Quasigroups and Related Systems **23** (2015), 271 − 282

# A new block ciphers based on wavelet decomposition of splines

*Alla B. Levina*

**Abstract.** This paper presents the idea of using wavelet decomposition of splines in cryptography. The cryptoalgorithms based on wavelet decomposition of splines uses just the algebraic calculation. With the help of algebraic formulas we can code and decode information that algorithms do not have the XOR using the round key, also it does not use S-boxes.

## 1. Introduction

The proposed paper discusses a new class of algorithms obtained using a new theory of the spline-wavelet decompositions on nonuniform sets. The theory of wavelet decomposition of splines has been used before to process discreet signals but never in cryptography.

Our proposal is to create cryptoalgoritms which will use only mathematical calculation that can process data blocks up to 2048 bits and more quickly. This research were carried out for splines of the first, second and third degree. Algorithms based on splines of an upper degree works slower but they remain stronger against different cryptoattacks.

The theory of wavelet-decomposition of splines can apply to different areas of cryptography. In this work we will illustrate just one way of use - creation of block ciphers, which will be presented on splines of the third degree.

The presented algorithms do not have the XOR operation with the round key and they do not use S-boxes as block ciphers GOST 28147-89 [10], 3DES [9], AES [3,8] and others. At the present time, only algorithm Threefish [11] is not using S-boxes and can process data blocks up to 1024 bits. Diffusion over multiple rounds we get by mathematical functions.

As a minus of the algorithms we can mention that not all the bytes are receiving enciphering on each round; some of them are just getting moved to several positions, unlike in the Feistel Structure. Present research explores algorithms which will cipher each byte on each round.

As a plus for the presented algorithms, we can mention that it is based only on the mathematical calculation, however it does help in the analyzing of algorithms.

2010 Mathematics Subject Classification: 11T71, 94A60, 68P25
Keywords: Block ciphers, splines, grids, formulas of decomposition and reconstruction.

The structure of the presented algorithms is absolutely new - and has the possibility for modernization of the process. The process of enciphering is based only on mathematical formulas, the formulas of decomposition from wavelet theory, process of deciphering is based on the formulas of reconstruction.

In this paper the basic concepts of algorithm, mathematical basics of process of enciphering/deciphering, illustration of spline-wavelet decomposition, and a demonstration of the work of algorithm is presented.

## 2. Idea of wavelet decomposition of splines

In this section we will briefly provide a concept of wavelet decomposition of splines [4, 5, 6]. We will illustrate spline-wavelet decomposition on the splines of the third degree. For splines of the first and second degree, the same theory is used.

On the set $X$ we build splines. Set $X$ consists of the elements $\{x_i\}_{i=0,\dots,L-1}$, where $\{x_i\}_{i=0,\dots,L-1}$ natural numbers. $L$ is the number of elements in the set $X$.

Splines of the third degree built on the set $X$ are presented in the formulas below:

$$\sum_{j=k-3}^{k} \omega_j(t) = 1, \quad t \in [x_k, x_{k+1})$$

$$\sum_{j=k-3}^{k} \frac{1}{3}\left(x_{j+1} + x_{j+2} + x_{j+3}\right)\omega_j(t) = t, \quad t \in [x_{k+1}, x_{k+2})$$

$$\sum_{j=k-3}^{k} \frac{1}{3}\left(x_{j+1}x_{j+2} + x_{j+1}x_{j+3} + x_{j+2}x_{j+3}\right)\omega_j(t) = t^2, \quad t \in [x_{k+2}, x_{k+3})$$

$$\sum_{j=k-3}^{k} x_{j+1}x_{j+2}x_{j+3}\,\omega_j(t) = t^3, \quad t \in [x_{k+3}, x_{k+4}).$$

With splines defined as $\omega_j(t)$ and $x_j$ elements of our set $X$.

