Application of the Codes of a Polynomial Residue Number System, Aimed at Reducing the Effects of Failures in the AES Cipher

Elena Pavlovna Stepanova, Igor Anatolyevich Kalmykov, Ekaterina Viktorovna Toporkova
Maksim Igorevich Kalmykov
Konstantin Aleksandrovich Katkov
North-Caucasus Federal University
Stavropol, Russia
1 Pushkina str.,
Stavropol, 355009 Russia
Denis Nikolaevich Rezenkov
Stavropol State Agrarian University
Stavropol, Russia
12 Zootekhnicheskiyper.,
Stavropol, 355017 Russia

JDIM
Journal of Digital Information Management

***ABSTRACT**: The aim of the work is to increase the reliability of the AES cipher by means of development and application of redundant codes of a polynomial residue number system (PRNS) that are able to correct the errors caused by failures.The known methods of counteracting failures do not take into account the specificities of the AES cipher, which leads to significant hardware costs. The problem can be solved through the use of error-correcting codes of a polynomial residue number system. However, it is impossible to use the known methods of search and correction of errors by the PRNS codes in the AES cipher. Therefore, the development of a method, the use of which will give the AES cipher the robustness property with respectto failures through the use of the PRNS codes, is a topical problem.*

Subject Categories and Descriptors
[F.2.1 Numerical Algorithms and Problems]; Number-theoretic computations; **[B.1.5 Microcode Applications]; [G. Mathematics of Computing];** Error analysis

General Terms:
AEScipher, Code Checking, Number Computations

Keywords: AES cipher, Failures, code of a polynomial residue number system, Error correction, Positional characteristics

Received: 1 November 2015, Revised: 27 December 2015, Accepted: 30 December 2015

1. Introduction

Currently, there is an increased interest of developers in the SPN ciphers that use a Substitution-Permutation Network. These codes have a good combination of such indicators as cryptographic robustness, efficiency, and relatively low hardware costs. However, during the operation of the encoder of the SPN ciphers, someequipment failures may occur. These failures can be both natural and anthropogenic. This will result in distortion of the result of encryption. The known methods of countering the effects of failures do not address the specific character of the SPN ciphers, which leads to significant hardware costs. This problem can be solved through the use of error-correcting codes of a polynomial residue number system.

In the first part of the article, we analyze the main methods of increasing the reliability of the AES encryption devices, including the methods allowing reducing the hardware costs. The possibility of implementation of the AES algorithm in the PRNS code is shown.

The second part is devoted to the development of new principles of constructing the corrective PRNScodes. The results of studying the corrective capacity of such codes are presented. It is shown that the PRNS codes with one control base allow only detecting errors. An algorithm is developed that enables to correct errors using a PRNS code. Examples of implementation of the algorithm in the replacement of Sub Bytes are given.

The following are the results of a comparative analysis of the developed error correction algorithm with the "2 *out of* 3" method of masking failures;there are also shown the prospects of development of the research topic.

2. Methods

2.1 Analysis of the main methods of counteracting failures in the AES cipher

One of the frequently used SPN ciphers is the AES cipher. This is connected with the fact that AES is ideally fit for the embedded data protection systems. Among the advantages of the AEScipher there are the following ones (Schneieret al., 2000;Nechvatalet al.,2000):

- Highefficiency on all platforms;
- High level of security;
- It can be easily implemented in the smart-cards.

However, the AES cipher does not possess the robustness property with respect to faults and failures that may occur during the operation of the cipher. Among the destructive influences of natural character on the AES cipher, the following ones may be indicated:

- Failures in the electrical power system;
- Changeinthesupplyvoltageoftheencoder;
- Changeinthe clock frequency of the encoder.

A malperformance of the encoder can be caused not only by the influences of natural character, but also by the attacks on the AES cipher. It is known that one of the effective methods of cryptanalysis is the attacks on the AES using the method of differential cryptanalysis on the basis of failures (Blomer,& Seifert,2003; Peacham, &Thomas, 2006). The attacksbased on faults (the fault attacks) are based on various specific impacts on the encoder in order to disrupt its normal functioning, causing the possibilities of failures in the encoder functioning (Parket al., 2011).

In the work (Bar-El et al., 2004), the main techniques and methods to counter failures in the process of the operation of the AES cipher are presented, among which one can distinguish:

- The introduction into the encoder of detectors of various influences, which would block the encoder in the event of detecting an influence;
- The use of the passive shielding of the encoder;
- various kinds of duplication of calculations with comparing of the results;
- The use of check-summing of some data fragments;
- Introduction into the program of random redundant computations.

However, these methods do not take into account the specificity of the AES cipher, resulting in the solutions with significant costs.

1.2 Implementation of the AES cipher in a Polynomial Residue Number System

Let us consider the ways to reduce the hardware costs for the implementation of the AES cipher, which will also reduce the number of failures. The basis of the first group is comprised by the methods of synthesis of the resulting logic function. For example, in (Song, 2003), the obtaining of the desired logic function in the PDNF or PCNF form is demonstrated. As a result, an optimized structure of the S-Box permutation blockis obtained.

Those methods are put into the basis of the second group that allows reducing the calculations in the field GF(2^8) to calculations in the finite fields of lower order. Such technique is considered in detail in (Akashi, 2001) and tested in the works (Mangardet al., 2003; Hodjat, &Verbauwhede, 2004; Nataleet al., 2007). The essence of it is reduced to the representation of elements of the field GF(2^8)in the form of elements of the field GF(2^4).

