# Performing Data Ciphering with the Help of Digital Filters

Ihor Koriakov[1], Oleksandr Pliushch[1], and Serhii Toliupa[1]

[1] *Taras Shevchenko National University of Kyiv, 60 Volodymyrska str., Kyiv, 01601, Ukraine*

**Abstract**

This paper deals with block and stream ciphers as the structures analogous to those of digital filters. A cipher implemented in the form of the canonical structure of the digital filter is considered. This cipher has a reversibility property when the points of transforms corresponding to the coefficients of the forward and backward branches of the filter are interchanged. In this implementation, for the block ciphers input block is the initial state of the filter, subkeys are the filtered input sequences of the filter's samples and the output block is the end-state of the filter. For the stream ciphers, the key is the initial state of the filter and gamma is the filter's output for the zero input sequence. Finally, an example of the stream cipher implemented in the form of the structure of the second-order infinite impulse response filter is presented. For hardware implementation, this stream cipher has the ultimate speed performance.

**Keywords**

Cryptography, block cipher, stream cipher, digital filter, canonical form.

## 1. Introduction

Cryptography is one the most important areas in modern communications, not least because it applies to pervasive computing and the Internet of Things [1, 2]. There are conflicting requirements when it comes to modern ciphers [3, 4]. On the one hand, they should be robust enough against any attempts to crack them, while on the other hand—they are supposed to be fast when it comes to encoding and decoding operations to secure required data transmission rates [5]. There are many studies aimed at resolving this contradiction [6–10]. The most promising way is to use hardware implementations of the cyphering and deciphering procedures [11, 12].

Surprisingly enough, there is an analogy between operations performed during these procedures and those carried out in digital filters. This paper draws the parallel between the two and determines that using an implementation of the coding and decoding operations embedded in the structure of the corresponding recursive digital filter permits achieving the ultimate speed that is of one tact encoding.

## 2. Structures Comparison of the Digital Filters and Ciphers

Traditionally, it is accepted to present block cipher structures as a sequence of operations. For example, an Advanced Encryption Standard (AES) cipher round is depicted in the form of a series of four consecutive operations, as shown in Fig. 1:

- SubBytes (state).
- ShiftRows (state).
- MixColumns (state).
- AddRoundKey (state).

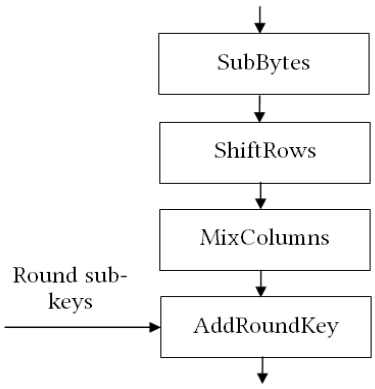


**Figure 1:** Traditional structure of cipher round

CPITS-2024: Cybersecurity Providing in Information and Telecommunication Systems, February 28, 2024, Kyiv, Ukraine
EMAIL: ikor@i.ua (I. Koriakov); oleksandr.pliushch@knu.ua (O. Pliushch); serhii.toliupa@knu.ua (S. Toliupa)
ORCID: 0009-0009-8776-1032 (I. Koriakov); 0000-0001-5310-0660 (O. Pliushch); 0000-0002-1919-9174 (S. Toliupa)

There are a lot of various objects with the same structural features that are analogous to the operation of division by a polynomial. For example, analogs of division by a polynomial are the following elements:

- Integrator.
- Filter with Infinite Impulse Response (IIR).
- Self-synchronizing scrambler.
- Encoder in the gaming mode with Cipher Feedback (CFB).

On the other hand, among the objects that are analogous to polynomial multiplication, one can name the following:

- Differentiator.
- Filter with Finite Impulse Response (FIR).
- Self-synchronizing descrambler.
- Decoder in the CFB mode.

Some filters combine the operations of the polynomial multiplication and division. For example, decimating the Cascaded Integral-Comb (CIC) filter performs first polynomial division and then polynomial multiplication.

Another example is an interpolating CIC filter, which first performs polynomial multiplication and after that—polynomial division.