For wavelet decomposition of splines, we take out one element $x_k$ from our set $X$ and we will obtain a new set $\overline{X}$. Elements of this set can be presented with the help of elements from the old set, as presented below:

$$\overline{x}_j = x_j \text{ if } j \leqslant k-1, \text{ and } \overline{x}_j = x_{j+1} \text{ if } j \geqslant k, \quad \xi = x_k.$$

On the new set $\overline{X}$, we can build new splines $\overline{\omega}_j$ *but these new splines can be present as a combination of splines which were built before on the set $X$. Also splines $\omega_j(t)$ can be obtained with the help of the new splines $\overline{\omega}_j(t)$ — it helps us to restore the information.*

This idea gives us two types of formulas: 1) formulas of decomposition 2) formulas of reconstruction. Step by step we take out elements from our primary set $X$ and build splines which use the new set (in this realization each time we take out just one element we get a new set and new splines, in another realization it can be taken a few elements each time).

The mathematical process of getting these formulas will not be introduced in this paper, instead we will present only a final version to show, how these formulas will look.

We have an information stream $c_i$ and we want to get new stream $\overline{c_i}$ based on the set $\overline{X}$. Formulas of decomposition will reform information stream $c_i$ in to the information stream $\overline{c_i}$ and wavelet part — element $b$. Formulas of reconstruction will restore stream $c_i$, using stream $\overline{c_i}$ and element $b$.

*Formulas of decomposition:*

$$\bar{c}_i = c_i \quad \text{if} \quad 0 \leqslant i \leqslant k-5, \quad \bar{c}_i = c_{i+1} \quad \text{if} \quad k-1 \leqslant i \leqslant M-2,$$

$$\bar{c}_{k-3} = \frac{\xi - \bar{x}_k}{\xi - \bar{x}_{k-3}} \cdot c_{k-4} + \frac{\bar{x}_k - \bar{x}_{k-3}}{\xi - \bar{x}_{k-3}} \cdot c_{k-3},$$

$$\bar{c}_{k-2} = (\xi - \bar{x}_k)(\xi - \bar{x}_{k+1}) \cdot c_{k-4} + (\xi - \bar{x}_{k+1})(\bar{x}_k - \bar{x}_{k-3}) \cdot c_{k-3} +$$

$$+ (\bar{x}_{k+1} - \bar{x}_{k-2})(\xi - \bar{x}_{k-3}) \cdot c_{k-2}) \cdot [\xi - \bar{x}_{k-2}]^{-1} [\xi - \bar{x}_{k-3}]^{-1},$$

$$b = c_{k-1} - \frac{\bar{x}_{k+2} - \xi}{\bar{x}_{k+2} - \bar{x}_{k-1}} \cdot \bar{c}_{k-2} - \frac{\xi - \bar{x}_{k-1}}{\bar{x}_{k+2} - \bar{x}_{k-1}} \cdot \bar{c}_{k-1}.$$

*Formulas of reconstruction:*

$$c_i = \bar{c}_i \quad \text{if} \quad 0 \leqslant i \leqslant k-4, \quad c_i = \bar{c}_i \quad \text{if} \quad k \leqslant i \leqslant M-1,$$

$$c_{k-3} = \frac{\bar{x}_k - \xi}{\bar{x}_k - \bar{x}_{k-3}} \cdot \bar{c}_{k-4} + \frac{\xi - \bar{x}_{k-3}}{\bar{x}_k - \bar{x}_{k-3}} \cdot \bar{c}_{k-3},$$

$$c_{k-2} = \frac{\bar{x}_{k+1} - \xi}{\bar{x}_{k+1} - \bar{x}_{k-2}} \cdot \bar{c}_{k-3} + \frac{\xi - \bar{x}_{k-2}}{\bar{x}_{k+1} - \bar{x}_{k-2}} \cdot \bar{c}_{k-2},$$

$$c_{k-2} = \frac{\bar{x}_{k+2} - \xi}{\bar{x}_{k+2} - \bar{x}_{k-1}} \cdot \bar{c}_{k-2} + \frac{\xi - \bar{x}_{k-1}}{\bar{x}_{k+2} - \bar{x}_{k-1}} \cdot \bar{c}_{k-1} + b.$$

We will now present how we can use this idea in construction of block ciphers, for it we will illustrate these formulas in a more readable way.

## 3. Specification

The presented algorithm is an iterated block cipher with a variable block length and it is relative to the class of block cipher algorithms.

## 3.1. Basic concepts of the algorithm

A process of enciphering and deciphering consists of $K$ identical rounds.

This algorithm can work with the block length up to 2048 bits and more. The number of rounds is denoted by $K$, $\mathbf{K}_{X\gamma}$ is a key length, $M$ is a block length (In the table below $M$ and $\mathbf{K}_{X\gamma}$ are bytes).