As is known, the AES cipher works with the bytes $S(j)$, considered as elements of the Galois field GF(2^8) with the generating polynomial $p(x)=x^8+x^4+x^3+x+1$. After transition to the ring of irreducible polynomials $p_1(x)=x^4+x+1$ and $p_2(x)=x^4+x^3+1$, which generate the finite Galois fields GF(2^4), the bytes of the input and intermediate data $S(j)$ are represented in the form of two residues $s_1(j) \equiv S(j) \bmod p_1(x)$ and $s_2(j) \equiv S(j) \bmod p_2(x)$.

Since in the foundation of cryptographic transformations of the AES cipher there are the operations of addition and multiplication modulo $p(x)=x^8+x^4+x^3+x+1$, they can be substituted by the corresponding operations over the residues $s_1(j)$ and $s_2(j)$ in the ring of polynomials $p_1(x)=x^4+x+1$ and $p_2(x)=x^4+x^3+1$. Thus, in the realization of the AES cipher, a PRNS code is used with two working bases.

In the PRNS code, the positional binarycode is represented in a polynomial form, and then the following collection of residues is put into correspondence to this polynomial (Duquesne,2011;Chu,& Benaissa, 2009;Gordenko*et al.*, 2014).

$$A(x)=(a_1(x),a_2(x),\ldots a_k(x)), \qquad (1)$$

where $a_i(x) \equiv A(x) \bmod p_i(x)$; $i=1,\ldots,k$.

This collection of bases of the PRNS code forms a working range of the system

$$P(x)=\prod_{i=1}^{k} p_i(x), \qquad (2)$$

Since the congruences with respect to one and the same modulus can be added term by term, then for two polynomials $A(x) = (a_1(x), a_2(x),..., a_k(x))$ and $B(x) = (b_1(x), b_2(x),..., b_k(x))$ the following is true

$$\left|A(x)+B(x)\right|^{+}_{p(x)} = \left(\left|a_1(x)+b_1(x)\right|^{+}_{p_1(x)},...,\left|a_k(x)+b_k(x)\right|^{+}_{p_k(x)}\right), \quad (3)$$

$$\left|A(x)\circ B(x)\right|^{+}_{p(x)} = \left(\left|a_1(x)\circ b_1(x)\right|^{+}_{p_1(x)},...,\left|a_k(x)\circ b_k(x)\right|^{+}_{p_k(x)}\right), \quad (4)$$

,

$$\left|A(x)\cdot B(x)\right|^{+}_{p(x)} = \left(\left|a_1(x)\cdot b_1(x)\right|^{+}_{p_1(x)},...,\left|a_k(x)\cdot b_k(x)\right|^{+}_{p_k(x)}\right). \quad (5)$$

where + and o are the operations of addition and subtraction modulo p.

The parallel data processing with respect to the PRNS bases allows providing high-speed execution of these modular operations. Thanks to this feature, the modular codes are widely used in the devices of real-time signal processing (Katkov, & Kalmykov, 2013; Kalmykov*et al.*, 2014).

The AES cipher works with the bytes, which are considered as the elements of the field $GF(2^8)$, where the generating polynomial $p(x) = x^8 + x^4 + x^3 + x + 1$. Each round of the AES encryption consists of the transformations:

- Substitution of the bytes, Sub Bytes;
- Byte-at-a-time shift of rows, Shift Rows;
- Mixing of columns, Mix Columns;
- Summation with the round key, Add Round Key.

To provide failure tolerance, a direct PN-PRNS converter, a PRNS-PN inverter, and an error correction block are introduced into the encryption module.

It is known that the transformation SubBytes requires significant hardware costs, which amount from 84% to 90% of the total hardware costs for the encoder. Let us consider the application of the PRNS code in conducting the SubBytes transformation. A byte of a clear text S arrives at the input of the converter from PC to the PRNS code, from the output of which the residues $s_1(x)$ and $s_2(x)$ are read off, where $s_1(x) \equiv S(x) \bmod p_1(x)$ and $s_2(x) \equiv S(x) \bmod p_2(x)$.

Thus, the byte $S(x)$ has the form of two four-digit blocks of data, represented with respect to the PRNS modules. In this form, the data come to the Sub Bytes converter. Moreover, $s_1(x)$ defines the column number, whereas the row number is given by the residue $s_2(x)$. Then the substitution table of Sub Bytes, which required a memory block of 256x8 bits, is represented now in the form of two tables of 256x4 bits. In the tables, there are the residues of the substitution $s'(x)$, reduced with respect to the moduli $p_1(x) = x^4 + x + 1$ and $p_2(x) = x^4 + x^3 + 1$. In Table 1 there are given the first three rows of the substitution table of S_1-block modulo $p_1(x) = x^4 + x + 1$.

Let us assume that a byte of the current state comes to the inputs of the S-block. The 4 high-order bits of the byte define the number of the table row, whereas the 4 low-order bits define the column number. For example, when the input is the state $\{00011001_2\} = \{19_{16}\}$, then the result at the output of the S-block will be $\{d4_{16}\} = \{11010100_2\}$, which is situated at the intersection of the 1st row and the 9th column.

