Advantages of an Automata-Based Stream Cipher

Pál Dömösi Géza Horváth

Faculty of Informatics
University of Debrecen
H-4028 Debrecen
Kassai Road 26.
Hungary
domosi@unideb.hu
horvath.geza@inf.unideb.hu

Abstract
In 2017, the authors described a novel symmetric stream cipher based on finite automaton without outputs such that its transition table forms a Latin square. [4] The system is called DH3 cipher, and is patented in the USA and the EU. [5, 6] In this paper we are going to compare this cryptosystem to the well-known and widely used Vernam cipher, [10] which is also known as OTP (one-time pad) system. The popularity of the OTP system is based on the property called "perfect secrecy". The concept of perfect secrecy was first defined by Claude Shannon in 1946, although the work was not declassified and published until three years later. [8] We are going to show that the DH3 cipher also satisfies the perfect secrecy property, and it has many advantages over the Vernam cipher.

1. Introduction

Stream ciphers have a long history, having been used by Julius Caesar in the ancient Roman Empire. They have remained popular ever since. This is due to a number of things. On the one hand, their design is simple, so they can be easily implemented in simple hardware devices, and on the other hand, their speed is very fast. Finally, Vernam's early 20th century system, [10] when applied under the right conditions, meets the criteria for so-called "perfect secrecy", a concept introduced by Claude Shannon in 1949. [8]

However, over the more than 100 years since Gilbert Vernam introduced his system, stream ciphers have faced many new challenges. Particularly in the 21st century, where new forms of attack have emerged, such as side-channel attacks. In this paper, we compare the classical Vernam cipher with the DH3 cipher, introduced in 2017, [4] which is based on the automata theory. We will first prove that the DH3 criptosystem also satisfies the perfect secrecy property and then show that in many cases it is more secure than using the Vernam cipher.

Automata theory provides a natural basis for designing cryptosystems and several such systems have been designed. Some of them are based on Mealy automata or their generalization, some of them are based on cellular automata, while others are based on compositions of automata.

Almost all cryptosystems can be modelled with Mealy machines (as sequential machines) or generalized sequential machines. During encryption the system first receives a preliminary input, which contains the encryption key automaton (and sometimes a secondary encryption key as well). Following this key the input contains the plaintext stream, as a string of input signals for the automaton, and starting the automaton from a given state, the encrypted message can indeed be generated as an output signal string initiated by the input signal string. Decryption is performed in a likewise manner: the encryption key is substituted by the decryption key so that plaintext and ciphertext messages change roles. This method has several variants. Many famous mechanical cryptographic machines are concrete examples of this interpretation. Some of these machines are still used as software versions.

A further generation of the cryptosystems based on Mealy machines is the family of public key FAPKC systems. They use the mathematical conjecture that it is difficult to find the reverse automaton for delayed, weakly invertible automata. Unfortunately these systems can be defeated. So as to prevent attacks there was developed a refinement of this system, called FAPKC-3. There are two methods to break this cryptographic system. Besides the vulnerability of FAPKC and FAPKC-3, the main problem of the most cryptosystems based on Mealy machines is that they suffer from the lack of systematic and massive cryptanalytic research.[1] These facts are serious drawbacks in their practical applications.

Almost from the very beginning of research into cellular automata, there have been serious attempts at cryptographic applications. Cryptosystems of this kind usually use the plaintext as the initial state of the cellular automaton, and the encryption key is the transition rules for the cells. The state reached after a given number of steps provides the encrypted message.

The best-known cellular automaton-based cryptosystems all share the common problem of serious realization difficulties: some systems are easy to defeat, the technical realization of others results in slow performance, and still others exhibit difficulties in the choice of the key-automaton. A further common drawback of cellular automaton-based cryptosystems is that their micro sized technical realization poses serious difficulties and they are not always economical either.

To overcome the discussed problems, a symmetric cryptosystem is described in [1]. It has a Rabin-Scott automaton as key for encryption and decryption. The applied key automaton performs encryption of the plaintext character by character, assigning an encrypted counterpart of variable length to each character, the encryption performs a ciphertext with a length substantially exceeding the length of the plaintext. This system has a serious disadvantage in that the ciphertext is significantly longer than the plaintext, with the ciphertext even being multiple times longer than the plaintext.

