Bulletin de la Société Royale des Sciences de Liège, Vol. 85, 2016, *p.* 840 - 848

Presentation an effective algorithm to attack Common Scrambling Algorithm (CSA) in digital video broadcasting (DVB) system

Hamidreza DAMGHANI 1*, Saeed Ghazi MAGHREBI[2], Mohammadreza Moniri HAMEDANI[3]

1. Student of Master of Degree in the field of Communications Engineering, Islamic Azad University, Yadegar Imam Khomeini (Shahr-e-Rey) Branch, Tehran, Iran
2. Assistant Professor of Engineering – Communications Engineering Department, Azad University, Yadegar Imam Khomeini (Shahr-e-Rey) Branch, Tehran, Iran
3. Assistant Professor of Engineering – Communications Engineering Department, Azad University, Yadegar Imam Khomeini (Shahr-e-Rey) Branch, Tehran, Iran

Abstract

Common Scrambling Algorithm (CSA) is used for encoding the video streams in video player broadcasting system. For a more security margin, this algorithm is as a cascade combination of stream and block cipher. Common Scrambling Algorithm has been secure against the classic attacks, while it has been vulnerable against the side-channel ones. One of side-channel attacks on Common Scrambling Algorithm in the digital video broadcasting (DVB) will be studied in this paper. Descrambling the block part of this algorithm, the common key for the whole algorithm is gained. With this, a general approach for a high security encoding against the side-channel attacks is provided.

Keywords: Common Scrambling Algorithm (CSA), digital video broadcasting, block cipher, stream cipher, side-channel attack, security.

1. Introduction

DVB Common Scrambling Algorithm (CSA) is used in digital video broadcasting (DVB) system for securing the MPEG-2 video transport stream in each encrypted send of Pay-Tv in Europe. This algorithm was determined by European Telecommunications Standards Institute and was accepted by DVB Consortium in 1994 [1, 2, 3, and 4].

Common Scrambling Algorithm application is highly dependent on content send, but previous studies indicate that such independence has not been noticed enough. In fact, a practical attack on the Common Scrambling Algorithm in DVB system affects all content broadcasters and makes all transport stream descrambled. This algorithm uses cascade and combination of

[*] Hamidreza Damghani (Corresponding author): Hamidreza.damghani@gmail.com

stream cipher and block cipher for more security margin and both parts use a common key for being scrambled. Since the common scrambling algorithm details has been also clear to the public, it has been studied separately in [5] stream and block cipher together with a practical attack provided on the stream cipher. Weinman and Wirt in [5] showed that the block cipher part of common scrambling algorithm is sustainable against the linear and algebra descrambling attacks, like slide simple attack. In 2005, K. Wirt studied about fault of block cipher part of common scrambling algorithm which was the first attempt to study common scrambling algorithm against side-channel attack [6, 7 and 8]. K. Wirt recognized faults exists in scrambling and descrambling procedures in block part of common scrambling algorithm, obtained information related to the common key and finally could break the common scrambling algorithm completely.

In 1999, Differential Power Analysis (DPA) method was introduced by Kocher [9] as a side-channel attack on common scrambling algorithm. He showed that a successful attack on Data Encryption Standard (DES) method requires study and evaluation (unsuccessful attacks) only about hundreds of times. Moreover, attacker needs only one digital sampling oscilloscope and a standard computer for processing the measured values. Therefore, as described above, we are going to gain the common key for the common scrambling algorithm via studying the special features in block cipher structure of the common scrambling algorithm and using the Differential Power Analysis (DPA) method, and finally to break the common scrambling algorithm completely.

The rest of this article is categorized in this way: there will be a short review on the common scrambling algorithm in the next two sections. In the section 4, Differential Power attack on the common scrambling algorithm will be presented by studying the block part of the common scrambling algorithm. In section 5, we will finally provide a summary of the article.