Consider the application of the PRNS code in the operation of the S-block. Let $S = \{00011001_2\} = \{19_6\}$. This value comes to the input of the direct PN-PRNS converter, on the output of which there will be

$$s_1(x) = S(x) \bmod x^4 + x + 1 = \{19_{16}\}$$

$$\bmod x^4 + x + 1 = x^4 + x^3 + 1 \bmod x^4 + x + 1 = x^3 + x = A;$$

$$s_2(x) = S(x) \bmod x^4 + x^3 + 1 = \{19_{16}\}$$

$$\bmod x^4 + x^3 + 1 = x^4 + x^3 + 1 \bmod x^4 + x^3 + 1 = 0.$$

$$\bmod x^4 + x^3 + 1 = 0$$

The residues $S\ (x) = (A,\ 0)$ come to the inputs of the substitution table of the S_1-block modulo $p_1(x)=x^4+x+1$ and the S_2-block modulo $p_2(x)=x^4+x^3+1$. The result of the substitution is determined from Tables 1 and 2. In Table 1, at the intersection of the column **A** and the row 0, there is the number 0. In Table 2, at the intersection of the column **A** and the row 0, there is the number 5. As a result of action of the state byte $S = \{ 00011001_2\} = \{ 19_{16}\} = (A, 0)$, a substitution byte is obtained, equal to $S'(x) = (0, 5)$ in PRNS. This corresponds to the substitution $S'(x) = \{d4_{16}\} = \{ 11010100_2\}$. Let us check it

$$s_1'(x) = \{d4_{16}\} \bmod x^4 + x + 1 =$$

$$x^7 + x^6 + x^4 + x^2 \bmod x^4 + x + 1 = 0;$$

$$s_2'(x) = \{d4_{16}\} \bmod x^4 + x^3 + 1 =$$

$$x^7 + x^6 + x^4 + x^2 \bmod x^4 + x^3 + 1 = x^2 + 1 = 5.$$

The transition from the processing of the bytes $S(j)$ modulo $p(x) = x^8 + x^4 + x^3 + x + 1$ to the ring of polynomials $p_1(x)=x^4+x+1$ and $p_2(x)=x^4+x^3+1$ can be used for the development of new principles of constructing the correcting PRNS codes. As a control base, we will use the polynomial $p_3(x)=x^4+x^3+x^2+x+1$. For the development of new principles of constructing the

$s_2(x)$	Residue $s_1(x)$ modulo $p_1(x)=x^4+x+1$															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	9	9	2	F	D	B	B	5	B	F	0	1	4	3	F	9
1	D	5	9	0	9	3	4	A	4	F	6	0	4	8	9	1
2	A	2	E	5	A	4	6	C	2	4	D	6	6	D	F	8

Table 1 .Substitution table of S$_1$-block modulo $p_2(x)=x^4+x^3+1$

In Table 2 there are given the first three rows of the substitution table of S$_2$-block modulo $p_2(x)=x^4+x^3+1$

$s_2(x)$	Residue $s_1(x)$ modulo $p_2(x)=x^4+x^3+1$															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	7	F	8	3	5	C	B	8	6	4	5	4	E	B	C	B
1	B	1	4	6	D	9	B	F	D	F	A	1	6	B	C	2
2	2	0	A	7	F	4	3	2	3	5	E	B	1	9	9	9

Table 2. Substitution table of S$_2$-block modulo $p_2(x)=x^4+x^3+1$

correcting PRNS codes, let us examine their correcting capacities.

2. Development of a method of searching and correcting errors in the PRNS code

2.1 Mathematical foundations of error correction using the PRNS codes

A qualitative leap in providing failure tolerance for the AES cipher is to use the redundant PRNScodes.The parallel and independent processing of residues is an ideal foundation for the correction of errors occurring due to the system malfunctions (Gordenkoet al., 2014). In this case, no data exchange takes place between the modules in the PRNS codes. This property of the PRNS codes is used for detection and correction of errors. However, to do this, it is necessary to introduce the control bases.

The introduction of *r* control bases of PRNS, which must satisfy the condition

$$\deg p_{k+r} \geq \deg p_{k+r-1} \geq \ldots \geq \deg p_{k+1} \geq \deg p_k \geq \deg p_{k-1} \ldots, \qquad (6)$$

leads to expansion of the working range to a full range

$$P^*(x) = \prod_{i=1}^{k+r} p_i(x) = P(x) \prod_{i=k+1}^{k+r} p_i(x). \qquad (7)$$

if *A(x)* belongs to the working range, that is

$$\deg A(x) < \deg P(x), \qquad (8)$$

then the code $A(x)=(a_1(x), a_2(x),..., a_{k+r}(x))$ will be considered allowable. Otherwise, the PRNS code contains errors.

If an error appeared with respect to the *i*-th base of the PRNS code, then the erroneous code equals

$$A^*(x) = (a_1(x), ,...., , a_{i-1}(x), a_i(x)+\Delta a_i(x), a_{i+1}(x),..., a_{k+r}(x)),$$

where $\Delta a_i(x)$ is the depth of the error with respect to the *i*-th base.

According to the Chinese remainder theorem, in the transference to the positional notation (PN), we will have

$$A^*(x) = \sum_{\substack{j=1 \\ j\neq i}}^{k+r} (a_j(x)B_j(x) + (a_i(x)+\Delta a_i(x))B_i(x)) \bmod P_{\text{full}}(x). \qquad (10)$$

Let us simplify the equality

$$A^*(x) = \sum_{\substack{j=1 \\ j\neq i}}^{k+r} (a_j(x)B_j(x) + (a_i(x)+\Delta a_i(x))B_i(x)) \bmod P_{\text{full}}(x) =$$

$$= A(x) + \Delta a_i(x)B_i(x) \qquad (11)$$

Ananalysis of the equality (11) shows thatthe error leads to the inequality

$$\deg A^*(x) > \deg P_{\text{work}}(x).. \qquad (12)$$

This property of PRNS is used for searching and correcting errors in the code.