[1] Among others, the vulnerability of these systems may be due to the well-known fact that automaton mappings are length and prefix preserving, and that gsm mappings are prefix preserving. Therefore, they seem to be vulnerable to adaptive chosen-ciphertext attacks.

In [2, 3] new block ciphers based on compositions of automata are introduced. Both systems use the following simple idea: Consider a giant-size permutation automaton such that the set of states and the set of inputs consist of all given length of strings over a non-trivial alphabet as all possible plaintext/ciphertext blocks. Moreover, consider a cryptographically secure pseudorandom number generator with large periodicity having the property that, getting its really random kernel, it serves a sequence of pseudorandom strings as inputs for the automaton. For each plaintext block the system calculates the new state into which the recent pseudorandom string takes the automaton from the state which is identified as the recent plaintext block. The string, identified as the new state, will be the ciphertext block ordered to the considered plaintext block. Of course, the ciphertext will be the catenation of the generated ciphertext blocks. The giant size of the automaton makes it infeasible to break the system by brute-force method.

The problem of this idea is that to store the transition matrix having 2^{128} states and 2^{128} input letters is impossible. The basic idea of the DH3 cipher is to operate on a giant secret transition matrix which is compressed into the memory using automata-theoretic methods. This problem can be overcome considering automata which consists of composition of automata. In this case, we should store only the transition matrix of the component automata and the structure of the composition. Moreover, if the component automata are isomorphic to each other then it is enough to store the transition matrix of one component automaton and the structure of the isomorphisms. By this recognition, the storage of automata having 2^{128} states and 2^{128} input letters can be easily solved. Because of the giant size of the matrix, there is no hope to attack the system by brute-force method. On the other hand, this giant matrix can be generated unambiguously by a bitstring of 782 bytes length. Note that this less than 1 kilobyte long string can be generated by an appropriate hash function using a secret password of any length. This cipher overcomes all of the discussed disadvantages, however DH3 uses more simple operations, because it does not work with automata compositions, the only operator we use is the direct access to an element in the transition matrix.

In this paper, after a short description of the Vernam and DH3 systems, we show that the DH3 cipher also satisfies the perfect secrecy property, just like the Vernam system, but unlike the Vernam system, it is protected against the man-in-the-middle attack on known plaintext. Based on all these features, we have chosen the DH3 cryptosystem as the 21st century successor to the Vernam system for stream encryption.

2. Preliminaries

2.1. Basic terms and notations

We start with some standard concepts and notations. By an *alphabet* we mean a finite nonempty set. The elements of an *alphabet* are called *letters*. A *word* over an alphabet Σ is a finite string consisting of letters of Σ. A word over a binary alphabet is called a *bit string*. The string consisting of zero letters is called the *empty word,* written by λ. The *length* of a word w, in

symbols $|w|$, means the number of letters in w when each letter is counted as many times as it occurs. By definition, $|\lambda| = 0$. At the same time, for any set H, $|H|$ denotes the cardinality of H. In addition, for every nonempty word w, denote by $\overrightarrow{w}$ the last letter of w. ($\overrightarrow{\lambda}$ is not defined.) If $u = x_1 \cdots x_k$ and $v = x_{k+1} \cdots x_\ell$ are words over an alphabet Σ (with $x_1, \ldots, x_k, x_{k+1}, \ldots, x_\ell \in \Sigma$), then their *catenation* $uv = x_1 \cdots x_k x_{k+1} \cdots x_\ell$ is also a word over Σ. In this case we also say that u is a *prefix* of uv and v is a *suffix* of uv. Catenation is an associative operation and, by definition, the empty word λ is the identity with respect to catenation: $w\lambda = \lambda w = w$ for any word w. For every word $w \in \Sigma^*$, put $w^0 = \lambda$, moreover, $w^n = ww^{n-1}, n \geq 1$. Let Σ^* be the set of all words over Σ, moreover, let $\Sigma^+ = \Sigma^* \setminus \{\lambda\}$. Σ^* and Σ^+ are the *free monoid* and the *free semigroup,* respectively, generated by Σ under catenation. In particular, we put $\Sigma^0 = \{\lambda\}, \Sigma^n = \{w : |w| = n\}, n \geq 1$, and $\Sigma^{(0)} = \Sigma^0, \Sigma^{(n)} = \{w : |w| \leq n\}, n \geq 1$. In addition, for every string $x_1 \cdots x_k$ with $x_1, \ldots, x_k \in \Sigma$, the string $x_k \cdots x_1$ is called a mirror image of $x_1 \cdots x_k$. We will use the notation p^R as the mirror image of p for every $p \in \Sigma^+$. Moreover, by definition, let $\lambda^R = \lambda$.