There are plenty of examples that show that encoding operations can be presented as an IIR filtering process. In particular, the iterative block cipher type of Substitution-Permutation Networks (SPN), which belongs to a well-known AES standard, can be implemented with the help of the IIR filter structure with polynomial division. Fig. 2 illustrates the first-order IIR filter structure with transfer function as follows:

$$H(z) = \frac{1}{A(z)} = \frac{1}{1 - a_1 z^{-1}} \qquad (1)$$

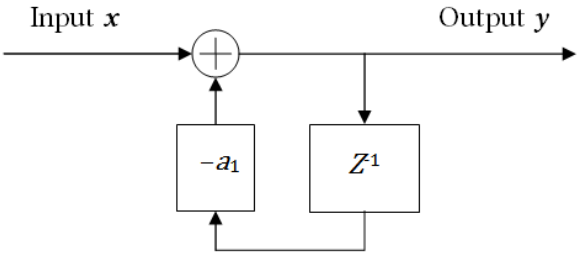


**Figure 2:** Structure of first-order IIR filter

In (1), $Z^{-1}$ represents a one-sample delay element (usually parallel register), $-a_1$ denotes the multiplier by $-a_1$ coefficient, and + stands for the adder. The filter is fed with signal samples *x*, while the resulting samples *y* are formed at the output.

The operation block diagram of the SPN cipher, which is suitable for implementation in the form of the filter, is shown in Fig. 3. Let us consider its elements.

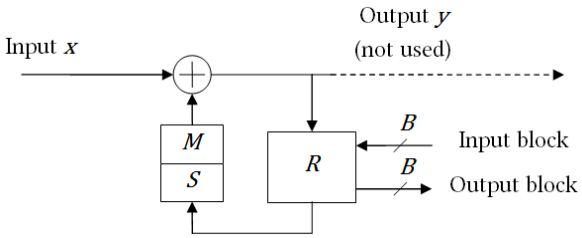


**Figure 3:** Structure of SPN encoder

In Fig. 3, *R* is a parallel register, which stores current values of *B* bits of the cipher; *S* and *M* represent round function (substitution and permutation, correspondingly); and + denotes adder (modulo-two addition).

If one compares the operation block diagram in Fig. 3 with the structure of the first-order IIR filter in Fig. 2, it is easy to see that the multiplier by $-a_1$ in Fig. 2 is replaced by the functional transformer in Fig. 3 that implements the round function; the input in Fig. 3, equivalent to the input of the filter in Fig. 2, is fed with subkeys *x*. The initial state of the operation block diagram in Fig. 3 is the input data block to be encoded; the state of the operation block diagram in Fig. 3 after several steps represents the resulting encoded block, while output *y* of the operations block diagram in Fig. 3 is not used.

Fig. 4 illustrates the structure of the second-order IIR filter with the following transfer function:

$$H(z) = \frac{1}{A(z)} = \frac{1}{1 - a_1 z^{-1} - a_2 z^{-2}}. \qquad (2)$$

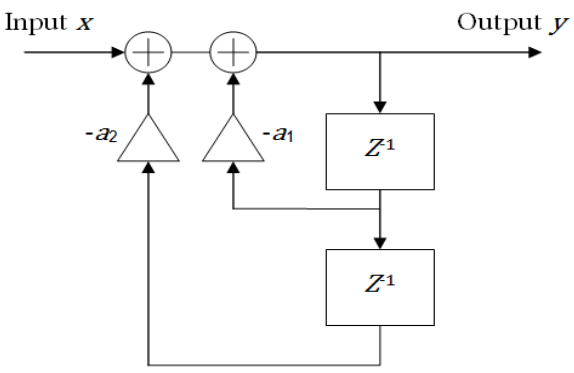


**Figure 4:** Structure of second-order IIR filter

Fig. 4 depicts a second-order filter with two delay elements $Z^{-1}$ and two constant coefficients $-a_1$ and $-a_2$, by (2).