Let $\mathbb{K} = (X, \gamma)$ be *a key*; here $X$ is an ordered set, $X = \{x_j\}_{j=0,\dots,L-1}$, where $L$ is a number of elements in the set $X$ and $\gamma$ is the order of ejection of elements from the set. The key consist from two sets.

We have determined the number of rounds by looking at the maximum number of rounds for which attacks have been found and has added a considerable level of security and provides a higher margin of safety, in some cases there can be less rounds held and the key will be smaller. Key length is equal to (number of rounds + 3)+(number of rounds) bytes.

Number of rounds and key length as a function of the block length is given in Table 1.

| | $K$ | $\mathbf{K}_{X\gamma}$ |
|---|---|---|
| $M = 8$ bytes | 6 | 15 |
| $M = 16$ bytes | 14 | 31 |
| $M = 24$ bytes | 22 | 47 |
| $M = 32$ bytes | 30 | 63 |
| $M = 64$ bytes | 62 | 127 |
| $M = 128$ bytes | 126 | 255 |
| $M = 256$ bytes | 254 | 511 |

Table 1.

Number of elements in the set $X$ as a function of the block length are presented in the Table 2.

| | L |
|---|---|
| $M = 8$ bytes | 9 |
| $M = 16$ bytes | 17 |
| $M = 24$ bytes | 25 |
| $M = 32$ bytes | 33 |
| $M = 64$ bytes | 65 |
| $M = 128$ bytes | 129 |
| $M = 256$ bytes | 257 |

Table 2.

The process of creating round key will be explained in section 3.2 more detailed.

A *sequence* $C = \{c_i\}_{i=0,\dots,M-1}$ is a *plaintext;* $|C| = M$ is a quantity of elements which are ciphered, $C$ is the ordered set.

Elements $\{c_i\}_{i=0,\dots,M-1}$ and $\{x_j\}_{j=0,\dots,L-1}$ are bytes (we are working with one-byte words, but we can also work with 4-bytes words).

Let us suppose that the set $X$ and $C$ can be periodic with the period $T$ so $x_j = x_{j+T}$ and $c_i = c_{i+T}, \quad \forall j \in \mathbf{Z}$.

The process ofenciphering bases on the formulas of decomposition from wavelet theory, after $K$ rounds we obtain the ciphertext. For deciphering we will use formulas of reconstruction.

The process of enciphering and deciphering consists of two steps: 1) creation of round key and 2) round transformation.

## 3.2. The round key creation

Round key transformation consist of two steps. We will now check the first round – all rounds are the same.

1. We eject element $x_{\gamma_1}$ from the primary set $X$. The received set is defined as $X_{-1}$ and $X_{-1} = \{x_{-1,j}\}^1$, elements of new set are equal:

$$x_{-1,j} = x_j \quad \text{if} \quad j < \gamma_1, \tag{1}$$

$$x_{-1,j} = x_{j+1} \quad \text{if} \quad j > \gamma_1. \tag{2}$$

The element $x_{\gamma_1}$ which has been taken out of the set $X$ is defined as $\xi$, $\xi = x_{\gamma_1}$.

In the next round we will be working with the set $X_{-1}$ and $x_{\gamma_2}$.

**Example:**

We will now calculate the set $X_{-1}$ from the set $X$. For example: $X$ consists of 6 bytes $(\{1, 3, 5, 9, 10, 6\}$ and the number of the element which will eject is $\gamma_i = 4$. All numerations starts from 0, it means that our ejected element $\xi = x_{\gamma_i} = 10$ and the new set $X_{-1} = (\{1, 3, 5, 9, 6\}$.