By an *automaton* we mean a deterministic finite automaton without outputs. In more details, an automaton is an algebraic structure $\mathcal{A} = (A, \Sigma, \delta)$ consisting of the nonempty and finite *state set* A, the nonempty and finite *input set* Σ, and a *transition function* $\delta : A \times \Sigma \to A$. The elements of the state set are the *states,* and the elements of the input set are the *input signals.* An element of A^+ is called a *state word* [2] and an element of Σ^* is called an input word. State and input words are also called *state strings* and *input strings,* respectively. If a state string $a_1 a_2 \cdots a_s$ $(a_1, \ldots, a_s \in A)$ has at least three elements, the states $a_2, a_3, \ldots, a_{s-1}$ are also called intermediate states. It is understood that δ is extended to $\delta^* : A \times \Sigma^* \to A^+$ with $\delta^*(a, \lambda) = a$, $\delta^*(a, xq) = \delta(a, x)\delta^*(\delta(a, x), q), a \in A, x \in \Sigma, q \in \Sigma^*$. In other words, $\delta^*(a, \lambda) = a$ and for every nonempty input word $x_1 x_2 \cdots x_s \in \Sigma^+$ (where $x_1, x_2, \ldots, x_s \in \Sigma$) there are $a_1, \ldots, a_s \in A$ with $\delta(a, x_1) = a_1, \delta(a_1, x_2) = a_2, \ldots, \delta(a_{s-1}, x_s) = a_s$ such that $\delta^*(a, x_1 \cdots x_s) = a_1 \cdots a_s$.

In the sequel, we will consider the transition of an automaton in this extended form and thus we will denote it by the same Greek letter δ. If $\overrightarrow{\delta(a, w)} = b$ holds for some $a, b \in A, w \in \Sigma^*$ then we say that w *takes* the automaton from its state a into the state b, and we also say that the automaton *moves* from the state a into the state b under the effect of w.

The transition function can be written in a matrix form, where the state set and the input set are ordered sets, and the element of the matrix in the i-th row and j-th column shows the state into which the automaton moves from its j-th state under the effect of i-th input signal. We call this *transition matrix* a *Latin square* if each row and column is a permutation of states.

2.2. The Vernam cipher

Vernam cipher is based on a very simple operation, the bitwise exclusive-or (XOR) operation. The basic idea is that given the plaintext and an equally long secret key, then the ciphertext characters are obtained by combining the characters of the plaintext with the corresponding

[2] The empty word is not considered as a state word.

characters of the key using the XOR operation. In the case of decryption, the same operation is used, the characters of the ciphertext are combined with the charaters of the key, using the XOR operation, and thus the plaintext is returned.

Example

In the following example, both the plaintext and the key were given in binary form, for the sake of simplicity. This is not a limitation, since all digital data can be entered in binary form.

plaintext: 01011101 10100101

key: 11100110 00110111

For encryption, to calculate the ciphertext, we use the bitwise exclusive-or (XOR) operation on the plaintext and the key. The result is the following:

01011101 10100101 XOR 11100110 00110111 = 10111011 10010010

ciphertext: 10111011 10010010

For decryption, to calculate the plaintext, we use the same bitwise exclusive-or (XOR) operation on the ciphertext and the key. The result is the following:

10111011 10010010 XOR 11100110 00110111 = 01011101 10100101

2.3. The DH3 cipher

The working of the DH3 cipher mainly differs from the Vernam cipher: it does not generate the ciphertexts in such a way that the plaintext bit stream is combined with a key bit stream by an exclusive-or operation (XOR). On the other hand, it has the main property of the stream ciphers: the plaintext digits are encrypted one at a time, and the transformation of successive digits varies during the encryption.