This is the Feistel scheme, implemented in the State Standard of Ukraine №28147, in which coefficient $-a_2$ equals unity, while $-a_1$ is represented by subsequent transformations $S$ (replacement nodes) and $M$ (cyclic shift), as shown in Fig. 5.

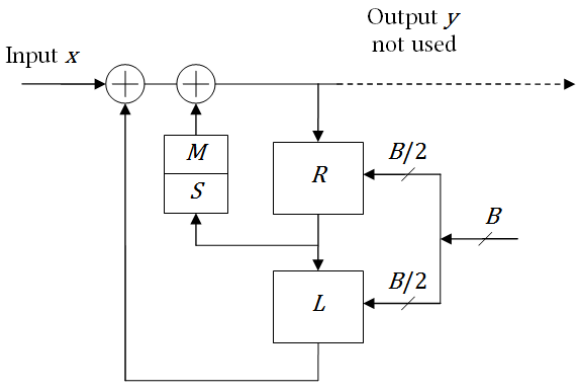


**Figure 5:** Structure of the second-order IIR filter that implements encoding scheme from the State Standard of Ukraine №28147

In this case, the input is not represented by the signal $x$, but rather by the state of the delay elements $R$ and $L$, both the right and the left parts, each with the size B/2, of the input block with the size $B$, while $x$ is just the sequence of the subkeys. Filter output $y$ is not used. The results of the transform are the states of the delay elements after filtration of the $r$ samples (rounds number) of the input vector $x$. For example in the State Standard of Ukraine, №28147 input $B$ constitutes 64 digits, registers $R$ and $L$ have 32 digits, $S$ is eight 4-digit tables and $M$ represents cyclic shift. Subkeys are 32 integer samples each with 32 digits. Addition operation with subkeys is performed at a somewhat different point (not after the delay in the register $R$, but rather before it); the position of this point does not influence the cipher properties. The operations are non-linear and are performed in the different rings with powers 1, 4, and 32, which makes this filter non-linear.

Let us consider another such IIR filter with two unity coefficients $a_i = 1$ and $a_j = 1$ while the other coefficients equal zeros.

Such a filter will have one pole, situated on the unity circle in the $z$-plane, and will be unstable so that its pulse response does not subside. Filter implementation in the form of the delay line, the input of which is fed with the sum of the signals at the taps $i$ and $j$ will just represent the generator with the no-zero initial state of the delay line. Fairly quickly, such a generator (under usual integer arithmetic without saturation) will break out of the linear mode and will overflow by the module of the bit depth. If $i = 24$ and $j = 55$, then it turns into a version of delayed Fibonacci filters, which is widely used in cryptography (pairs of taps $i$ and $j$ might be different, the period of maximum length is given by only some of the pairs).

For synchronous stream cipher, as a rule, key $x$ (subkeys) at the initial stage forms the state of the registers, and then input $x$ assumes zero value and the sequence of the cipher states is determined by the functions $a$, while functions $b$ determine the output values of $y$, which is used as a gamma of the cipher. In particular, when the bit depth equals unity, and the coefficients amount to "0" or "1", one arrives at the classical structure of the Linear Feedback Shift Register (LFSR).

If this structure is used as a stream cipher, of which input $x$ during encoding will be plaintext, and output $y$ will represent ciphertext, then for decoding operation it is necessary to interchange coefficients $a$ and $b$: in other words, interchange numerator and denominator.

All this allows one to come to a very simple but at the same time promising conclusion: block and stream ciphers can be presented structurally by similar schemes, which in turn are in many respects similar to classical structures of the digital filters.

## 3. Ciphers with Canonical Structure of the Digital Filter

The transfer function of the recursive digital filter looks as follows:

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{m=0}^{M} b_m z^{-m}}{1 - \sum_{n=1}^{N} a_n z^{-n}} \qquad (3)$$

where $b_m$ is the coefficients of the non-recursive filter section, $a_n$—coefficients of the recursive section, and $z^{-t}$ is the signal's delay for $t$ samples.

