Block ciphers based on wavelet decomposition of splines

Alla Levina

St. Petersburg State University

This paper present new symmetric cryptoalgorithms based on wavelet decomposition of splines first, second and third degree on a mesh.

- **Spline** is a special function defined piecewise by polynomial
- **Wavelet** is a mathematical function used to divide a given function or continuous-time signal into different scale components.

$$X: \quad \ldots < x_{-1} < x_0 < x_1 < \ldots$$

$$\omega_j(t) = (t - x_j)(x_{j+1} - x_j)^{-1} \quad \text{if} \quad t \in [x_j, x_{j+1}),$$
$$\omega_j(t) = (x_{j+2} - t)(x_{j+2} - x_{j+1})^{-1} \quad \text{if} \quad t \in [x_{j+1}, x_{j+2}),$$
$$\omega_j(t) = 0 \quad t \notin [x_j, x_{j+2}], \quad \text{if } \operatorname{supp} \omega_j = [x_j, x_{j+2}].$$

$$\overline{x}_j = x_j \text{ if } j \le k, \text{ and } \overline{x}_j = x_{j+1} \text{ if } j \ge k+1, \quad \xi = x_{k+1},$$

$$\overline{X}: \quad \ldots < \overline{x}_{-1} < \overline{x}_0 < \overline{x}_1 < \ldots$$

$\overline{\omega}_k$:

$$\overline{\omega}_{k-1}(t) = \omega_{k-1}(t) + \overline{\omega}_{k-1}(x_{k+1})\omega_k(t),$$
$$\overline{\omega}_k(t) = \overline{\omega}_k(x_{k+1})\omega_k(t) + \omega_{k+1}(t).$$

$$c_0, c_1, ..., c_{2L-1}$$
$$a_j = (c_2 j + c_{2j+1})/2$$
$$b_j = (c_2 j - c_{2j+1})/2$$
$$c_2 j = a_j + b_j, c_2 j + 1 = a_j - b_j$$

The presented algorithms has the Feistel Structure, but there is no XOR operation with the round key. Algorithms have easy mathematical structure. Now is making researches to analyze the secrecy of providing algorithms, as it have been shown already presented algorithms are strong to brute-force attacks and to some types of chosen-plaintext attacks.

A process of enciphering and deciphering consists of K identical rounds, all calculations are carried out by simple module N. *Key* $\mathbb{K} = (X, \gamma)$; here $X = \{x_j\}_{j=0,\dots L-1}$ is a mesh, where L is number of nodes in the mesh X. $\gamma = \{\gamma_n\}_{n\in[1,\dots,K]}$, is the order of nodes removed from the mesh; on each round only one node is removed from the mesh and γ_n is the number of casually chosen node x_j. A sequence $C = \{c_i\}_{i\in\ Z}$, $|c_i| = M$ is a plaintext; $|c_i|-$quantity of elements which are ciphered.

This algorithm can work with the block length up to 1024 bit. In the table is presented number of rounds, key length as a function of the block length for first order splines

	$(K, \mathbb{K}_{X\gamma})$	L
$M = 128$	$(14, 248)$	17
$M = 256$	$(30, 504)$	33
$M = 512$	$(62, 1016)$	65
$M = 1024$	$(126, 2040)$	129

where M is a block length, K— number of rounds, L — number of nodes in the mesh, $\mathbb{K}_{X\gamma}$ key length.

Key length is equal to (number of rounds + 3)+(number of rounds) bytes. On each round round key will became smaller on 2 bytes. Using of splines and their wavelet decompositions lead to rather wide variety of the keys defined by mesh and order of ejection of nodes.

The offered algorithms can be also applied to the key transfer.

On each round one node with the number γ_j is taken out from the mesh, where j is number of the round. The process of enciphering is based on the formulas of decomposition from wavelet theory; as result we get sequence $\{c_{-K,i}\}_{i\in\ Z}$. The plaintext is restored with the help of formulas of reconstruction from wavelet theory.

Let us write down and count formulas of decomposition for the splines of the first degree:

$$c_i^{-1} = c_i \quad \text{if} \quad 0 \le i < \gamma_1 \tag{1}$$

$$c_i^{-1} = c_{i+1} \quad \text{if} \quad \gamma_1 \le i \le M-1 \tag{2}$$

$$b^{-1} = c_{\gamma_1} - (x^{-1}_{\gamma_1+1} - x^{-1}_{\gamma_1})(x^{-1}_{\gamma_1+1} - \xi)^{-1} c_{\gamma_1-1} - (x^{-1}_{\gamma_1} - \xi)(x^{-1}_{\gamma_1+1} - \xi)^{-1} c_{\gamma_1+1} \tag{3}$$

As a result after K rounds we have got two sequences

$$\{b^{-n}\}_{n=1,2,\dots,K}, \quad \{c_i^{-K}\}_{i=0,1,2,\dots,M-K}.$$

Sequence $\{c_i^{-K}, b^{-n}\}_{n=1,2,\dots,K;i=0,1,2,\dots,M-K}$ *is the ciphertext.*

We write down and count formulas of reconstruction for the splines of the first degree:

$$c_i^{-K+1} = c_i^{-K} \quad \text{if} \quad 0 \le i < \gamma_K \tag{4}$$

$$c_i^{-K+1} = c_{i-1}^{-K} \quad \text{if} \quad \gamma_K + 1 \le i \le M-K \tag{5}$$

$$c_{\gamma_k}^{-K+1} = (x^{-K}_{\gamma_K+1} - x^{-K}_{\gamma_K})(x^{-K}_{\gamma_K+1} - \xi)^{-1} c^{-K}_{\gamma_K-1} + (x^{-K}_{\gamma_K} - \xi)(x^{-K}_{\gamma_K+1} - \xi)^{-1} c^{-K}_{\gamma_K} + b^{-K} \tag{6}$$