2. Definitions

Definitions of symbols used in this paper are given below:

$\boldsymbol{K}$**:** The 64-bit common key used for scrambling the block part and stream part of the common scrambling algorithm.

$\boldsymbol{K^E}$: Shows Expanded key which is obtained from the Key schedule process in the block cipher part.

$\boldsymbol{SB_i}$: The i-th 8- byte block of the scrambled massage

$\boldsymbol{CB_i}$: Recognizes the i-th 8- byte block of the coding output of the common scrambling algorithm stream

$\boldsymbol{IB_i}$: Intermediate Blocks

$\boldsymbol{DB_i}$: The i-th 8- byte block of the Plaintext (Descrambled)

$\boldsymbol{DR\ and\ SR}$ **:** Respectively, remaining for Plaintext and Descrambled messages

$\boldsymbol{IV}$**:** Initialization vector for the scrambling part of the common scrambling algorithm stream

3. Review of the common scrambling algorithm stream

The common scrambling algorithm consists of two different coding cascade, stream coding and block coding parts, both of which use a 4-bit common key for scrambling.

Figure 1 depicts the scrambling process in the common scrambling algorithm [5]. In scrambling the block part of the common scrambling algorithm, an L-byte massage (plaintext) is divided into 8-byte blocks to create the temporary output ($\boldsymbol{IB_i}$) together with the common key in the Cipher Block Chaining (CBC) mode.

Notice that if the message length (L) is not a multiple of 8, there will be a remainder shown in figure 1 by *DR*. In ciphering the stream part of the common scrambling algorithm, block cipher output $\boldsymbol{IB_i}$ is used as an input for current cipher; together with the common key create the ciphered information of scrambling algorithm$\boldsymbol{SB_i}$.

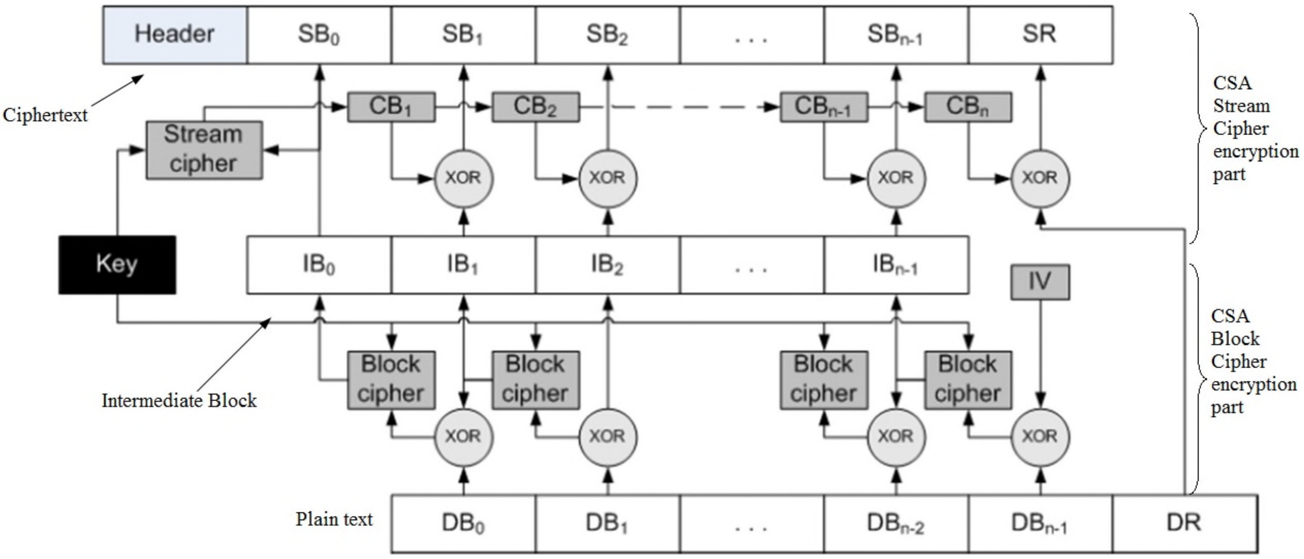

Fig. 1: Common scrambling algorithm for the digital video broadcasting system

4. Block cipher part of the common scrambling algorithm

Block cipher part of the common scrambling algorithm is the repeat of 56 block coding. This part consists of two inputs: a 64-bit block and a 64-bit running key, output of both will be a 64-bit block.