2. We enter the following designations - elements which we use in the process of enciphering we mark $...^{en}$, for deciphering $...^{de}$:

$$A^{en}_{-1} = \xi - x_{-1,\gamma_1}, \quad B^{en}_{-1} = \xi - x_{-1,\gamma_1-1}, \quad C^{en}_{-1} = \xi - x_{-1,\gamma_1-2},$$

$$D^{en}_{-1} = \xi - x_{-1,\gamma_1-3}, \quad E^{en}_{-1} = \xi - x_{-1,\gamma_1+1}, \quad F^{en}_{-1} = x_{-1,\gamma_1+2} - x_{-1,\gamma_1-1}.$$

$$A^{de}_{-1} = x_{-1,\gamma_1} - x_{-1,\gamma_1-3}, \quad B^{de}_{-1} = x_{-1,\gamma_1} - \xi \quad C^{de}_{-1} = x_{-1,\gamma_1+1} - x_{-1,\gamma_1-2},$$

$$D^{de}_{-1} = x_{-1,\gamma_1+1} - \xi, \quad E^{de}_{-1} = x_{-1,\gamma_1+2} - x_{-1,\gamma_1-1}, \quad F^{de}_{-1} = x_{-1,\gamma_1+2} - \xi.$$

These designations will help us in the future realization of the formulas. As we can see from calculating these elements we are using our new set; $X_{-1}$ and element $\xi$ from our set $X$.

1. To avoid misunderstanding with numeration in this work if it's written $\{...\}_{-i,j}$ $-i$ is a number of the round and $j$ is a number of the element, if it's just $\{...\}_{-i}$ $-i$ is a number of round.

With the help of these designations we can calculate elements of the round key:

$$I^{en} = \frac{A^{en}_{-1}}{D^{en}_{-1}}(\mathrm{mod} N), \quad II^{en} = \frac{E^{en}_{-1}}{C^{en}_{-1}}(\mathrm{mod} N), \quad III^{en} = \frac{B^{en}_{-1}}{F^{en}_{-1}}(\mathrm{mod} N).$$

$$I^{de} = \frac{B^{de}_{-1}}{A^{de}_{-1}}(\mathrm{mod} N), \quad II^{de} = \frac{D^{de}_{-1}}{C^{de}_{-1}}(\mathrm{mod} N), \quad III^{de} = \frac{F^{de}_{-1}}{E^{de}_{-1}}(\mathrm{mod} N).$$

We use mod N, where N is a prime number and it gives us possibility to get elements that will take one byte. All future calculations will be made by mod, we will use the same mod as in algorithm Rijndael $x^8+x^4+x^3+x+1$.

**Example:**

Now we will illustrate how we calculate elements $I^{en}$, $II^{en}$, $III^{en}$, $I^{de}$, $II^{de}$, $III^{de}$. For it we will use set $X_{-1} = (\{1,3,5,9,6\}$, $\xi = 10$, $\gamma_1 = 4$.

$$A^{en}_{-1} = \xi - x_{-1,\gamma_1} = 10 - 6 = 4, \quad B^{en}_{-1} = \xi - x_{-1,\gamma_1-1} = 10 - 9 = 1,$$

$$C^{en}_{-1} = \xi - x_{-1,\gamma_1-2} = 10 - 5 = 5, \quad D^{en}_{-1} = \xi - x_{-1,\gamma_1-3} = 10 - 3 = 7,$$

$$E^{en}_{-1} = \xi - x_{-1,\gamma_1+1} = 10-1 = 9, \quad F^{en}_{-1} = x_{-1,\gamma_1+2} - x_{-1,\gamma_1-1} = 3-9 = -6.$$

$$A^{de}_{-1} = x_{-1,\gamma_1} - x_{-1,\gamma_1-3} = 6 - 3 = 3, \quad B^{de}_{-1} = x_{-1,\gamma_1} - \xi = 6 - 10 = -4,$$

$$C^{de}_{-1} = x_{-1,\gamma_1+1} - x_{-1,\gamma_1-2} = 1-5 = -4, \quad D^{de}_{-1} = x_{-1,\gamma_1+1} - \xi = 1-10 = -9,$$

$$E^{de}_{-1} = x_{-1,\gamma_1+2} - x_{-1,\gamma_1-1} = 3-9 = -6, \quad F^{de}_{-1} = x_{-1,\gamma_1+2} - \xi = 3-10 = -7.$$

Instead of mod N, we will use mod 11, we need a prime number, for this example it will be easer to use 11.