Under the conditions $N = M$ and $b_0 = 1$, the reverse filter is feasible and can be obtained by

interchanging in (3) nominator and denominator:

$$H^{-1}(z)=\frac{A(z)}{B(z)}=\frac{\sum_{n=1}^{N}a_n z^{-n}}{1-\sum_{m=1}^{M}b_m z^{-m}} \qquad (4)$$

which from the output sequence of the forward filter restores its input one.

Filter's difference equation can be presented in the following form:

$$y(k)=x(k)-\sum_{n=1}^{N}a_n D_n(k-1)+\sum_{n=1}^{N}b_n D_n(k-1) \qquad (5)$$

where $y_k$ is the output sample of the filter, $x_k$ is the input sample of the filter, $D_n()$ is the value of the signal at the $n$-th tap of the delay line, which is determined as follows:

$$D_n(k)=D_{n-1}(k-1)\,|\,n=N,\ldots,2 \qquad (6)$$

and

$$D_1(k)=x(k)-\sum_{n=1}^{N}a_n D_n(k-1)\cdot \qquad (7)$$

The Canonical Structure of the Filter (CSF), corresponding to the equations (4)–(7), can be presented in the form shown in Fig. 6.

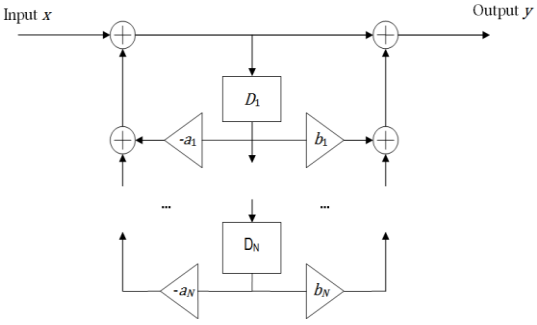


**Figure 6:** Canonical structure of the filter

The reversibility property of such a structure (restoration of the input signal from the output one) under commutation of the coefficients $a_n$ and $b_n$ is preserved, even if the addition operation is replaced by any commutative operation, and multiplication—by any functional transformation, including irreversible one.

CSF allows designing the cipher in which encoding and decoding schemes differ by the interchange of the transfer points for $a_n$ and $b_n$.

If one fixes the size of the internal state of the cipher (for example, 128 bits), then the filter order $N$ and cipher width $B$ (bit depth of the delay line registers) can be scalable. This scalability holds up for $N = 1$ and $B = 128$ as well as up to $N = 128$ and $B = 1$. For example, if one sets the element to be encoded as a bite of data, then one gets the cipher with $N = 16$ and $B = 8$.

In this case, in a certain sense, $N$ appears as an analog of the number of block cipher rounds, while the transforms $a_n$ and $b_n$ are analogs of the round functions.

And, if for $B = 1$, the functions are set by the simple substitutions: input values, inverse values of the input, constant "0" and constant "1" (this leads to classical stream ciphers with single digit-depth registers), for $B = 8$ the most reasonable seems to be random permutations 8×8, set by the table; for $B = 128$, the problem requires a special consideration.

Let us consider a specific version of the CSF cipher with $N = 1$ and $B = 128$, for which encoding and decoding schemes are presented in Fig.7.

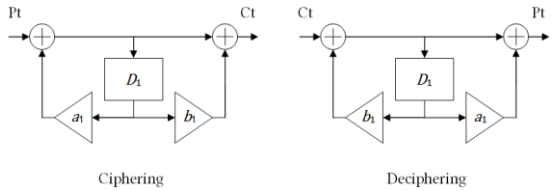


**Figure 7:** CSF cipher scheme with $N = 1$ and $B = 128$

In Fig. 7, $D_1$ is a delay element in the form of a 128-digit parallel register; and $a_1$ and $b_1$ are functional transforms, approximated to fixed random substitutions with the size 128×128. Addition operations are presented by 128-digit modulo-two adders.

Transform dependence on the key can be introduced either into the initial state (as in stream ciphers), or into functional transforms, for example. For the case presented in Fig. 8, the 256-bit-long key is split into two halves: $K_1$ and $K_2$, which are added on modulo-two addition with the inputs and outputs of the functional transformers $a_1$ and $b_1$ by the Even–Mansour cipher [6].