a. Round function

Suppose that $S = (s_0.s_1.\ldots.s_7)$ indicates the internal byte state in an arbitrary round. Function $\emptyset$ posses different internal states S, using the non-linear byte permutationπ^i and π', of the round i to the round $i+1$. These permutation are dependent to another permutation σ. It means that $\pi' = \sigma \circ \pi$. Symbol $\circ$ means that two function π' and π are related together through the permutation σ. In the other word, this symbol is the same combination function symbol *fog.* Bit permutation maps bit 0 to 1, bit 1 to 7, bit 2 to 5, bit 3 to 4, bit 4 to 2, bit 5 to 6, bit 6 to 0 and bit 7 to 3. Relation (1) shows arrangement of bits in the common scrambling algorithm coding:

$$\varphi(s_0 \ldots s_7.k) = (s_1.s_2 \oplus s_0.s_3 \oplus s_0.s_4 \oplus s_0.s_5.s_6 \oplus \pi'(k \oplus s_7).s_7.s_0 \oplus \pi(k \oplus s_7)) \qquad (1)$$

In relation (1), $\oplus$ means XOR. Reverse round of the common scrambling algorithm for descrambling of a message is gained by the relation (2):

$$\varphi^{-1}(s_0 \ldots s_7.k) = (s_7 \oplus \pi(s_6 \oplus \mathrm{k}).s_0.s_7 \oplus s_1 \oplus \pi(s_6 \oplus \mathrm{k}).s_7 \oplus s_2 \oplus \pi(s_6 \oplus \mathrm{k}). \\ s_7 \oplus s_3 \oplus \pi(s_6 \oplus \mathrm{k}).s_4.s_5 \oplus \pi'(s_6 \oplus \mathrm{k}).s_6) \qquad (2)$$

b. Common key schedule

Suppose that ρ is the bit permutation for a 64-bit message string. Expanded key $K^E = (k_0^E.k_1^E.\ldots.k_{447}^E)$ is obtained from the relation (3) recursively:

$$k_{0\ldots63}^E = k_{0\ldots63}$$

$$k_{64i\ldots64(i+1)}^E = \rho\left(k_{64(i-1)\ldots64i-1}^E\right) \oplus 0x0i0i0i0i0i0i0i0i.\, where\ 1 \le i \le 6 \qquad (3)$$

In this relation, $0x0i0i0i0i0i0i0i0i$ is a 64-bit hexadecimal constant value which function ρ becomes XOR per each i. For example, for $i = 3$, hexadecimal constant value will be $0x0303030303030303$. In this relation, function K^E is expanded as 64-bit from the main common key function (*K)* for the size of 448 bits. To do that, recursive steps are used. In order to create a non-conformity pattern, the number $0x0i0i0i0i0i0i0i0i$ is also used that by movement of i in the period of 0 to 6 a disorganized pattern in expanding the common key is reached and the previous data are used in maintaining its integrity.

c. Scrambling and descrambling

A plaintext $P = (P_0, \dots, P_7)$ provides a coded data $C = (C_0, \dots, C_7)$ by the function *Φ*. By using the reverse function *Φ*, Φ^{-1}, the coded data C is also descrambled. Relations (4) and (5) show coding and decoding respectively for both plaintext and coded data.