$$I^{en} = \frac{A^{en}_{-1}}{D^{en}_{-1}}(\mathrm{mod} 11) = \frac{4}{7}(\mathrm{mod} 11) = 4 \cdot 8(\mathrm{mod} 11) = 10,$$

$$II^{en} = \frac{E^{en}_{-1}}{C^{en}_{-1}}(\mathrm{mod} 11) = \frac{9}{5}(\mathrm{mod} 11) = 9 \cdot 9(\mathrm{mod} 11) = 4,$$

$$III^{en} = \frac{B^{en}_{-1}}{F^{en}_{-1}}(\mathrm{mod} 11) = \frac{1}{-6}(\mathrm{mod} 11) = -2(\mathrm{mod} 11) = 9.$$

$$I^{de} = \frac{B^{de}_{-1}}{A^{de}_{-1}}(\mathrm{mod} 11) = \frac{-4}{3}(\mathrm{mod} 11) = 6,$$

$$II^{de} = \frac{D^{de}_{-1}}{C^{de}_{-1}}(\mathrm{mod} 11) = \frac{-9}{-4}(\mathrm{mod} 11) = 5,$$

$$III^{de} = \frac{F^{de}_{-1}}{E^{de}_{-1}}(\mathrm{mod} 11) = \frac{-7}{-6}(\mathrm{mod} 11) = 3.$$

All the calculations by mod goes by the rules of calculation in finite fields.

On each round, key transformation goes as it was presented.

## 3.3. Process of enciphering

The process of enciphering also consists of two steps. For the encoding of information we will use formulas of decomposition for the splines of the third degree.

On the first round our plaintext is $\{c_i\}_{i=0,\dots,M-1}$.

*First round:*

1. With the help of round key, we will present formulas of decomposition for splines of the third degree, and we will code our plain text.

$$c_{-1,j} = c_j \quad \text{if} \quad 0 \leqslant j \leqslant \gamma_1 - 5, \tag{3}$$

$$c_{-1,j} = c_{j+1} \quad \text{if} \quad \gamma_1 - 1 \leqslant j \leqslant M - 2, \tag{4}$$

$$c_{-1,\gamma_1-3} = \big(I^{en} \cdot (c_{\gamma_1-4} - c_{\gamma_1-3}) + c_{\gamma_1-3}\big)(\mathrm{mod} N), \tag{5}$$

$$c_{-1,\gamma_1-2} = \big(I^{en} \cdot II^{en}(c_{\gamma_1-4} - c_{\gamma_1-3}) + II^{en} \cdot (c_{\gamma_1-3} - c_{\gamma_1-2}) + c_{\gamma_1-2}\big)(\mathrm{mod} N), \tag{6}$$

$$b_{-1} = \big(c_{\gamma_1-1} - c_{-1,\gamma_1-2} + III^{en} \cdot (c_{-1,\gamma_1-2} - c_{-1,\gamma_1-1})\big)(\mathrm{mod} N). \tag{7}$$

As we can see from the formulas (3) – (7) on the first round formulas of decomposition, we are taking out element $c_{\gamma_1-1}$ from our plain text, elements $c_{-1,\gamma_1-2}$ and $c_{-1,\gamma_1-3}$ is getting transformed, with the help of formulas (5) – (6) while other elements of our plain text starts from the element $\gamma_1 - 1$ that we are moving. Element $b_{-1}$ is the element of a wavelet stream, which will help us to restore the initial information.

From

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $c_0$ | $c_1$ | $c_2$ | $c_3$ | $c_4$ | $c_5$ | $c_6$ | $c_7$ |

we obtain

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $c_0$ | $c_1$ | $I^{en} \cdot (c_1 - c_2) + c_2$ | $I^{en} \cdot II^{en}(c_1 - c_2) + II^{en} \cdot (c_2 - c_3) + c_3$ | $c_5$ | $c_6$ | $c_7$ | $b_{-1}$ |

Figure 1. Illustration of the process of enciphering for 8 bytes on the first round if $\gamma_1 = 5$.

We use mod N as it will insure that we will still stay in the byte in spite of multiplying.

2. At the end we make a shift of sequence $c_{-1,j}$ as follows:

$$c_{-1,0} \to c_{-1,1} \to c_{-1,2} \dots \to c_{-1,M-1} \to c_{-1,0}.$$

It gives us more transformations.

