe the introduction of Cellular Message Encryption Algorithm, used for control channel encryption in Cellular Telephony scheme. Secondly why this algorithm is weak against security of cipher text being transmitted and finally how we can modify the Cellular Message Encryption Algorithm to be more secure for control data transmission.

**Keywords:-** CMEA, Plain Text, Cipher Text, Key, Block Cipher, T-Box.

## I. INTRODUCTION

CMEA [1] is not used to protect the voice over communication media but it is used to protect control channel data like numbers dialed by the user, bank account and credit card numbers etc. It uses 64 bit key with the variable length of block cipher (2-6 bytes block). It has three steps:

a) First left to right diffusion (statistical form of plain text) non-linear step.
b) Secondly performs subsequent changes using XOR operation right half to left half in linear manner.
c) Third again non linear step reverse of first step

Non-linearity is obtained using 8-bit table called T-box. Values in T-box are calculated as:

$T(n) = C((C(((C(((C((n \oplus K_0)+K_1)+n) \oplus K_2)+K_3)+n) \oplus K_4)+K_5)+n) \oplus K_6)+K_7)+n$

C is a 8-bit lookup table is called Cave Table. CMEA Algorithm used for encryption process as follows:

**Algorithm:** Cellular Message Encryption Algorithm
**Input:** $P_i$ (Plain Text)
**Output:** $C_i$ (Cipher Text)
**Step 1:** $y_0 \leftarrow 0$
**Step 2:** for $i \leftarrow 0,\ldots., n-1$
**Step 3:** $P_i' \leftarrow P_i + T(y_i \oplus i)$
**Step 4:** $y_{i+1} \leftarrow y_i + P_i'$
**Step 5:** for $i \leftarrow 0,\ldots,\lfloor n/2 \rfloor - 1$
**Step 6:** $P_i'' \leftarrow P_i' \oplus (P'_{n-1-i} \vee 1)$
**Step 7:** $z_0 \leftarrow 0$
**Step 8:** for $i \leftarrow 0,..n-1$
**Step 9:** $z_{i+1} \leftarrow z_i + P_i'$
**Step 10:** $C_i \leftarrow P_i'' - T(z_i \oplus i)$

In This Algorithm from step first completes from line no. 1 to 4, second step from line no. 5 to 6 and third step from 7 to 10. Here + and – are treated as addition and subtraction modulo 256. $\oplus$stands for logical bitwise exclusive or and $\vee$ stands for logical bitwise or [1].

## II. WHY IS CMEA WEAK

### A. Property 1:

If the plaintext is of the form P = {1-x,1-x,…1-x}And the cipher text is of the from C = {-x,...}, then there is a very high probability that T(0)=x, where T(i) is the output of the T-Box corresponding to input 'i' [2].

### B. Property 2:

If the plaintext is of the form P = {1-T(0), 1-T(0),….., 1-T(0), k-T(0),0}and the cipher text is C = {t-T(0),…} Where $k = ((n-1) \oplus j) - (n-2)$, then there is a very high probability that t = T(j) [2].

**C. Property 3:**

The CMEA algorithm uses a skewed CAVE Table [13]. The CAVE Table is not a permutation and 92 of the possible 256 values do not occur.

**D. Property 4:**

The CMEA algorithm uses a four round T-box which can be subjected to meet-in-the-middle attack [13].

**E. Property 5:**

The Least Significant Bit of the cipher text is always the complement of the Least Significant Bit of the plaintext. i.e. one bit always leaks [3].

**F. Property 6:**

The T-Box has some Key-Equivalence classes. As mentioned in Section 3.2, simultaneously complementing the Most Significant Bits of $K_{2i}$ and $K_{2i+1}$ for i = 0,1,2 leaves the action of the T-Box unchanged. This reduces the key length of the CMEA to 60 bits instead of 64 bits since we can assume the Most Significant Bits of $K_0$, $K_2$, $K_4$ and $K_6$ to all be 0 or 1 [4].

## III. MODIFICATIONS IN CMEA

**A. Modification 1:**

The update equation of P' needs to be changed so that Properties 1 and 2 do not work. Thus the modified equation is of the form:

$P'_i = P_i + T(y_i \oplus f(i,n)$

Such that as we vary i from 0 to n-1 (where n is the number of byte blocks in the plaintext) the T-Box is not predictably accessed. In the original CMEA property 1 exists because for a particular nature of the input plaintext and the key the T-Box was always referred at the point 0. So, the function f(i,n) should be such that the T-Box is accessed at different points. After considering several forms of the function f(i,n) the proposed function is f(i,n) = 2i%n, hence the update equation becomes :

$P'_i = P_i + T(y_i \oplus 2i \% n)$

Thus the algorithm is transferred to:

$y_0 = 0$
$for(i = 0; i < n; i++)$
{
$P'_i = P_i + T(y_i \oplus 2i\%n)$
$y_{i+1} = y_i + P'_i$
}

$for(i = 0; i < \lfloor n/2 \rfloor; i++)$
$P''_i = P'_i \oplus (P'_{n-i-1} \vee 1)$

$z_0 = 0$
$for(i = 0; i < n; i++)$
{
$z_{i+1} = z_i + P''_i$
$C_i = P''_i - T(z_i \oplus 2i\%n)$
}

**B. Modification 2:**

The CAVE Table is replaced with the AES S-box which can be efficiently implemented [9]. Thus the distribution is no more skewed and all the possible 256 values appear as a possibility.