$$S^0 = P$$

$$S^r = \varphi\left(S^{r-1}, (k_{8r}, \dots, k_{8r+7})\right) \; for\ all\ 1 \le r \le 56 \tag{4}$$

$$C = S^{56}$$

$$S^0 = C$$

$$S^r = \varphi^{-1}\left(S^{r-1}, (k_{448-8r}, \dots, k_{445-8r})\right) \; for\ all\ 1 \le r \le 56 \tag{5}$$

$$P = S^{56}$$

5. Proposed attack on common scrambling algorithm

In this section, we offer a method for attacking on the common scrambling algorithm. For more description, different steps of the proposed attack are offered in the following. To do that, we study the block cipher part of the common scrambling algorithm by the help of Differential Power Analysis. It is supposed that the attacker focuses on descrambling the common scrambling algorithm. Its final result is the plaintext of the block cipher part which is the same plaintext of the common scrambling algorithm. Three steps should be carried out as follows:

Step1. Collecting the power calculation stages. The attacker observes *m* final descrambling of block cipher part and selects data sample by the relation (6) in each stage n.

$$T_i[j](1 \le i \le m \,.\, 1 \le \mathrm{j} \le \mathrm{n}) \qquad (6)$$

Step2. Selecting an estimation parameter. An important part of the process is how to select numbers of bits as the estimation parameter. As this parameter should be related to the plaintext and power calculation stages, if it is selected properly, we would act better to select function of the 3rd step. These bits can be selected by two methods as follows:

Step 2.1. Following steps are carried out in the descrambling process:

$$\begin{gathered} \mathrm{s}_0^{56} = \mathrm{s}_7^{55} \oplus \pi(\mathrm{s}_6^{55} \oplus \mathrm{k}_{0\ldots7}^{\mathrm{E}}) \\ \mathrm{p}_0 = \mathrm{s}_0^{56} \\ \mathrm{s}_7^{56} = \mathrm{s}_6^{55} \\ \mathrm{p}_7 = \mathrm{s}_7^{56} \end{gathered} \qquad (7)$$

Therefore, it can be written:

$$s_7^{55} = s_0^{56} \oplus \pi(s_6^{55} \oplus k_{0\ldots7}^{E}) = \mathrm{p}_0 \oplus \pi(\mathrm{p}_7 \oplus k_{0\ldots7}^{E}) \qquad (8)$$

We select some bits in s_7^{55} as the estimation bits. As $\mathbf{p}_7$ and p_0 are active, we can use the Brute Force attack on $k_{0\ldots7}^{E}$ and benefit the estimation bits s_7^{55} in the third step.

Step 2.2. In the descrambling process, following stages are carried out:

$$\begin{gathered} s_7^{56} = s_0^{55} \oplus \pi(s_7^{55} \oplus k_{440\ldots447}^{E}) \\ c_7 = s_7^{56} \\ s_6^{56} = s_7^{55} \\ c_6 = s_6^{56} \end{gathered} \qquad (9)$$

So it can be written:

$$s_0^{55} = s_7^{56} \oplus \pi(s_7^{55} \oplus k_{440\ldots447}^{E}) = c_7 \oplus \pi(c_6 \oplus k_{440\ldots447}^{E}) \qquad (10)$$

We select some bits in s_0^{55} as the estimation bits. As $\mathbf{c}_6$ and $\mathbf{c}_7$ are active, we can use the Brute Force attacker on $k_{440\ldots447}^{E}$ and benefit the estimation bits s_0^{55} in the third step.

Step 3. Data analysis in order to reduce noise

Step 3.1. After estimation parameter selection, we can find out the relation between active data and the key 8-bit part. Suppose that the selected function *D(b)* is a function of b_{th} bit of the key, as $1 \leq \mathrm{b} \leq 8$. The b_{th} bit is determined only by estimation. We suppose that the function *D(b)* is a Boolean function, in a way that if the b_{th} is zero, then value of the function is *D(b)=0*, otherwise *D(b)=1* . In this case, power orders *T_i[j]* are divided into two groups according to *D(b)* as follows:

$$\begin{gathered} |R_0| + |R_1| = m \\ |R_0| = \{T_i[j] | D(b) = 0\} \\ |R_1| = \{T_i[j] | D(b) = 1\} \end{gathered} \qquad (11)$$