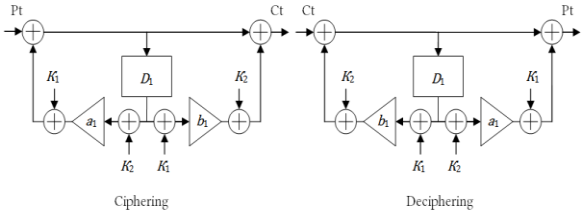


**Figure 8:** Example of the scheme of subkeys introduction into functional transforms

Functional transformers $a_1$ and $b_1$ can be implemented in the form of "wide" $S$-blocks, comprising each, for instance, 16 tables of fixed

random substitutions $S_0$—$S_{15}$ with the size 8×128. Concatenation of the 16 inputs of these blocks forms the 128-bit input of the transform, while the modulo-two addition of all 16 128-bit table inputs creates the output of the transform. The structure of such a functional transformer is shown in Fig. 9.

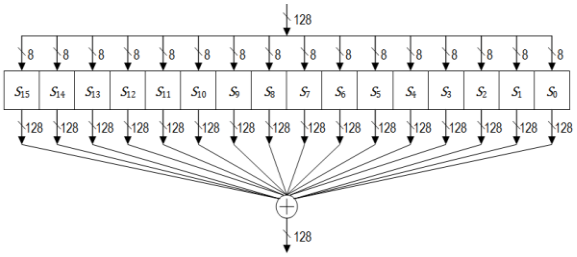


**Figure 9:** Diagram of the "wide" S-block

Any selection of the S-blocks is most likely not required because the combinational power of such a block is very large (2 power around 500000) and the probability of getting a randomly "bad" block is negligibly small. Such *S*-blocks for encoding *N*B* bits are used once, while in SPN ciphers one *S*-block is used repeatedly a hundred times for encoding one block of data, which requires a very strict approach to its formation. Of course, memory volume for the realization of "wide" *S*-blocks is rather significant—128 Kbite for a certain example, but currently such a volume is a fairly available resource.

A supposed advantage of CSF class ciphers appears to be monotonous scalability on *N* and *B*, which in addition to cipher adaptation to the particular tasks allows one to research reduced versions of the ciphers. In addition, hardware implementation of the cipher permits reducing the encoding time of *B* bits to one tact encoding.

## 4. Example of the Cipher with the Structure of the 2-Order Filter

In [7], it is considered a stream cipher named "Krip" with the structure of 2nd order IIR filter that comprises functional transforms φ and ψ, which correspond to *a* and *b* coefficients in the filter structure.

As the transform φ, round function of the cipher "Kalyna" with a block length of 256 bits is used, and as a transform ψ is a round function of the same cipher with a block length of 512 bits [8].

Fig. 10 demonstrates the diagram of the "Krip" cipher in the encoding mode. As the key, initial states of the cipher *S*1 and *S*2 with a total length of 512 bits are used, while vector *X* is a plaintext and vector *Y* is a ciphertext.

While encoding with the help of the transform starting from S1, S2, a gamma vector is formed that is summed with the vector of the plaintext *X* to obtain the vector of the ciphertext *Y*. Plaintext vector *X* is used for forming a new value of the state *S*1, while the previous value of *S*1 is transferred into *S*2.

During decoding, from *S*1, *S*2 with the help of the transform, a gamma vector is formed, which is added to the vector of the ciphertext *Y* to obtain the vector of the plaintext *X*. Plaintext vector *X* is then used for forming a new value of the state *S*1, while the previous value of *S*1 is transferred into *S*2.