From

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $c_0$ | $c_1$ | $I^{en} \cdot (c_1 - c_2) + c_2$ | $I^{en} \cdot II^{en}(c_1 - c_2) + II^{en} \cdot (c_2 - c_3) + c_3$ | $c_5$ | $c_6$ | $c_7$ | $b_{-1}$ |

we obtain

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $c_7$ | $c_0$ | $c_1$ | $I^{en}\cdot(c_1-c_2)+c_2$ | $I^{en}\cdot II^{en}(c_1-c_2)+II^{en}\cdot(c_2-c_3)+c_3$ | $c_5$ | $c_6$ | $b_{-1}$ |

Figure 2. Illustration of the shift.

On the next round we will be working with the set $\{c_{-1,j}\}_{j=0,\dots,M-2}$.

All rounds except the final round go the same, on the last round we do not do the shift.

As a result, after $K$ rounds we get two sequences

$$\{b_{-n}\}_{n=1,2,\dots,K}, \quad \{c_{-K,j}\}_{j=0,1,2,\dots,M-K-1}.$$

*Sequence* $\{c_{-K,j}, b_{-n}\}_{n=1,2,\dots,K;j=0,1,2,\dots,M-K-1}$ *is the ciphertext.*

**Example:**

We continue our previous example $I^{en}=10$, $II^{en}=4$, $III^{en}=9$.

We will use mod 11 for our calculations again.

The plain text will consist also of 6 bytes $C=\{4,6,7,9,1,8\}$, $\gamma_1=4$. With the help of formulas (3) – (4) we get:

$c_{-1,0}=c_0=4$, $c_{-1,3}=c_4=1$, $c_{-1,4}=c_5=8$.

Elements $c_{-1,1}$ and $c_{-1,2}$ we calculate by formulas (5) – (6):

$c_{-1,1}=(10\cdot(4-6)+6))(\mathrm{mod}11)=8$,

$c_{-1,2}=(4\cdot 10(4-6)+4\cdot(6-7)+7)(\mathrm{mod}11)=0$.

Element $b_{-1}=(9-0+9\cdot(0-1))(\mathrm{mod}11)=0$.

We obtain $c_{-1}=\{4,8,0,1,8\}$, $\quad b_{-1}=0$.

After the shift, our text for the next round will be $\{8,4,8,0,1,8,0\}$.

## 3.4. Process of deciphering

The process of decryption goes by analogy with the process of encryption.

For the restoring of information we will use formulas of reconstruction from wavelet theory. For deciphering we need the reverse order of round keys.

1. We write formulas of reconstruction for the splines of third degree:

$$c_{-K+1,j}=c_{-K,j} \quad \text{if} \quad 0\leqslant j\leqslant \gamma_K-4, \tag{8}$$

$$c_{-K+1,j}=c_{-K,j-1} \quad \text{if} \quad \gamma_K\leqslant j\leqslant M-K, \tag{9}$$

$$c_{-K+1,\gamma_K-3}=\left(I^{de}\cdot(c_{-k,\gamma_k-4}-c_{-k,\gamma_k-3})+c_{-k,\gamma_k-3}\right)(\mathrm{mod}N), \tag{10}$$

$$c_{-k+1,\gamma_k-2}=\left(II^{de}\cdot(c_{-k,\gamma_k-3}-c_{-k,\gamma_k-2})+c_{-k,\gamma_k-2}\right)(\mathrm{mod}N), \tag{11}$$

$$c_{-k+1,\gamma_k-1}=\left(III^{de}\cdot(c_{-k,\gamma_k-2}-c_{-k,\gamma_k-1})+c_{-k,\gamma_k-1}+b_{-k}\right)(\mathrm{mod}N). \tag{12}$$

2. At the end we make a shift of sequence $c_{-1,i}$ as follows:

$$c_{-1,0}\leftarrow c_{-1,1}\leftarrow c_{-1,2}\ldots\leftarrow c_{-1,M-K+1}\leftarrow c_{-1,0}$$

On the next round we will be working with the set $X_{-K+1}$. We will also use $\{c_{-K+1,i}\}_{i=0,....,M-K+1}$ and $b_{-K+1}$, on the second round of the process of deciphering we get the sequence $\{c_{-K+2,i}\}_{i=0,....,M-K+2}$ etc.

**Example:**
To illustrate the process of enciphering we will restore information $\{4, 6, 7, 9, 1, 8\}$, if we know ciphertext $\{4, 8, 0, 1, 8\}$, $b_{-1} = 0$ and we know round key $I^{de}$, $II^{de}$,$III^{de}$.