Step 3.2. Calculating the average power for each stage. Subtracting the average of two series obtained through the relation (6) and (11), power derivative value of the stage$\Delta[j]$ is earned as follows:

$$\Delta[j] = \frac{1}{|R_1|} \sum_{T_i[j] \epsilon R_1}^{|R_1|} T_i[j] - \frac{1}{|R_0|} \sum_{T_i[j] \epsilon R_0}^{|R_0|} T_i[j] \qquad (12)$$

If the estimation bit is true, then value of the function *D(b)* with the probability 1 is equal to real value of the function. Following this, total power value of other data like error and noise calculation, which are uncorrelated with the function *D(b),* will be zero if $m \to \infty$. As power is correlated with the numerical value of data in flat and smooth area, the value will be maximum.

If the estimation bit is false, then the function *D(b)* will be uncorrelated with real data resulted from system. So grouping of R_0 and R_1 would be meaningless which couldn't create average power difference. As a result, $\Delta[j]$ will tend to zero with $m \to \infty$ and derivative of stages will be flat and smooth.

Step 3.3. Attacker will be able to calculate value of $k^E_{440...447}$ or $k^E_{0...7}$ after s^{55}_7 or s^{55}_0 are calculated in the step 2. So, the attacker would be able to obtain the common key recursively by the help of the above mentioned expanded key value, via the relation (3).

Complexity of the above stages are averagely only of the class 2^{11}.

6. Conclusion

In this paper, Differential Power Analysis (DPA) method was used to obtain the common key and break of the common scrambling algorithm. Some features of the block cipher part of the common scrambling algorithm were also mentioned which finally were used for breaking the whole algorithm. Since simultaneous descrambling of two block and stream coding parts is a difficult and time wasting process, obtaining some features of two block and stream coding parts and descrambling only one of them to reach the common key makes the process of attacking on the whole easier. On the other hand, it is not simple to expand an attack on the stream coding part of the common scrambling algorithm to obtain the common key, so it is highly necessary to study this part of the algorithm in order to find out on its weak points to attack on them.

References

[1] European Telecommunications Standards Institute. ETSI Technical Report 289, Support for use of scrambling and conditional Access (CA) within digital broadcasting systems, 1996.

[2] S. Bewich. ”Descrambling DVB data according to ETSI common scrambling specification”, UK Patent Applications GB2322994A /GB2322995A, 1998.

[3] D. D. Watts, R.S.P. Ashley, and K. G. Jacobus. ”System and apparatus for block wise encryption and decryption of data”, US Patent Application US5799089, 1998.

[4] Pseudonymous authors. ”CSA–known facts and speculations”, 2003, http://csa.irde.to

[5] R. P. Weinmann, K. Wirt., ”Analysis of the DVB common scrambling algorithm”, Eighth IFIP TC–6 TC–11, Conference on Communications and Multimedia Security, CMS 2004. Proceedings Kluwer Academic Publishers, 2004.

[6] K. Wirt. ”Fault attack on the DVB common scrambling algorithm”, Computational Science and Its Applications, 2005, pp. 577–584.

[7] D. Boneh, R. A. DeMillo, and R.J. Lipton, ”On the importance of checking cryptographic protocols for faults”, EUROCRYPT’97, LNCS 1233, Springer-Verlag, Berlin, 1997, pp. 37–51.

[8] E. Biham, A. Shamir. ”Differential fault analysis of secret key crypto systems”,Crypto 1997, LNCS 1294, Springer-Verlag, Berlin, 1997, pp. 513–525.

[9] P. Kocher, J. Jaffe, and B. Jun.,”Differential power analysis”, CRYPTO’99, LNCS 1666, Springer-Verlag, Berlin, 1999, pp.388–397.