This stream cipher is based on finite automaton without outputs. For encryption the system uses the transition matrix of a key-automaton without outputs as the secret key. This transition matrix forms a Latin square. For decryption the system also has a so-called inverse key-automaton which moves from a given state b under a given input sign x^R into the state a if and only if the original key-automaton moves from the state a into the state b under the effect of x. The input alphabet, the state set, the plaintext alphabet and the ciphertext alphabet coincide. A further element of the system is a pseudorandom generator which generates pseudorandom character strings with a given length.

During encryption the plaintext is read sequentially character by character and the key automaton assigns to each plaintext character the corresponding state which is the same as the read plaintext character. The corresponding ciphertext character will be the state into which the key-automaton moves from the assigned state under the effect of the next (initially the first) pseudorandom string (with a given length). The apparatus creates the ciphertext by linking

these character strings together.

During decryption the ciphertext is read in sequentially character by character and the inverse key automaton assigns to each ciphertext character the corresponding state which is the same as the original plaintext character. The corresponding plaintext character will be the state into which the inverse key-automaton moves from the assigned state under the effect of the mirror image of the next pseudorandom string. The apparatus recreates the plaintext by linking these character strings together.

Next we give a formal description of this system.

Key automaton

Consider an automaton $\mathcal{A} = (A, \Sigma, \delta)$ with $A = \Sigma$, where for every $a, b \in A$ $(a \neq b)$ and $x, y \in \Sigma$ $(x \neq y)$, $\delta(a, x) \neq \delta(b, x)$ and $\delta(a, x) \neq \delta(a, y)$. Thus, $\mathcal{A}$ is a permutation automaton, i.e., each row of the transition matrix forms a permutation of the state set. This is an essential property to ensure the unambiguity of the ciphertext for any plaintext.

For the security, we also assume that all columns of the transition table also form a permutation of the state set.

Let $\mathcal{A}^{-1} = (A, \Sigma, \delta^{-1})$ be the automaton for which $\delta^{-1}(b, x) = a$ with $a, b \in A,\ x \in \Sigma$ if and only if $\delta(a, x) = b$.

In what follows $\mathcal{A}$ will be called the *key-automaton* and $\mathcal{A}^{-1}$ will be called the *inverse key automaton.*

Encryption

Let $p_1, \ldots, p_k \in A$ be a plaintext and let $r_1, \ldots, r_k \in \Sigma^+$ be random strings generated by the pseudorandom number generator starting by a seed r_0. We note that $|r_0|, \ldots, |r_k| = n$ holds for a fixed positive integer n.

The ciphertext will be $c_1 \cdots c_k$ with $c_1 = \overrightarrow{\delta(p_1, r_1)}, \ldots, c_k = \overrightarrow{\delta(p_k, r_k)}$.

Decryption

Let $c_1, \ldots, c_k \in A$ be a ciphertext and let $r_1, \ldots, r_k \in \Sigma^+$ be the same random strings generated by the pseudorandom number generator starting by a seed r_0.

The decrypted plaintext will be $p_1 \cdots p_k$ with $p_1 = \overrightarrow{\delta^{-1}(c_1, (r_1)^R)}, \ldots p_k = \overrightarrow{\delta^{-1}(c_k, (r_k)^R)}$.

Example

In the following simple example we are going to use a key automaton $\mathcal{A} = (A, \Sigma, \delta)$, where $A = \Sigma = \{0, 1, 2, 3\}$. We use the $\mathcal{A}^{-1} = (A, \Sigma, \delta^{-1})$ automaton for decryption.

	A			
δ	0	1	2	3
0	1	2	3	0
1	2	0	1	3
2	3	1	0	2
3	0	3	2	1

	A			
δ^{-1}	0	1	2	3
0	3	0	1	2
1	1	2	0	3
2	2	1	3	0
3	0	3	2	1

plaintext: 0123

pseudorandom strings: 11, 21, 30, 31

ciphertext: 1030

3. Results

3.1. Perfect secrecy

In the 1940s, Claude Shannon invented the concept of perfect secrecy for cryptosytems [8]. The essence of the idea is that the ciphertext should not provide any information about the plaintext beyond the length of the plaintext, i.e. any ciphertext should be equally likely to be associated with any plaintext. This means that even with infinite computational power, it is not possible to get back the plaintext from the knowledge of the ciphertext encrypted by a cipher that satisfies the perfect secrecy property. Cryptosystems satisfying the perfect secrecy property remain secure even after the advent of quantum computers, so they are quantum-safe.

