A Novel Stream Cipher Based on Deterministic Finite Automaton

Pál Dömösi & Géza Horváth

University of Debrecen
Faculty of Informatics
Department of Computer Science

Topics

1. Cryptosystems Based on Mealy Automata
2. Cryptosystems Based on Cellular Automata
3. Cryptosystems Based on Finite Automata
   (Dömösi Cryptosystem, 2008, stream cipher)
4. Cryptosystems Based on Finite Automata Network
   a) Cryptosystems Based on Gluškov Product of Automata
      (DH1, DH2 Cryptosystems, 2013, block ciphers)
   b) Cryptosystems Based on Temporal Product of Automata
      - Stream Cipher (DH3 Cryptosystem, 2017)
      - Block Cipher (DH4 Cryptosystem, 2017)

Mealy automaton

Definition

A Mealy automaton $\mathcal{M} = (Q, T, V, q_0, \delta, \mu)$ *is a 6-tuple, consisting of the following:*

- Q – *finite set of states,*
- T – *input alphabet,*
- V – *output alphabet,*
- q_0 – *initial state,* $q_0 \in Q$,
- δ – *transition function:* $Q \times T \to Q$,
- μ – *output function:* $Q \times T \to V$.

Example

Example: Mealy automaton

$$\mathcal{M} = (\{a, b, c\}, \{0, 1\}, \{x, y, z\}, a, \delta, \mu)$$

δ, μ	$\rightarrow a$	b	c
0	(c, x)	(b, y)	(b, y)
1	(b, z)	(a, x)	(a, y)

Input:	0	1	0	0	1	1
Transition:	c	a	c	b	a	b
Output:	x	y	x	y	x	z

Cryptosystem based on Mealy automaton

Example: Cryptosystem based on Mealy automaton using 2 bits long blocks

$$\mathcal{M} = (\{a, b\}, \{0, 1, 2, 3\}, \{0, 1, 2, 3\}, a, \delta, \mu)$$

δ, μ	$\rightarrow a$	b
0	$(a, 1)$	$(b, 2)$
1	$(a, 3)$	$(b, 3)$
2	$(b, 0)$	$(a, 1)$
3	$(b, 2)$	$(a, 0)$

Input:	3	1	1	3
Transition:	b	b	b	a
Output:	2	3	3	0

δ', μ'	$\rightarrow a$	b
0	$(b, 2)$	$(a, 3)$
1	$(a, 0)$	$(a, 2)$
2	$(b, 3)$	$(b, 0)$
3	$(a, 1)$	$(b, 1)$

Input:	2	3	3	0
Transition:	b	b	b	a
Output:	3	1	1	3

Attack on cryptosystem based on Mealy automaton

Attack:

It is well known that Mealy automata mappings have the following two properties:

- They are length preserving. (For any input string ts length is the same as the length of the output.)
- They are prefix keeping. (The image of the prefix of the input string will also be the prefix of the output string. More formally: for every input string $w, v \in T^*$ the output for the string wv will start with the output string assigned to w.)

We can use choosen plaintext attack or choosen ciphertext attack with these two properties, to break the system:

- we can easily find the key automaton, or
- we can find an automaton which is equivalent to the key automaton.

Celluar automaton

Definition

A one dimensional cellular automaton consists of the following:

- Q – *finite set of states,*
- $(q_1, q_2, \ldots, q_m)$ – *initial state vector,* $q_1, ..., q_m \in Q$,
- f – *transition function* $Q^m \to Q^m$.
- *Let* $(q_1^1, q_2^1, \ldots, q_m^1)=(q_1, q_2, \ldots, q_m)$, *and* $(q_1^{t+1}, q_2^{t+1}, \ldots, q_m^{t+1}) = f(q_1^t, q_2^t, \ldots, q_m^t)$ *for each* $t \geq 1$.

Cryptosystems based on cellular automata

Basic idea:

We can use cellular automata as a cryptosystem in the following way:

- the transition function is the key,
- the plaintext block is the initial state vector,
- the cypertext block is generated by the cellular automaton.

Symmetric cryptosystem based on cellular automaton

Example: In this exaplpe we use 1 bit long cells, – $Q = \{0, 1\}$ – so we use each calculation in modulo 2 class. We encrypt 5 btis long blocks.
Transition function:

$$f(q_1, q_2, q_3, q_4, q_5) = (q_2, q_3, q_1, q_5 + q_1 q_2, q_4 + q_2 q_3).$$

Input:	1	0	0	1	1
Output:	0	0	1	1	1

$$f^{-1}(q_1, q_2, q_3, q_4, q_5) = (q_3, q_1, q_2, q_5 - q_1 q_2, q_4 - q_2 q_3).$$