With the help of formulas (8) – (12) we get:

$c_0 = c_{-1,0} = 4,$

$c_4 = c_{-1,3} = 1,\ c_5 = c_{-1,4} = 8,$

$c_1 = \Big(6 \cdot (4 - 8) + 8\Big)(\text{mod}11) = 6,$

$c_2 = \Big(5 \cdot (8 - 0) + 0\Big)(\text{mod}11) = 7,$

$c_3 = \Big(3 \cdot (0 - 1) + 1 + 0\Big)(\text{mod}11) = 9.$

We obtain the plain text $c = \{4, 6, 7, 9, 1, 8\}$.

This small example just illustrated how formulas of reconstruction and decomposition works.

## 4. Implementation

This algorithm has been implemented on the 32-bit processors. These results were obtained with Java, which does not provide the best of possible results.

The speed figures given in the Table 3 have been scaled on the Pentium 2,33 GHz. It was calculated only for blocks equal to 32, 64, 128 bytes, and it is far from the best results which can obtained.

| Block length, key length | Encryption time for the block | Cycles per bytes for encryption |
|---|---|---|
| 32 bytes, 63 bytes | 0,00004 sec | 2920 |
| 64 bytes, 127 bytes | 0,00008 sec | 2864 |
| 128 bytes, 255 bytes | 0,000189 sec | 3398 |

Table 3. Performance for the process of enciphering (Java).

| Block length, key length | Decryption time for the block | Cycles per bytes for encryption |
|---|---|---|
| 32 bytes, 63 bytes | 0,000051 sec | 3653 |
| 64 bytes, 127 bytes | 0,0001 sec | 3589 |
| 128 bytes, 255 bytes | 0,000238 sec | 4273 |

Table 4. Performance for the process of deciphering (Java).

Results which are presented in this work are very far from the results which can be obtained, C++ will give us a much better result and the program can be optimized.

## 5. Strength against known attacks

The presented algorithm has an absolutely new structure and it is difficult to analyze the strength of this algorithm and other algorithms based on wavelet decomposition of splines.

### 5.1. Differential cryptanaysis

Differential cryptanalysis [1] attacks are possible if there are predictable difference in propagations over all but a few rounds larger than $2^{1-M}$, if M is the block length. Usually they are based on the analysis of results of S-boxes, which we do not have in these algorithms.

If we will check formulas of ciphering (5) – (7) we can see that the XOR operation will not give us any information. Lets examine these formulas. Elements of one plain text is $c_i$ and the other is $c_i^{'}$.

$$I^{en} \cdot (c_{\gamma_1-4} - c_{\gamma_1-3}) + c_{\gamma_1-3} \otimes I^{en} \cdot (c^{'}_{\gamma_1-4} - c^{'}_{\gamma_1-3}) + c^{'}_{\gamma_1-3}$$
$$I^{en} \cdot II^{en}(c_{\gamma_1-4} - c_{\gamma_1-3}) + II^{en} \cdot (c_{\gamma_1-3} - c_{\gamma_1-2}) + c_{\gamma_1-2} \otimes$$
$$I^{en} \cdot II^{en}(c^{'}_{\gamma_1-4} - c^{'}_{\gamma_1-3}) + II^{en} \cdot (c^{'}_{\gamma_1-3} - c^{'}_{\gamma_1-2}) + c^{'}_{\gamma_1-2}$$
$$c_{\gamma_1-1} - c_{-1,\gamma_1-2} + III^{en} \cdot (c_{-1,\gamma_1-2} - c_{-1,\gamma_1-1}) \otimes$$
$$c^{'}_{\gamma_1-1} - c^{'}_{-1,\gamma_1-2} + III^{en} \cdot (c^{'}_{-1,\gamma_1-2} - c^{'}_{-1,\gamma_1-1})$$

As we can see, the data from the parts is all that we can gather; where we have $I^{en}$, $II^{en}$, $III^{en}$:

$$I^{en} \cdot (c_{\gamma_1-4} - c_{\gamma_1-3}), \quad I^{en} \cdot II^{en}(c_{\gamma_1-4} - c_{\gamma_1-3}) + II^{en} \cdot (c_{\gamma_1-3} - c_{\gamma_1-2}),$$
$$III^{en} \cdot (c_{-1,\gamma_1-2} - c_{-1,\gamma_1-1}).$$

These formulas do not give to us any predictable difference in propagations, because in each round $I^{en}$, $II^{en}$, $III^{en}$ are different and they depend only on the round key. It depends on $\gamma_i$ in $i$ - th round, as they are different on each round.