The great advantage of the Vernam system is that it is certified to meet the requirements of perfect secrecy under certain conditions. These conditions are the following:

1. The key must be at least as long as the plaintext.
2. The key must be truely uniformly random.
3. The key must never be reused in whole or in part.
4. The key must be kept completely secret by the communicating parties.

In this paper, we do not wish to go into what we mean by truly uniformly random key, but, because of the difficulty of satisfying all the preconditions in practice, apart from some diplomatic, intelligence and military applications, Vernam system is used in very few cases in a form that satisfies all the preconditions of perfect secrecy. However, since this is an important property, especially for high security systems, now we prove that under the same conditions DH3 also has the perfect secrecy property. The formal definition of the perfect secrecy property is the following.

Definition 3.1 *[9] A cryptosystem has perfect secrecy if the posteriori probability that the plaintext is x, given that the ciphertext y is observed, is identical to the a priori probability that the plaintext is x.*

In the following theorem, a one-time pad cryptosystem is a Vernam system whose keys satisfy the above preconditions.

Theorem 3.2 *[7] The one-time pad is a perfectly-secret encryption scheme.*

Our main new theorem is the following:

Theorem 3.3 *If the keys used meet the requirements of the one-time pad system, DH3 cipher is also a perfectly-secret encryption scheme.*

Proof: We use a constructive proof, in which we show that the DH3 cryptosystem can simulate the operation of the Vernam system in a single round using a special transition matrix. To ensure that the DH3 encryption scheme meets the prerequisites, in this case we will use the required real random numbers instead of pseudorandom numbers. In the proof, we use the 4-bit version of the DH3 system, for which we specify the transition matrix with which the encryption can be performed in a one-round case. This half-byte transition matrix is shown in Figure 1.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	0	3	2	5	4	7	6	9	8	11	10	13	12	15	14
2	2	3	0	1	6	7	4	5	10	11	8	9	14	15	12	13
3	3	2	1	0	7	6	5	4	11	10	9	8	15	14	13	12
4	4	5	6	7	0	1	2	3	12	13	14	15	8	9	10	11
5	5	4	7	6	1	0	3	2	13	12	15	14	9	8	11	10
6	6	7	4	5	2	3	0	1	14	15	12	13	10	11	8	9
7	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8
8	8	9	10	11	12	13	14	15	0	1	2	3	4	5	6	7
9	9	8	11	10	13	12	15	14	1	0	3	2	5	4	7	6
10	10	11	8	9	14	15	12	13	2	3	0	1	6	7	4	5
11	11	10	9	8	15	14	13	12	3	2	1	0	7	6	5	4
12	12	13	14	15	8	9	10	11	4	5	6	7	0	1	2	3
13	13	12	15	14	9	8	11	10	5	4	7	6	1	0	3	2
14	14	15	12	13	10	11	8	9	6	7	4	5	2	3	0	1
15	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Figure 1: The transition matrix for the 4-bit version of the DH3 cipher to simulate the Vernam cipher.

Figure 2 shows the binary version of this transition matrix, and it can be easily verified that the 4-bit, one-round version of DH3 performs exactly the XOR operation between the given 4-bit plaintext portion and the given 4-bit portion of the random number with the given transition matrix.

□

	0000	0001	0010	0011	0100	0101	0110	0111	1000	...	1111
0000	0000	0001	0010	0011	0100	0101	0110	0111	1000	...	1111
0001	0001	0000	0011	0010	0101	0100	0111	0110	1001	...	1110
0010	0010	0011	0000	0001	0110	0111	0100	0101	1010	...	1101
0011	0011	0010	0001	0000	0111	0110	0101	0100	1011	...	1100
0100	0100	0101	0110	0111	0000	0001	0010	0011	1100	...	1011
0101	0101	0100	0111	0110	0001	0000	0011	0010	1101	...	1010
.	.	.	.	.	.	.	.	.	.		.
.	.	.	.	.	.	.	.	.	.		.
.	.	.	.	.	.	.	.	.	.		.
1111	1111	1110	1101	1100	1011	1010	1001	1000	0111	...	0000

Figure 2: The binary version of the transition matrix given in Figure 1.