2.2 Algorithms for calculating the positional characteristics of the PRNS codes

Since the PRNS code is not positional, it is impossible to find out whether the inequality (12) holds judging from the code's appearance.To solve this problem, one uses positional characteristics (PC), which will allow determining the "*location*" of the code combination relative to the working range. In addition, the value of PC is determined by all residues, calculated with respect to the bases of PRNS.

One of the most popular PC is the interval number (Berezhnoyet al., 2004;Gapochkin et al., 2014; Kalmykov et al., 2015). In the works (Hu Zhengbing, et al., 2015;Yatskivet al., 2013), the algorithms are presented that allow correcting errors in the PRNS code using expansion of the system of bases. The error syndrome is defined as the difference between the redundant residues $a_{k+1}(x), a_{k+2}(x)$ of the PRNS code $A(x) = (a_1(x), a_2(x),..., a_{k+2}(x))$ and the result of calculation of the residues $a_{k+1}(x), a_{k+2}(x)$ using the working bases. Another characteristic, which can be used to correct an error in the PRNS code, is a trace. The corresponding calculation algorithm is given in (Gordenkoet al., 2014). The paper (Duninet al., 2014) presents an error correction algorithm which uses a positional characteristic: the coefficients of a mixed-radix system (MRS).

Using these PC in the redundant PRNS codes allows detecting errors. This is connected with the fact that to search for a singl eerror with the help of the PRNS code, one redundant base suffices. Since in the AES cipher the polynomials $p_1(x)=x^4+x+1$ and $p_2(x)=x^4+x^3+1$ representthe elements in GF(2^8), then we use as the control base the polynomial $p_3(x)=x^4+x^3+x^2+x+1$.

In the work (Chu, & Benaissa, 2013) it is suggested to use the PRNS code for the detection of errors, appearing due to failures in the AES encryption. In addition, as the working bases of PRNS, the author proposes to use $p_1(x)=x^4+x+1$ and $p_2(x)=x^4+x^3+1$; as a control base he chooses $p_3(x)=x^4+x^3+x^2+x+1$.

As a result of this, the implementation of the SubBytes transformation has the form of three tables of the size 256 x 4 bits. It is indicated in the work that such approach allows detecting 100% of all single errors and up to 93.75% of multiple errors, arising due to the failures in the operation of the encoder.

To detect an error, the author implements an inverse PRNS-PC transformation with the subsequent comparison with $p(x)=x^8+x^4+x^3+x+1$. This transformation can be implemented on the basis of an intermediate MRS

$$A(x) = v_1(x) + v_2(x)p_1(x) + v_3(x)p_1(x)p_2(x) = v_1(x) + v_2(x)p_1(x) + v_3P(x), \tag{20}$$

where $v_i(x)$ are the coefficients of the GPS; $P(x) = p_1(x)p_2(x)$ is the working range.

Thus, to detect an error, one can use the MRS coefficients. This allows reducing the hardware and time costs for the error search in PRNS as compared with (Chu, &Benaissa, 2013).

Obviously, if the condition $deg\, A(x) < \deg P(x)$ does not hold, then the leading coefficient of MRS will be $v_{k+1}(x) \neq 0$. This means that there is an error in the PRNS code.

Consider an example. In the PRNS system with the working bases $p_1(x)=x^4+x+1$, $p_2(x)=x^4+x^3+1$ and the control base $p_3(x)=x^4+x^3+x^2+x+1$, we calculate the orthogonal bases and represent them in GPS.

$$B_1(x)=x^8+x^4+x^4+x^2+1=[1,\ \ x^3+x+1,\ \ 1];$$
$$B_2(x)=x^{10}+x^8+x^5+x^4+x^4+x^2+1=[0,\ \ x^3+x+1,\ \ x^2+x];$$
$$B_2(x)=x^{10}+x^8+x^5+x^4+x^4+x^2+1=[0,\ \ 0,\ \ x^2+x+1].$$

Let us transfer into MRS the PRNS code $\{d4\}=x^7+x^6+x^4+x^2=(0, x^2+1, x^3+x^2+1)$·

Let us multiply the residues by the orthogonal bases, represented in MRS module by module, taking into account the excess of the module in the calculation of the subsequent coefficient. The results are shown in Table 5.

Since the leading coefficient $v_3 = 0$, the PRNS code contains no errors.

Suppose that an error with respect to the first base has occurred. Then the PRNS code $A^*=(1, x^2+1, x^3+x^2+1)$. Let us calculate the MRS coefficients. The results are shown in Table 6.

Since the leading coefficient $v_3 = 1$, the PRNS code contains an error.