Input:	0	0	1	1	1
Output:	1	0	0	1	1

Public key cryptosystem based on cellular automaton

Example:

$$f_1(q_1, q_2, q_3, q_4, q_5) = (q_2, q_3, q_1, q_5 + q_1q_2, q_4 + q_2q_3),$$

$$f_2(q_1, q_2, q_3, q_4, q_5) = (q_4, q_5, q_1 + q_4q_5, q_2 + q_1q_4, q_3 + q_1),$$

$$f = f_2(f_1(q_1, q_2, q_3, q_4, q_5)) =$$

$$(q_1q_2 + q_5, q_2q_3 + q_4, q_1q_2q_3 + q_1q_2q_4 + q_2q_3q_5 + q_4q_5 + q_2,$$

$$q_2q_1 + q_2q_5 + q_3, q_1 + q_2).$$

$$f^{-1}(q_1, q_2, q_3, q_4, q_5) = f_1^{-1}(f_2^{-1}(q_1, q_2, q_3, q_4, q_5)).$$

Cryptosystems based on cellular automata

Problems:

We have two main problems using cryptosystems based on cellular automata:

- the cellular automaton must be reversible,
- the cryptosystem must be secure.

There is no known universal method for key generation.
There are dedicated attacks for most cryptosystems based on cellular automata.

Dömösi cryptosystem (2008)

Definition

A modified finite automaton $\mathcal{A} = (Q, T, q_0, \delta, K, F)$ *used in the Dömösi cipher is a 6-tuple, consisting of the following:*

- Q – *finite set of states,*
- T – *input alphabet,*
- q_0 – *initial state,* $q_0 \in Q$,
- δ – *transition function:* $Q \times T \to Q$,
- F - *set of code states,* $F \subseteq Q$,
- K - *code alphabet (there is a one-to-one mapping between the code states and the letters of the code alphabet).*

Dömösi cryptosystem (2008)

Example:

$$\mathcal{A} = (\{a, b, c\}, \{0, 1\}, \delta, a, \{a, b\}, \{0, 1\})$$

	0	1	
δ	a	b	c
0	c	a	b
1	b	c	a

plaintext: "0011"
ciphertext: "0101110"

		0		0	1		1
a	c	a	c	a	b	c	b
0	1	0	1	1	1	0	

Dömösi cryptosystem (2008)

Advantages:

- There is no known attack on this cipher.
- Encryption and decryption are fast.
- Key generation is easy.

Disadvantage:

- The ciphertext is multiple times longer than the plaintext.

Finite automaton

Definition

A deterministic finite automaton $\mathcal{A} = (Q, T, \delta)$ is a triple, consisting of the following:

- *Q – finite set of states,*
- *T – input alphabet,*
- *δ – transition function: $Q \times T \to Q$.*

Finite automaton

Example:

$$\mathcal{A} = (\{a, b, c\}, \{1, 2, 3\}, \delta)$$

δ	a	b	c
1	c	b	a
2	a	c	b
3	b	a	c

Ecryption with finite automaton

Example:

$$\mathcal{A} = (\{a, b, c\}, \{1, 2, 3\}, \delta)$$

δ	a	b	c
1	c	b	a
2	a	c	b
3	b	a	c

- the set of states is the plaintext and ciphertext alphabet $\{a, b, c\}$
- input letters are pseudorandom numbers $\{1, 2, 3\}$
- each row is a permutation of the states
- the key is the transition matrix

plaintext:	a	b	b	a	b	a	b	a
pseudorandom numbers:	1	2	1	2	3	1	3	3
ciphertext:	c	c	b	a	a	c	a	b

Gluškov product

Given an automaton $A = (Q, T, \delta)$, let Q be written in a form $Q = Q_1 \times Q_2 \times \ldots \times Q_n$ for some $|Q_i| \geq 1$ and $n \geq 1$.
Then A can be written as a composition of n finite automata:

$$A = A_1 \times A_2 \times \ldots \times A_n(T, (\varphi_1, \varphi_2, \ldots, \varphi_n)).$$

