CONSIDERATIONS ABOUT THE POSSIBILITIES TO IMPROVE AES S-BOX CRYPTOGRAPHIC PROPERTIES BY MULTIPLICATION

Florin MEDELEANU¹, Ciprian RACUCIU², Marius ROGOBETE³

¹Ministry of National Defense, Romania
²Titu Maiorescu University, Romania
³Alstom GRID Romania, Romania
E-mail: fmedeleanu@dcis.ro

The algorithm Rijndael was tested and chosen as the Advanced Encryption Standard (AES) in 2001, at the end of a security evaluation that lasted for years. Testing and evaluation process proved the algorithm’s strength and efficiency. The algorithm AES is a block cipher with a SPN (Substitution Permutation Network) structure. The cornerstone of the algorithm’s security is the substitution box (S-box) and its strength relies on its special algebraic construction which uses mixed operation in different order Galois Fields (base field and extended field). This paper is a study on the possibilities to improve some cryptographic properties of Rijndael Substitution box and the effect of these changes.

Key words: S-box, Galois Field factoring, non-linearity, algebraic expression.

1. INTRODUCTION

The algorithm Rijndael was developed by Joan Daemen and Vincent Rijmen, and was selected by National Institute of Standards and Technology (NIST) as standard encryption algorithm (Advanced Encryption Standard - AES) in 2001. The algorithm is a symmetric block cipher based on a substitution-permutation network (SPN). AES has long been a subject of interest for developers, cryptanalysts and the researchers, due to following:

– Security: the algorithm was extensively tested for long period of time (4 years, by NIST, in order to be adopted as a standard and 14 years, by the academic community, since the adoption as standard). During the evaluation, the algorithm was tested against linear and differential cryptanalysis attacks. The exceptional resistance against these attacks was obtained by so called „design by cryptanalysis” technique. This design approach means that the developers intended to create an algorithm especially resistant to some specific attacks, in this case linear and differential cryptanalysis attacks.

– Speed: AES is the fastest algorithm compared to other block algorithm offering the same level of cryptographic strength.

– Simplicity: The structure of the algorithm is easy to understand and can be easily decomposed in parts to be analyzed and evaluated.

2. THE STRUCTURE OF AES

The algorithm AES is fully described in FIPS-PUB-197 standard. For the sake of completeness, a brief description of the algorithm presenting the main parameters will also be given here. AES is a key-iterated block cipher composed of repeated application of round transformations on the data block. The block length size of AES is 128 bit length and there are three variants of the algorithm, with various key sizes: 128, 192 and 256 bits. The input and output of AES are considered one-dimensional arrays with the dimension of 8-bit bytes. For the encryption operation the inputs are the plaintext block and the key, and the output is the ciphertext block. For the decryption operation the inputs are the ciphertext block and the key, and the output...
is the plaintext block. The round transformation of AES operates on an intermediate result named \textit{state}. The state is interpreted as a rectangular matrix of bytes having 4 rows and a number of columns \(N_b\) which is equal to the block length (128) divided by 32, hence 4 columns. The key is mapped onto a two-dimensional cipher key. The cipher key is also a rectangular array with four rows similar to the state. The number of columns of the cipher key is denoted by \(N_k\) and is equal to the key length (128, 192 or 256) divided by 32, hence 4, 6 or 8 columns. The bytes of the key are mapped onto the bytes of the cipher key in the order: \(k_{0,0}, k_{1,0}, k_{2,0}, k_{3,0}, k_{0,1}, k_{1,1}, k_{2,1}, k_{3,1}, k_{0,2}…\)

\begin{figure}[h]
\centering
\begin{tabular}{cccc}
\hline
\(p_0\) & \(p_4\) & \(p_8\) & \(p_{12}\) \\
\hline
\(p_1\) & \(p_5\) & \(p_9\) & \(p_{13}\) \\
\hline
\(p_2\) & \(p_6\) & \(p_{10}\) & \(p_{14}\) \\
\hline
\(p_3\) & \(p_7\) & \(p_{11}\) & \(p_{15}\) \\
\hline
\end{tabular}
\end{figure}

\begin{figure}[h]
\centering
\begin{tabular}{cccccccc}
\hline
\(k_0\) & \(k_4\) & \(k_8\) & \(k_{12}\) & \(k_{16}\) & \(k_{20}\) \\
\hline
\(k_1\) & \(k_5\) & \(k_9\) & \(k_{13}\) & \(k_{17}\) & \(k_{21}\) \\
\hline
\(k_2\) & \(k_6\) & \(k_{10}\) & \(k_{14}\) & \(k_{18}\) & \(k_{22}\) \\
\hline
\(k_3\) & \(k_7\) & \(k_{11}\) & \(k_{15}\) & \(k_{19}\) & \(k_{23}\) \\
\hline
\end{tabular}
\end{figure}

Fig.1 – The state and the key \((N_b = 4\) and \(N_k = 6)\).