However, the PC data of the PRNS codes cannot be used to eliminate the consequences of failures in the AES encryption. This is due to the fact that to correct a single error in the PRNS code, two redundant bases are used.The value $v_3 = 1$ canappear, ifanerror with respect to the control base took place. Inthiscase, the combination $A^* = (0, x^2+1, x^2)$. As a result of collision, it is impossible

Modules	$p_1(x)=x^4+x+1$	$p_2(x)=x^4+x^3+1$		$p_3(x)=x^4+x^3+x^2+x+1$
PRNS code	Product	Product	Excess of the polynomial $p_2(x)$	Result
$a_1(x)=0$	0	0	0	0
$a_2(x)=x^2+1$	0	x^3+x^2	$x+1$	$1+(x+1)=x$
$a_3(x)=x^3+x^2+1$	0	0	0	x
MRS coefficients	$v_1(x)=0$	$v_2(x)=x^3+x^2$	-	$v_3(x)=0$

Table 5. Calculation of the MRS coefficients

Modules	$p_1(x)=x^4+x+1$	$p_2(x)=x^4+x^3+1$		$p_3(x)=x^4+x^3+x^2+x+1$
PRNS code	Product	Product	Excess of the polynomial $p_2(x)$	Result
$a_1(x)=1$	1	x^3+x^2+1	0	1
$a_2(x)=x^2+1$	0	x^3+x^2	$x+1$	$1+(x+1)=x$
$a_3(x)=x^3+x^2+1$	0	0	0	x
MRS coefficients	$v_1(x)=0$	$v_2(x)=1$	-	$v_3(x)=1$

Table 6. Calculation of the MRS coefficients

to identify the distorted digit in the PRNS code.

Let us considerusingthe interval number as a PC. It is known that

$$l(x)=\left[\frac{A(x)}{P(x)}\right]=\left[\frac{\sum_{i=1}^{3}a_i(x)B_i(x)\bmod P'(x)}{P(x)}\right] \qquad (14)$$

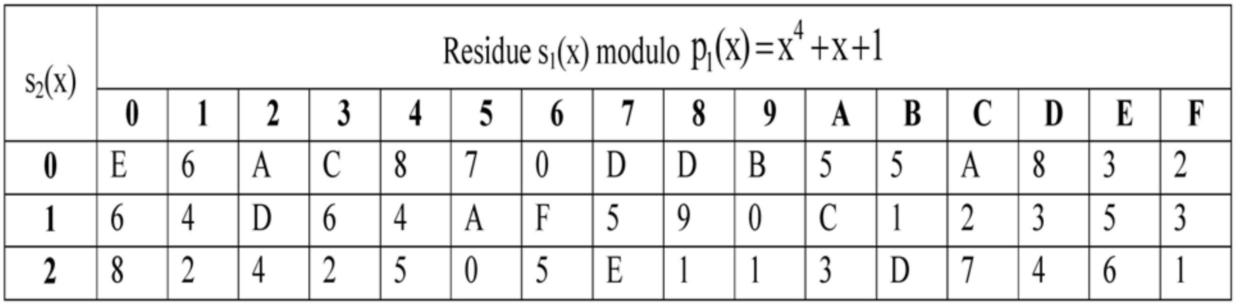
If $l(x)=0$, then the PRNS code does not contain an error, otherwise, it contains one. In Table 5, there are given the values of the interval number for the errors of the PRNS code.

The analysis of Table 5 shows that, in using one control base, there takes place coincidence of interval numbers for different errors.

Consequently, there arises a need to develop new principles of constructing the PRNS codes, aimed at correcting an error caused by a failure.

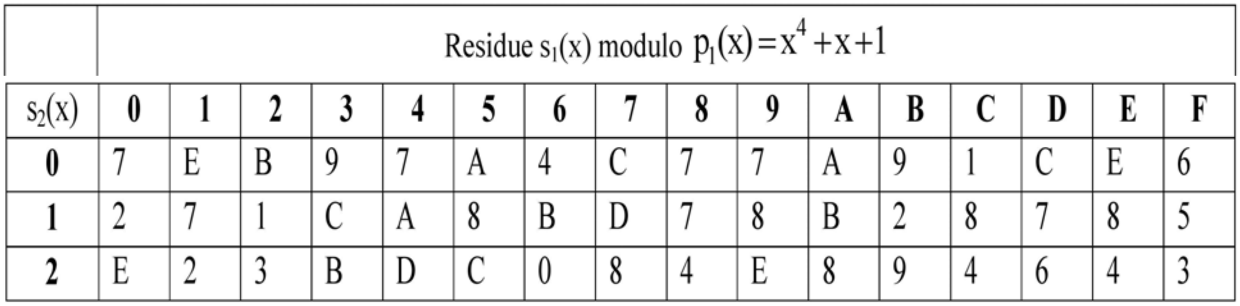
Base	Depth of the error	Interval number
$p_1(x)=x^4+x+1$	$\Delta a_1(x)=1$	1
	$\Delta a_1(x)=x$	$x+1$
	$\Delta a_1(x)=x^2$	x^2+x+1
	$\Delta a_1(x)=x^3$	x^3+x^2+x+1
$p_2(x)=x^4+x^3+1$	$\Delta a_2(x)=1$	x^2+x
	$\Delta a_2(x)=x$	x^3+x^2
	$\Delta a_2(x)=x^2$	x^2
	$\Delta a_2(x)=x^3$	x^3+1
$p_3(x)=x^4+x^3+x^2+x+1$	$\Delta a_3(x)=1$	x^2+x+1
	$\Delta a_3(x)=x$	x^3+x^2+x+1
	$\Delta a_3(x)=x^2$	$x+1$
	$\Delta a_3(x)=x^3$	x^2+x

Table 5. Interval numbersfor the errors of the PRNS code



$s_2(x)$	Residue $s_1(x)$ modulo $p_1(x)=x^4+x+1$															
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	E	6	A	C	8	7	0	D	D	B	5	5	A	8	3	2
1	6	4	D	6	4	A	F	5	9	0	C	1	2	3	5	3
2	8	2	4	2	5	0	5	E	1	1	3	D	7	4	6	1