**C. Modification 3:**

The T-box previously had 4 rounds. The number of rounds of the T-box has been increased to 8 rounds to prevent meet-in-the-middle attack. The output of the 4 round T-box is recycled again through the T-box.

**D. Modification 4:**

The or with 1 in the second stage of the CMEA is removed. This removes the property that the LSB of the cipher text is always the complement of the LSB of the plaintext. This can be explained by using the modified version and resorting to its truth table [11].

Using the modified version:

$C_0 = ((P_0 + T(0)) \oplus P'_2) - T(0)$

From the truth table (Table 1) we see that the Least Significant Bits of the plaintext and the cipher text are no longer related.

**Table 1:** Truth Table for LSB of plaintext and cipher text

| P[0] | T[0] | P'[2] | (P[0]xorT[0])XORp'[2]) | C[0] |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 |

### E. Modification 5:

From Table 2 we saw that by complementing the Most Significant Bits of $K_0$ and $K_1$ or a similar pair of odd and even keys, the output does not change [6]. However the Carry out in the 2 cases is different. In order to incorporate the effect of the carry from the MSB, it was exclusive or-ed with the Least Significant Bit of the resultant f(X). The Truth Table showing the carry over is given in Table 2.

For 50,000 random combinations of data and keys, we get the average number of changes in cipher texts (byte wise) by simultaneously changing the Most Significant Bits of all four pairs of Equivalence Class Keys as approximately 99.5% as opposed to 0% of the times earlier.

**Table 2:** Truth Table showing the carry 0 and 1

| | | | | CARRY = 0 | | CARRY = 1 | |
|---|---|---|---|---|---|---|---|
| | X | K0 | K1 | f(X) | New Carry | f(X) | New Carry |
| a1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| a2 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| | 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| | 1 | 1 | 1 | 1 | 0 | 0 | 1 |

## IV. CONCLUSIONS

In the present paper the original CMEA algorithm has been modified. It has been shown that the T-box provides sufficient security margin to the cipher CMEA-I in the face of linear and differential cryptanalysis. In short, the paper demonstrates that with suitable modifications the original CMEA algorithm can be made strong and hence can be suitable for wireless security.

## REFERENCES


[1] TIA TR45.0.A, "Common Cryptographic Algorithms", June 1995.

[2] D. R. Chowdhury, D. Mukhopadhyay, "Customizing cellular message encryption algorithm", International journal of Network security Vol.7, Issue 2, 2008.

[3] T. Chardin and R. Marinier, "Cryptanalysis of the Improved Cellular Message Encryption Algorithm ", Palaiseau, France.

[4] D. Wagner, B. Schneier, and J. Kelsey, "Cryptanalysis of the cellular encryption algorithm. In Jr., B.S.K., ed.: Advances in Cryptology", Springer, 1997.

[5] J. Daemen and V. Rijmen, "The Design of Rijndael", Springer-Verlag, 2002.

[6] B. Gladman, "Implementations of AES (Rijndael) in C/C++ and assembler", 2007. (http://fp.gladman plus.com/cryptography technology/rijndael)

[7] H. M. Heys, "A tutorial on linear and differential cryptanalysis", 2007. (www.engr.mun.ca/showard/PAPERS/ldc tutorial.ps).

[8] M. Matsui, "Linear cryptanalysis method for DES cipher", Advances in Cryptology (Eurocrypt'93), LNCS 765, pp. 386-397, Springer-Verlag, 1993.

[9] S. Morioka and A. Satoh, “An optimized S-box circuit architecture for low power AES design”, in Proceedings of Cryptographic Hardware and Embedded Systems, pp. 271-295, Springer-Verlag, Aug. 2002.
[10] S. Morioka and A. Satoh, “A 10-Gbps full-AES crypto design with a twisted BDD S-box architecture”, IEEE Transactions on VLSI Systems, vol. 12, no. 7, pp. 686-691, July 2004.
[11] D. Mukhopadhyay and D. RoyChowdhury, “An efficient end to end design of Rijndael cryptosystem in 0.18 mu CMOS”, 18th International Conference on VLSI Design, pp. 405-410, Jan. 2005.
[12] V. Rijmen, “Efficient implementation of the Rijndael-Sbox," 2007. (http://www.esat.kul euven.ac.be/ rijmen/ rijndael)
[13] D. Wagner, B. Schneier, and J. Kelsey, “Cryptanalysis of the cellular message encryption algorithm”, Crypto'97, pp. 526-537, 2002.