### 5.2. The Square attack

The Square attack [2] based on the analysis of chosen plaintexts of which part is held constant and another part varies through all possibilities.

Here we can see the same situation as with the differential cryptanalysis. The process of enciphering is based on the multiplication of elements of plain text on the elements $I^{en}$, $II^{en}$, $III^{en}$. Changes of plaintexts would not give us any information about round key and the key.

## 5.3. Linear cryptanalysis

For the linear cryptanalysis [7] we need to obtained a linear approximation of the form:

$$P_{i1} \otimes P_{i2} \otimes ... \otimes P_{ia} \otimes c_{j1} \otimes c_{j2} \otimes ... \otimes c_{jb} = K_{k1} \otimes K_{k2} \otimes ... \otimes K_{kn}$$

where $P_n, c_n, K_n$ bytes of plaintext, ciphertext and key.

This attack is also based on the analysis of S-boxes, which we are not considering. Yet it has not been discovered to be the way to obtain linear approximation.

## 5.4. Possible attack

There is an attack which is possible to apply to the algorithm in the way it appears to look at this juncture.

As we see from formulas $(5)-(7)$, two of equations are linear and one is square. We will use a plain text attack and we need a possibility to get cipher bytes after each round. If we will get such possibilities for restoring a key we will need to use $2^{8 \cdot 2 \cdot K}$, where $K$ is a number of round. If we will find $I^{en}$ we will find $II^{en}$, but $III^{en}$ we can not find this way.

Results of this attack is presented in the Table 5.

| | results of attack |
|---|---|
| M = 8 bytes | $2^{96}$ |
| M = 16 bytes | $2^{224}$ |
| M = 24 bytes | $2^{352}$ |
| M = 32 bytes | $2^{480}$ |
| M = 64 bytes | $2^{992}$ |
| M = 128 bytes | $2^{2016}$ |
| M = 256 bytes | $2^{4064}$ |

Table 5.

We can avoid this attack if we will encipher each byte on the round; it will require more time but it will be secure.

# 6. Conclusion

We have presented block cipher based on wavelet decomposition of splines of the third degree. Research for using spline-wavelet decomposition can be applied in different areas of cryptography and the presented algorithm can be improved on and can show better results in security and speed.

This is new way of creating block ciphers which are only based on the mathematics, which can help in the analysis and proofing of strength.

## References

[1] **E. Biham and A. Shamir**, *Differential cryptanalysis of DES-like cryptosystems,* J. Cryptology **4** (1991), 3 – 72.

[2] **J. Daemen, L.R. Knudsen and V. Rijmen**, *The block cipher Square*, Fast Software Encryption, LNCS **1267** (1997), 149 – 165.

[3] **J. Daemen, V. Rijmen**, *The Design of Rijndael: AES - The Advanced Encryption Standard*, Springer, (2002).

[4] **Y.K. Demjanovish**, *Minmal splines and splaches*, Vestnic of St. Petersburg Univ. **1** (2008), 8 – 22 (in Russian).

[5] **Y.K. Demjanovish**, *Splaches and minimal splines*, St. Petersburg, (2003), 200 – 207 (in Russian).

[6] **Y.K. Demjanovich**, *On wavelet decompositions of linear space for arbitrary field and some applications*, J. Math. Modelling B **20** (2008), 104 – 108 (in Russian).

[7] **M. Matsui**, *Linear cryptanalysis method for DES cipher*, Advances in Cryptology, Proc. Eurocrypt93. LNCS **765**, (1994), 386 – 397.

[8] **AES page** http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[9] **DES and 3 DES page** at http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf

[10] **GOST 2814789 page** http://gostshifr.narod.ru/

[11] **Threefish page** http://www.skein-hash.info/

Received February 1, 2014
Revised 17 December 2014

ITMO University
49 Kronverksky Ave.
St.Petersburg, 197101
Russia
E-mail: levina@cit.ifmo.ru