Table 6. Residues $a_3(x)$ modulo $p_3(x)=x^4+x^3+x^2+x+1$



	Residue $s_1(x)$ modulo $p_1(x)=x^4+x+1$															
$s_2(x)$	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	7	E	B	9	7	A	4	C	7	7	A	9	1	C	E	6
1	2	7	1	C	A	8	B	D	7	8	B	2	8	7	8	5
2	E	2	3	B	D	C	0	8	4	E	8	9	4	6	4	3

Table 7. Residues $a_4(x)$ modulo $p_3(x)=x^4+x^3+x^2+x+1$

2.3 Development of an algorithm of error correction by a PRSN code with one control base

To correct an error, we introduce into the code $A(x)=(a_1(x), a_2(x),..., a_k(x))$ a control base $p_{k+1}(x)$, which satisfies

$$\deg p_{k+1}(x) \geq \deg p_k(x). \qquad (13)$$

To correct a single error, a corrupted digit in the code,we compute

$$a_{k+1}(x)=\sum_{i+1}^{k} a_i(x), \qquad (14)$$

$$a_{k+2}(x)=\sum_{i+1}^{k}(i(x)a_i(x)) \operatorname{mod} p_{k+1}(x), \qquad (15)$$

where $i(x)$ is the polynomial form of the i-thsequence number, Σ is the summation modulo two.

In this case, the PRNS code has the form $A(x)=(a_1(x), a_2(x),..., a_k(x), a_{k+1}(x), a_{k+2}(x))$.To correct the error, we calculate

$$a^*_{k+1}(x)=\sum_{i+1}^{k} a_i(x), \qquad (16)$$

$$a^*_{k+2}(x)=\sum_{i+1}^{k}(i(x)a_i(x)) \operatorname{mod} p_{k+1}(x). \qquad (17)$$

Then the error syndrome is computed

$$\delta_1(x)=a_{k+1}(x)+a^*_{k+1}(x), \qquad (18)$$

$$\delta_2(x)=a_{k+2}(x)+a^*_{k+2}(x). \qquad (19)$$

If $\delta_1(x)=0$ and $\delta_2(x)=0$, then the PRNS code does not contain an error. Otherwise, the PRNS code contains an error. On the basis of the values of $\delta_1(x)$ and $\delta_2(x)$, the error correction in the code is performed.

Results

2.1 Realization of the Sub Bytes transformation in the PRNScode

To eliminate the consequences of failures in the AES cipher, we introduce the Tables 3 and 4. Table 6 contains the data on the sum of residues of the working bases $p_1(x)=x^4+x+1$ and $p_2(x)=x^4+x^3+1$, obtained on the basis of (14).

In Table 7 the data are presented on the weighted sum of the working bases $p_1(x)=x^4+x+1$ and $p_2(x)=x^4+x^3+1$, obtained on the basis of (15).

The usage of Tables 6 and 7 allows carrying out verification of the SubBytes performance. This takes place in the error correction block.

Analogous solutions can be presented also for other transformations of the AES encryption: Shift Rows, Mix Columns and Add Round Key.

Let $S = \{00011001_2\} = \{19_{16}\}$. The residues $S(x) = (A, 0)$ come to the inputs of the substitution table of the S_1-block modulo $p_1(x) = x^4 + x + 1$ and the S_2-block modulo $p_2(x) = x^4 + x^3 + 1$. As a result of action of the state byte $S = \{00011001_2\} = \{19_{16}\} = (A, 0)$, a substitution byte is obtained, equal to $S'(x) = (0, 5)$ in PRNS. This corresponds to the substitution $S'(x) = \{d4_{16}\} = \{11010100_2\}$.

When at the input of Table 6 there are delivered the values $S(x) = (A, 0)$ of the residuesof the current state, then the value $\{5_{16}\} = \{0101_2\} = \{x^2 + 1\}$ will be read off from the output of the substitution table.This number is situated at the intersection of the column **A** and the row **0** in Table 6. When at the input of Table7 there are delivered the values $S(x) = (A, 0)$ of the residues of the current state, then the value $\{A_{16}\} = \{1010_2\} = \{x^3 + x\}$ will be read off from the output of the substitution table. This number is situated at the intersection of the column **A** and the row **0** in Table 7.

Suppose that in the process of operation of the AES cipher there were no failures. Then, after completion of the substitution operation using two substitution tables of the S_1-block and S_2-block, the checking for errors is carried out in the PRNScombination. In this case, the expressions (16) and (17) are used. Then we have

$$s_3^*(x) = \sum_{i+1}^{2} s_i'(x) = 0 + x^2 + 1 = x^2 + 1 = \{5_{16}\}.$$

$$s_4^*(x) = \sum_{i+1}^{2} (i(x)s_i'(x)) \bmod m_3(x) = (0 + x(x+1)) \bmod x^4 + x^3 + x^2 + x + 1 =$$

$$= x^3 + x = \{A_{16}\}$$

According to(18) and (19), the calculation of the error syndrome is performed. If the syndrome equals to zero, then no failure has taken place. The values of the residues $s_1'(x) = 0000_2$ and $s_2'(x) = 1010_2$ with respect to working bases of the PRNS code will participate in the subsequent round transformations.