Our last remark is that based on this proof it is easy to construct the corresponding transition matrix for the one-byte long version of the DH3 cipher, to simulate the operation of the Vernam system.

3.2. Known plaintext attack on the Vernam system

Claiming that the Vernam system is "perfectly-secret encryption scheme" if the right preconditions are met can give a false sense of security to ordinary users. In the following, we describe an attack against the Vernam system that could cause serious problems for users. It is a combination of an active man-in-the-middle attack and the known plaintext attack.

Suppose Alice wants to send an encrypted message to Bob using the Vernam cipher. The key meets all the prerequisites. At the same time, an active attacker can intercept the ciphertext sent by Alice, modify it, and then forward it to Bob. Furthermore, suppose that the attacker also obtains the plaintext of the message. Then the attacker can easily create a fake message and send it to Bob on behalf of Alice. All that is required is to perform an XOR operation on the plaintext and the ciphertext to obtain the key. With this key, he can then encrypt any message he creates - of the same length as the original plaintext - and send it to Bob, making it appear as if the message was sent by Alice.

As we have seen, the Vernam cipher is defenseless against this attack. The DH3 cipher, however, is protected against the same attack method, since the active attacker knows both the plaintext and the ciphertext, but without knowledge of the transition matrix, he cannot determine the key from them, and thus cannot generate the ciphertext for a fake message.

The above example shows that although the Vernam system is a "perfectly-secret encryption scheme", it is not protected against the known plaintext attack, while the DH3 system is protected.

4. Conclusions

The most important proof in this article is that the DH3 cipher is a "perfectly-secret encryption scheme". We have also shown that the Vernam system is unprotected against a known plaintext attack, while the DH3 system is protected. Finally, we have to note that applying multiple rounds improve the security of the DH3 system, while the security of the Vernam system does not change. This makes the HD3 system the most successful candidate to replace the Vernam system.

5. Acknowledgement

This work was supported by the Research Institute for Mathematical Sciences, an International Joint Usage/Research Center located in Kyoto University.

References

[1] P. DÖMÖSI, A Novel Cryptosystem Based on Finite Automata without Outputs. In: M. ITO, Y. KOBAYASHI, K. SHOJI (eds.), *Automata, Formal Languages, and Algebraic Systems*. World Scientific, 2010, 23–32.

[2] P. DÖMÖSI, G. HORVÁTH, A novel cryptosystem based on abstract automata and Latin cubes. *Studia Scientiarum Mathematicarum Hungarica* 52 (2015) 2, 221–232.

[3] P. DÖMÖSI, G. HORVÁTH, A novel cryptosystem based on Gluškov product of automata. *Acta Cybernetica* 22 (2015), 359–371.

[4] P. DÖMÖSI, G. HORVÁTH, A Novel Stream Cipher Based on Deterministic Finite Automaton. *In: R, Freund; F, Mráz; D, Prusa (eds.) Ninth Workshop on Non-Classical Models of Automata and Applications (NCMA 2017), Short Papers, Technical University of Vienna,* (2017), 11–16.

[5] P. DÖMÖSI, G. HORVÁTH, Symmetric Key Stream Cipher Cryptographic Method and Device. *US Patent, No. US11165563B2* (2021).

[6] P. DÖMÖSI, G. HORVÁTH, Symmetric Key Stream Cipher Cryptographic Method and Device. *EU Patent, No. EP3639464* (2021).

[7] J. KATZ, Y. LINDELL, *Introduction to Modern Cryptography*. CRC Press, 2007.

[8] C. E. SHANNON, Communication Theory of Secrecy Systems. *Bell System Technical Journal* 28 (1949) 4, 656–715.

[9] D. R. STINSON, *Crpyptography Theory and Practice*. CRC Press, 1995.

[10] G. S. VERNAM, Secret Signaling System. *US Patent, No. US1310719A* (1919).