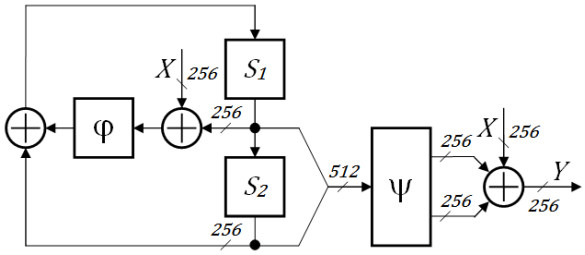


**Figure 10:** Cipher scheme in the encoding mode

Fig. 11 demonstrates the scheme of the "Krip" cipher in the gaming mode (CTR). Here, the first encoded vector $Y_0$ is used as an initial value of the counter *t* that is incremented in every following encoding tact to form a new value of $S_1$.

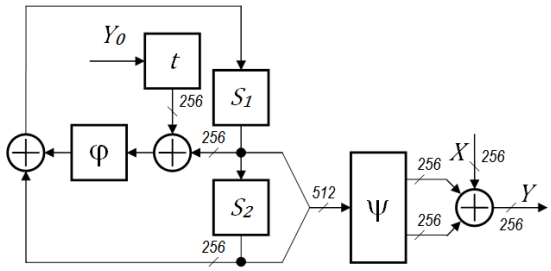


**Figure 11:** Cipher scheme in gaming mode

Such a stream cipher with the hardware implementation in the form of the second-order IIR filter wields maximum speed performance by encoding input vector in just one tact of operation.

## 5. Conclusions

Encoding and decoding operations can be presented as filtering processes in IIR filters. This is true for both block ciphers and stream

ciphers. This allows one to come to a very clear and promising conclusion: block and stream ciphers can be represented structurally by similar schemes, which in turn are in many respects similar to classical structures of the IIR filters.

The example of the stream cipher with the hardware implementation in the structure of the 2nd order IIR filter secures maximum speed performance, which allows encoding input vector in just one tact of operation.

## References


[1] A. Kuznetsov, et al., Stream Ciphers in Modern Real-time IT Systems: Analysis, Design and Comparative Studies, Studies in Systems, Decision and Control 375 (2022) doi: 10.1007/978-3-030-79770-6.

[2] M. Khairallah, Hardware Oriented Authenticated Encryption Based on Tweakable Block Ciphers, Computer Architecture and Design Methodologies (2022). doi: 10.1007/978-981-16-6344-4.

[3] Z. Hu, et al., Bandwidth Research of Wireless IoT Switches, in: IEEE 15th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (2020). doi: 10.1109/tcset49122.2020.2354922.

[4] V. Sokolov, et al., Method for Increasing the Various Sources Data Consistency for IoT Sensors, in: IEEE 9th International Conference on Problems of Infocommunications, Science and Technology (PICST) (2023) 522–526. doi: 10.1109/PICST57299.2022.10238518.

[5] V. Sokolov, P. Skladannyi, H. Hulak, Stability Verification of Self-Organized Wireless Networks with Block Encryption, in: 5th International Workshop on Computer Modeling and Intelligent Systems, vol. 3137 (2022) 227–237.

[6] S. Even, Y. Mansour, A Construction of a Cipher from a Single Pseudorandom Permutation, ASIACRYPT 1991, LNCS 739 (1993) 210–224.

[7] L. Kovalchuk, I. Koriakov, A. Alekseychuk, KRIP: High-Speed Hardware-Oriented Stream Cipher Based on a Non-Autonomous Nonlinear Shift Register, Cybernetics Syst. Anal. 59(1) (2023) 21–32.

[8] R. Oliynykov, et al., A New Encryption Standard of Ukraine: The Kalyna Block Cipher. Cryptology ePrint Archive, (2015). URL: http://eprint.iacr.org/2015/650

[9] T. Cusick, C. Ding, A. Renvall, Stream Ciphers and Number Theory. Elsevier Science (1998).

[10] G. Baumslag, et al., A Course in Mathematical Cryptography. Walter de Gruyter GmbH (2015).

[11] M. Vladymyrenko, et al., Analysis of Implementation Results of the Distributed Access Control System. in: 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (2019). doi: 10.1109/picst47496.2019.9061376.

[12] A. Carlsson, et al., Sustainability Research of the Secure Wireless Communication System with Channel Reservation, in: 2020 IEEE 15th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (2020). doi: 10.1109/tcset49122.2020.235583.