Suppose that a failure occurred, which has caused a change of the residue value with respect to $p_1(x) = x^4 + x + 1$, and suppose that its depth $\Delta s_1(x) = 1$. Then

$$s_1^{err}(x) = s_1'(x) + \Delta s(x) = 0 + 1 = 0001_2$$

Then the following code will be delivered to the input of the error correction block

$$S'(x) = (s_1^{err}(x), s_2'(x), s_3'(x), s_4'(x)) = (0001_2, 0101_2, 0101_2, 1010_2) = \\ = (1, x^2 + 1, x^2 + 1, x^3 + x) .$$

Let us calculate $s_3^*(x)$ and $s_4^*(x)$ in accordance with (16) and (17)

$$s_3^*(x) = \sum_{i+1}^{2} s_i'(x) = 1 + x^2 + 1 = x^2 = \{4_{16}\}. .$$

$$s_4^*(x) = \sum_{i+1}^{2} (i(x)s_i'(x)) \bmod m_3(x) = (1 + x(x+1)) \bmod x^4 + x^3 + x^2 + x + 1 =$$

$$= x^3 + x + 1 = \{B_{16}\}$$

Let us compute $\delta_1(x)$ and $\delta_2(x)$. We use $s_3'(x)$ and $s_4'(x)$ from Tables 3 and 4. We obtain

$$\delta_1(x) = s_3'(x) + s_3^*(x) = (x^2 + 1) + (x^2) = 1.$$

$$\delta_2(x) = s_4'(x) + s_4^*(x) = (x^3 + x) + (x^3 + x + 1) = 1.$$

Since $\delta_1(x) \neq 0$ and $\delta_2(x) \neq 0$, then a failure occurred in the execution of Sub Bytes. Using the values of $\delta_1(x)$ and $\delta_2(x)$,we determine that the error $\Delta s_1(x) = 1$. Then we carry out a correction

$$s_1'(x) = s_1^{err}(x) + \Delta s(x) = 0001 + 0011 = 0000_2 .$$

Discussion

To simulate the developed algorithm of error correction,we used the Altera processor from the Cyclone III family (the EP3C80F484C6 model). In the experiment, the description was realized in the Verilog HDL language. To compare the simulation results, we used the modules, which are built in the Quartus II development environment and which define the number of logic elements. To simulate the 32-Bit Non-redundant AES, we needed 160 LUT used for RAM. Therefore, in the implementation of the “2 out of 3” backup system, the LUT number will increase to 480. Since the developed algorithm uses two control residues, the hardware implementation of the 32-Bit redundant PRNS AES, excluding the cost of direct and inverse code conversion and error correction, will be 320 LUT used for RAM. For the direct conversion of the 8-bit code into the 4-bit residues and back, we needed additional 11 LUT used for RAM. To perform error correction, it is necessary to carry out the following procedures:

- To calculate the control residues $a_3(x)$, $a_4(x)$;

- To calculate the error syndromes and $\delta_3(x)$, $\delta_4(x)$;

- To select the correcting values;

- To perform correction of the PRNS code.

These operations must be carried out after each stage of transformations in the AES algorithm. The simulation results showed that the correction of the PRNS code required 60 LUT used for RAM. Thus, the hardware costs of implementing 32-Bit redundant PRNS AES amounted to 391 LUT used for RAM. The conducted studies have shown that due to the developed algorithm of error correction, 1.22 times less hardware costs are required in comparison with the classical redundancy system “2 *out o*f 3”. The obtained results testify to the effectiveness of the use of non-positional modular codes to counteract the consequences of the failure-based attacks.

Conclusion

The paper analyzed the main methods of eliminating the consequences of failures in the AES encryption. The studies presented in the paper demonstrate that the use of the PRNS code enables performing AES encryption in the finite fields $GF(2^4)$ of lower order with smaller time and hardware costs. Besides,a PRNS code with one redundant base is able to detect errors caused by the failures in the work of the encoder. In order to solve the problem of error correction, an algorithm of error correction in the PRNS code having minimal redundancyis developed. The usage of the developed algorithm ensures correcting all single errors and up to 80 percent of the double errors that occur due to failures.

In this case, the considered algorithm corrects errors with lower hardware costs in comparison with the method of masking failures “2 out of 3”. By virtue of such triple redundancy, all the single errors will be corrected. However, the use of a corrective PRNS code allows reducing the introduced redundancy. The conducted studies have shown that due to the developed algorithm of error correction, 1.22 times less hardware costs are required in comparison with the redundancy system “2 *out of* 3”. The obtained results testify to the effectiveness of the use of non-positional modular codes to counteract the consequences of the failure-based attacks.

The prospects of further research are connected with the development of new algorithms allowing correcting the errors of higher multiplicity. The algorithm presented in the article is capable of correcting 100% of single and up to 80% of double errors. The impossibility of correcting all double errors is due to the fact that the PC values of the PRNS code with one control base for different sets of erroneous bits will be the same. The development of an algorithm that can resolve this collision will allow correcting 100% of double errors without increasing the code redundancy.

Also promising is the task of investigating the possibility of application of the developed principles of encoding the redundant PRNS codes for the linear and nonlinear cryptographic transformations in the “*Kuznechik*” encryption algorithm, which is an AES-like cipher.

References

[1] Satoh, A., Morioka, S.,Takano, K., Munetoh, S. (2001). A Compact Rijndael Hardware Architecture with S-BoxOptimization. *In*: Advances in Cryptology – ASIACRYPT, 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9-13, Proceedings (2248), p. 239-254.

[2] Bar-El, H., Choukri, H., Naccache, D., Tunstall, M., Whelan, C. (2004).The Sorcerer’s Apprentice Guide to Fault Attacks.Retrieved April 9, 2016 from http://citeseer.ist.psu.edu.