The AES has a number of rounds indicated by \(N_r\), which depends on the size of the key. For key sizes of 128, 192 or 256 bits, the number of rounds is respectively 10, 12 or 14. The AES encryption process is composed of the initial key addition, denoted by \textit{AddRoundKey}, followed by \(N_r - 1\) applications of the transformation \textit{Round}, and finally one application of \textit{FinalRound}. The initial key addition and every round take as input the \textit{State} and a round key. The round key for round \(i\) is denoted by \textit{ExpandedKey} \([i]\), and \textit{ExpandedKey} \([0]\) denotes the input of the initial key addition. The derivation of \textit{ExpandedKey} from the \textit{CipherKey} is denoted by \textit{KeyExpansion}.

The round transformation is denoted \textit{Round}, and is a sequence of four transformations, called steps. The final round of the cipher is slightly different, it is denoted \textit{FinalRound}, and is a sequence of three steps instead of four steps. Below, in the form of command script, there are described the operations of rounds that operates on arrays to which pointers \((\textit{State}, \textit{ExpandedKey}[i])\) are provided.

\begin{verbatim}
Round (State, ExpandedKey[i])
{
    SubBytes (State);
    ShiftRows (State);
    MixColumns (State);
    AddRoundKey (State, ExpandedKey[i]);
}

FinalRound (State, ExpandedKey [N_r])
{
    SubBytes (State);
    ShiftRows (State);
    AddRoundKey (State, ExpandedKey [N_r]);
}
\end{verbatim}

The transformation \textit{SubByte} (Substitution of Bytes) is the only non-linear step of the cipher. \textit{SubByte} is a layer of permutation consisting of S-box applied bytes of the state. The figure 2 describes the effect of \textit{SubByte} transformation to the state. The designers of the AES applied the following design criteria for S-box, as follows:

1. Non-linearity
   a) Correlation. The maximum input-output correlation amplitude must be as small as possible.
   b) Difference propagation probability. The maximum difference propagation probability must be as small as possible.

2. Algebraic complexity. The algebraic expression of Substitution of Bytes operation (S_{RD}) in GF(2^8) has to be complex.
Considerations about the possibilities to improve AES S-box cryptographic properties by multiplication

The AES S-box was chosen being defined by the following function in $\text{GF}(2^8)$:

$$g : a \rightarrow b = a^{-1}.$$ (1)

In the formula (1), the elements of $\text{GF}(2^8)$ are considered to be polynomials having a degree smaller than eight, with coefficients in the finite field $\text{GF}(2)$. By definition, the multiplication is done modulo the irreducible polynomial $m(x) = x^8 + x^4 + x^3 + x + 1$, and the multiplicative inverse $a^{-1}$ is defined accordingly. The value 00 is mapped onto itself. By definition, $g$ has a very simple algebraic expression, that is $g = x^{254}$. This simple expression could allow algebraic manipulations to be used to effectuate algebraic attacks on the cipher. Therefore, the authors of the AES built the S-box as the sequence of $g$ and an invertible affine transformation $f$. The affine transformation $f$ has no impact on the non-linearity properties, but if it is properly chosen, it allows $S_{\text{RD}}$ to have a complex algebraic expression. The authors have chosen an affine transformation that has a very simple description, but a complicated algebraic expression if it is combined with the transformation $g$.

3. THE ALGEBRAIC EXPRESSION OF THE AES S-BOX

In [1], the authors present five methods to find the algebraic expression of Rijndael S-box. These methods are as follows:

(1) Lagrange formula;
(2) Partition equivalence;
(3) Solve the equations of the polynomial base;
(4) Solve the dual trace of natural base;
(5) Solve the $q$-polynomial.

The authors of the present paper started to find the algebraic expression by using the Lagrange interpolation formula in finite fields. In order to do this, having in mind the high complexity of the calculus involved, the authors developed a program in Mathematica®. They noticed that for polynomial expression that has to approximate more than 13 points, the calculus complexity is too high and the program cannot cope with it. So they changed the approach and made the decision to find the algebraic expression of AES S-box through solving a linear system with coefficients in $\text{GF}(2^8)$. For this purpose, the authors also developed a program in Mathematica® to solve the linear equation system described below. An algebraic expression that describes any transformation in finite fields, including an S-box, has the general expression:

$$SB(x) = b_0 + b_1 \cdot x + b_2 \cdot x^2 + b_3 \cdot x^3 + \ldots + b_{253} \cdot x^{253} + b_{254} \cdot x^{254} + b_{255} \cdot x^{255}.$$ (2)

By setting all possible values for $x \in \text{GF}(2^8)$ and considering the coefficients $\{b_0, b_1, \ldots b_{255}\}$ as variables, it is obtained the linear system with coefficients in $\text{GF}(2^8)$:
Many papers, including [1] and [2] give the expression of the AES- GF S-box, in the form:

$$f_{RD}(x) = 05 \cdot x^{254} + 09 \cdot x^{253} + F9 \cdot x^{251} + 25 \cdot x^{247} + F4 \cdot x^{239} + 01 \cdot x^{223} + B5 \cdot x^{191} + 8F \cdot x^{127} + 63 \quad (4)$$

Solving the linear system (3) the coefficients $$b_0, b_1, \ldots, b_{255}$$ of algebraic expression of the S-box are found easily. For example, solving the system (3) for unmodified AES S-box, the coefficients $$\{b_0, b_1, \ldots, b_{255}\} = \{A256[99],0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
We can notice that for \( f_{RD}(239)(x) \) there are involved 254 terms.