[3] Berezhnoy, V.V., Chervyakov, N.I., Shchelkunova,Yu.O., Shilov, A. A. (2004). Neural Network Realization in the Polynomial Residue Number Systemof the Digital Signal ProcessingOperationswith Increased Number of Digits. Neurocomputers: Development, Application, 5-6, 94-98.

[4] Blomer, J., Seifert, J.-P. (2003). Fault Based Cryptanalysis of the Advanced Encryption Standard (AES). Retrieved April 9, 2016 from http://wwwcs.uni-paderborn.de.

[5] Chu, J., Benaissa, M. (2009). Polynomial Residue Number System $GF(2^m)$ Multiplier Using Trinomials. *In*: 17th European Signal Processing Conference, August 24-28,Glasgow, Scotland, p. 958-962.

[6] Chu, J., Benaissa, M. (2013). Error Detecting AES Using Polynomial Residue Number System. *Microprocessors and Microsystems,* 37, 228-234.

[7] Dunin, A., Rassvetaev, V., Loboykin, V. (2014).Neural network implementation of an error correction algorithm in the modular code on the basis of coefficients of a generalized polyadic system. *Modern high technologies*,12-1,145-151.

[8] Duquesne, S.(2011). RNS Arithmetic in F_p^k and Application to Fast Pairing Computation. *Journal of Mathematical Cryptology,* 5 (1) 51-88.

[9] Gapochkin, A., Zavorotinsky, D, Solodkin,I. (2014).Improvement of the algorithm for computing the interval number used for error correction in a modular code. *Modern high technologies*, 11,12-15.

[10] Gordenko, D. V., Rezenkov, D.N., Sarkisov, A.B. (2014). Methods and Algorithms of Reconfiguration of Non-Positional Computational Structures for Providing the Failure Robustness of the Special Processors. Stavropol:Fabula Publishers.

[11] Hodjat, A.,Verbauwhede, I. (2004).Minimum Area Cost for a 30 to 70Gbits/s AESProcessor. *In:* Proceedings of IEEE Computer Society Annual Symposium on VLSI, 83-88.

[12] Zhengbing, Hu., Yatskiv, V., Sachenko, A. (2015). Increasing the data transmission robustness in WSN using the modified error correction codes on a residue number system.*Electronics and electrical technology,* 1 (21) 76-81. http://dx.doi.org/10.5755/j01.eee.21.1.6657.

[13] Park, J. H., Moon, S. J., Choi, D. H., Kang, Y. S., Ha, J. C.(2011). Differential Fault Analysis for Round-Reduced AES by Fault Injection. *ETRI Journal,* 33 (3) 434-442.

[14] Kalmykov, I.A., Katkov, K. A., Naumenko, D.O., Sarkisov, A.B., Makarova, A.V.(2014). Parallel Modular Technologies in Digital Signal Processing. *Life Science Journal,* 11(11s) 435-438. Retrieved April 9, 2016 from http://www.lifesciencesite.com.

[15] Kalmykov, I. A., Katkov, K. A., Timoshenko, L.I., Dunin, A.V., Gish, T.A.(2015). Application of Modular Technologies in the Large-Scale Analysis of Signals.*Journal of Theoretical and Applied Information Technology*, 80 (3) 391-400.Retrieved April 9, 2016 from http://www.jatit.org/volumes/Vol80No3/2Vol80No3.pdf.

[16] Katkov, K. A.,Kalmykov, I.A. (2013). Application of Parallel Technologies in Navigation Management under the Conditions of Artificial Ionospheric Disturbances. *World Applied Sciences Journal,* 26 (1) 108-113.

[17] Mangard, S., Aigner, M.,Dominikus, S. (2003).A Highly Regular and Scalable AES Hardware Architecture. *IEEE Transactionson Computers,* 52 (4) 483-491.

[18] Natale, G. D., Flottes, M. L., Rouzeyre, B. (2007).On-Line Self-Test of AES Hardware Implementations. *In:* DSN 2007Workshopon Dependable and Secure Nanocomputing in Conjunction with the 37*th* Annual IEEE/IFIP International Conference on Dependable Systems and Networks, June 28, 2007. Edinburgh International Conference Centre.

[19] Nechvatal, J., Barker, E., Bassham, L., Burr, W., Dworkin, M., Foti, J.,Roback, E. (2000). Report on the Development of the Advanced Encryption Standard (AES).Retrieved April 9, 2016 from http://csrc.nist.gov

[20] Peacham, D., Thomas, B.A. (2006).DFA Attack against the AES Key Schedule. Retrieved April 9, 2016 from http://www.siventure.com.

[21] Schneier, B., Kelsey, J., Whiting, D., Wagner, D., Hall, C., Ferguson, N., Kohno, T., Stay, M. (2000).The Twofish Team’s Final Comments on AES Selection. Retrieved April 9, 2016 from http://csrc.nist.gov.

[22] Park, S.J. (2003). Analysis of AES Hardware Implementations. ECE 679 Advanced Security and Cryptography. Oregon State University,USA. Retrieved April 9, 2016 from http://islab.oregonstate.edu/koc/ece679/project/ 2003/park.pdf.

[23] Yatskiv, V., Yatskiv, N., Su, Jun, Sachenko, A., Zhengbing, Hu (2013). The use of a modified correction code based on a reside number system in WSN, *In*: Proceedings 7*th* IEEE Int. Conf. Intelligent Data Acquisition and Advanced Computing Systems, (IDAACS 2013), 1, p. 513-516. Berlin, Germany.