### 5. RESULTS

After solving the system (7) for any \( i \in \{1,2,\ldots,254\} \), the authors noticed that:

(i) For \( i \in \{1,2,4,8,16,32,64,128\} \), the number of terms involved in the resulting algebraic expression remains unchanged, that is 9;

(ii) For \( i \in \{1,2,\ldots,254\}\setminus\{1,2,4,8,16,32,64,128\} \), the number of terms involved in the resulting algebraic expression increases, being larger than 91 with a maximum of 254;

(iii) However, only 128 of the resulting \( f_{RD}^{(i)}(x) \) are permutations, that is almost half;

(iv) The authors considered the general expression (2) having 256 terms. But in GF(2^8) for any value of \( x \), excepting the null value, the following expression holds:

\[ x^{255} = 1. \] (8)

As consequence, the coefficient \( b_{255} \) found by solving the linear system (7) is null for any \( i \in \{1,2,\ldots,254\} \). This was expected because the value of the coefficient \( b_{255} \) can be included in the value of \( b_0 \) in all equations of (7), due to (8), but not in the first equation. This exception explains the null value for \( b_{255} \).

#### Table 1

<table>
<thead>
<tr>
<th>( \frac{i}{k} )</th>
<th>( \frac{1}{9} )</th>
<th>( \frac{29}{9} )</th>
<th>( \frac{49}{9} )</th>
<th>( \frac{791}{9} )</th>
<th>( \frac{89}{9} )</th>
<th>( \frac{993}{9} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/9</td>
<td>14/9</td>
<td>16/9</td>
<td>19/9</td>
<td>22/9</td>
<td>23/163</td>
<td>26/92</td>
</tr>
<tr>
<td>29/9</td>
<td>29/163</td>
<td>31/218</td>
<td>329/3</td>
<td>37/93</td>
<td>38/93</td>
<td>41/93</td>
</tr>
<tr>
<td>49/9</td>
<td>44/9</td>
<td>46/163</td>
<td>47/217</td>
<td>49/93</td>
<td>52/92</td>
<td>53/162</td>
</tr>
<tr>
<td>89/9</td>
<td>71/163</td>
<td>73/93</td>
<td>74/93</td>
<td>76/93</td>
<td>77/162</td>
<td>79/219</td>
</tr>
<tr>
<td>993/9</td>
<td>83/162</td>
<td>86/163</td>
<td>88/93</td>
<td>89/163</td>
<td>91/219</td>
<td>92/163</td>
</tr>
<tr>
<td>101/163</td>
<td>97/93</td>
<td>98/93</td>
<td>101/163</td>
<td>103/219</td>
<td>104/92</td>
<td>106/162</td>
</tr>
<tr>
<td>113/163</td>
<td>108/219</td>
<td>112/91</td>
<td>116/163</td>
<td>118/219</td>
<td>121/217</td>
<td>122/219</td>
</tr>
<tr>
<td>121/163</td>
<td>124/218</td>
<td>127/254</td>
<td>128/9</td>
<td>131/91</td>
<td>133/93</td>
<td>134/92</td>
</tr>
<tr>
<td>139/163</td>
<td>139/163</td>
<td>142/163</td>
<td>143/218</td>
<td>146/93</td>
<td>148/93</td>
<td>149/163</td>
</tr>
</tbody>
</table>
In Table 1 there are given the results after solving the systems of equations (7) for any \( i \in \{1,2,\ldots,254\} \). In this table \( i \) represent the order of multiplication for elements of standard AES S-box, and \( k \) represents the number of terms for corresponding algebraic expression. The missing orders of multiplication (e.g. \( k \in \{3,5,6\ldots,252\} \)) correspond to noninvertible maps which are not appropriate for building S-boxes, and they are not listed here.

Taking into consideration additional requirement for an S-box, given in [2], that are non-linearity and difference propagation probability, the authors noticed that the non-linearity of algebraic expressions (i) remains 112 and difference propagation probability is \( 2^6 \), but for algebraic expressions (ii) the non-linearity decreases under 100 (in average 96) and difference propagation probability is higher than \( 2^6 \).

6. CONCLUSIONS

Even if the proposed purpose of S-box multiplication (that is to increase the number of terms involved in the algebraic expression of S-box) was attained for some powers of \( i \), the authors noticed that the other design criteria for S-box deprecated. These deprecated properties are non-linearity and difference propagation probability. This study concludes that it is not possible to improve all design criteria for S-box at the same time by multiplication.

REFERENCES