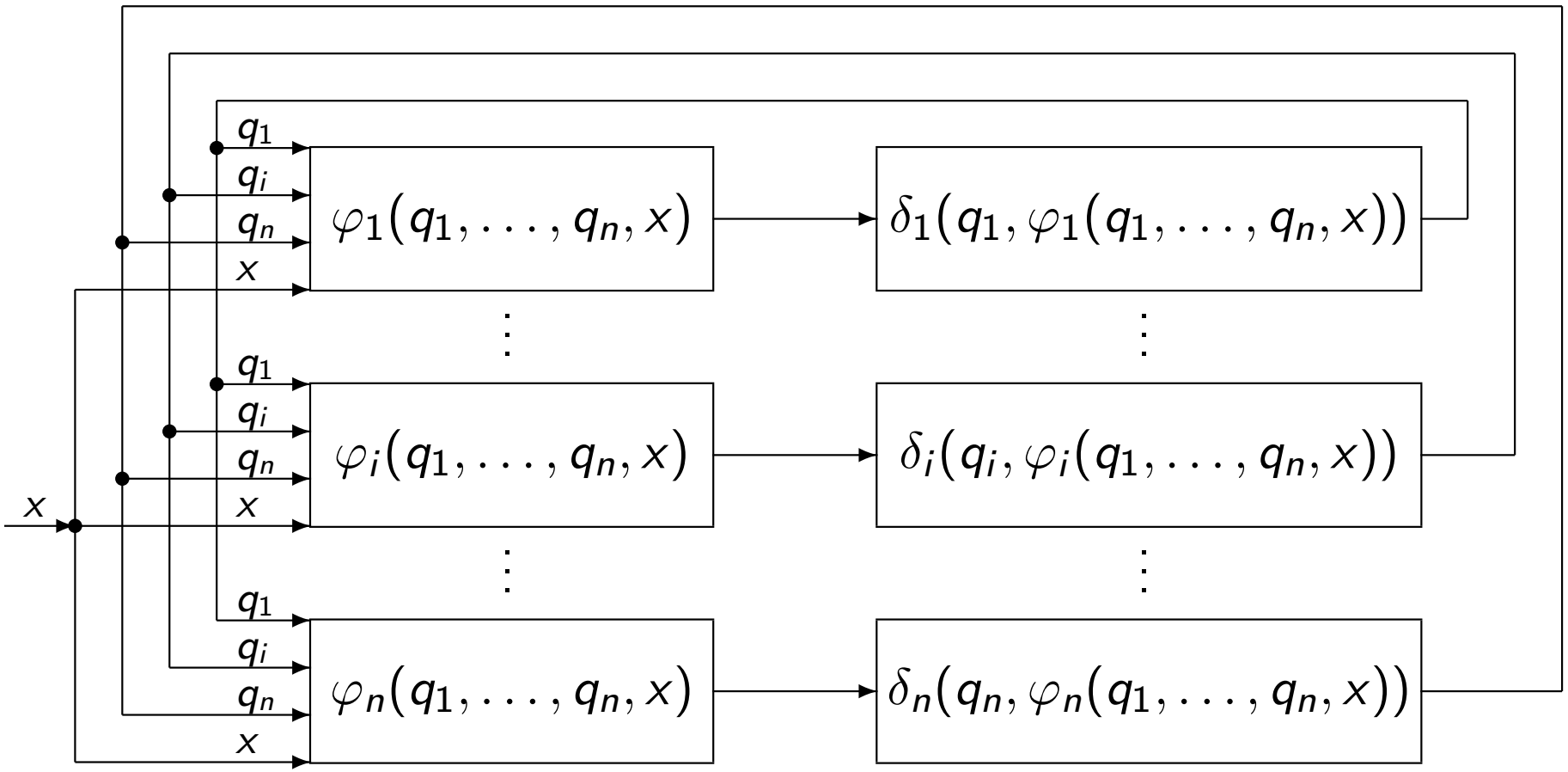
The component automaton A_i has a form $A_i = (Q_i, T_i, \delta_i)$, and we use the feedback function φ_i to calculate the actual element of T_i.
$\varphi_i : \ Q_1 \times Q_2 \times \ldots \times Q_n \times T \to T_i$.
$\delta_i : \ Q_i \times T_i \to Q_i$.
Finally,

$$\delta((q_1, q_2, \ldots, q_n), x) = (\delta_1(q_1, \varphi_1(q_1, \ldots, q_n, x)), \ldots$$

$$\ldots, \delta_n(q_n, \varphi_n(q_1, \ldots, q_n, x))),$$

where $q_i \in Q_i$ és $x \in T$.

Gluškov product



Example

Let the finite automaton $A_1 = (\{a, b\}, \{0, 1\}, \delta_1)$ with transition function:

δ_1	a	b
0	b	a
1	a	b

Let the finite automata network
$A = A_1 \times A_1 \times A_1(\{0, 1\}(\varphi_1, \varphi_1, \varphi_1))$, where
$\varphi_1(a, a, a, 0) = \varphi_1(a, a, b, 0) = \ldots = \varphi_1(b, b, b, 0) = 0$
$\varphi_1(a, a, a, 1) = \varphi_1(a, a, b, 1) = \ldots = \varphi_1(b, b, b, 1) = 1.$
In this case the transition function of A has the following form:

δ	aaa	aab	aba	abb	baa	bab	bba	bbb
0	bbb	bba	bab	baa	abb	aba	aab	aaa
1	aaa	aab	aba	abb	baa	bab	bba	bbb

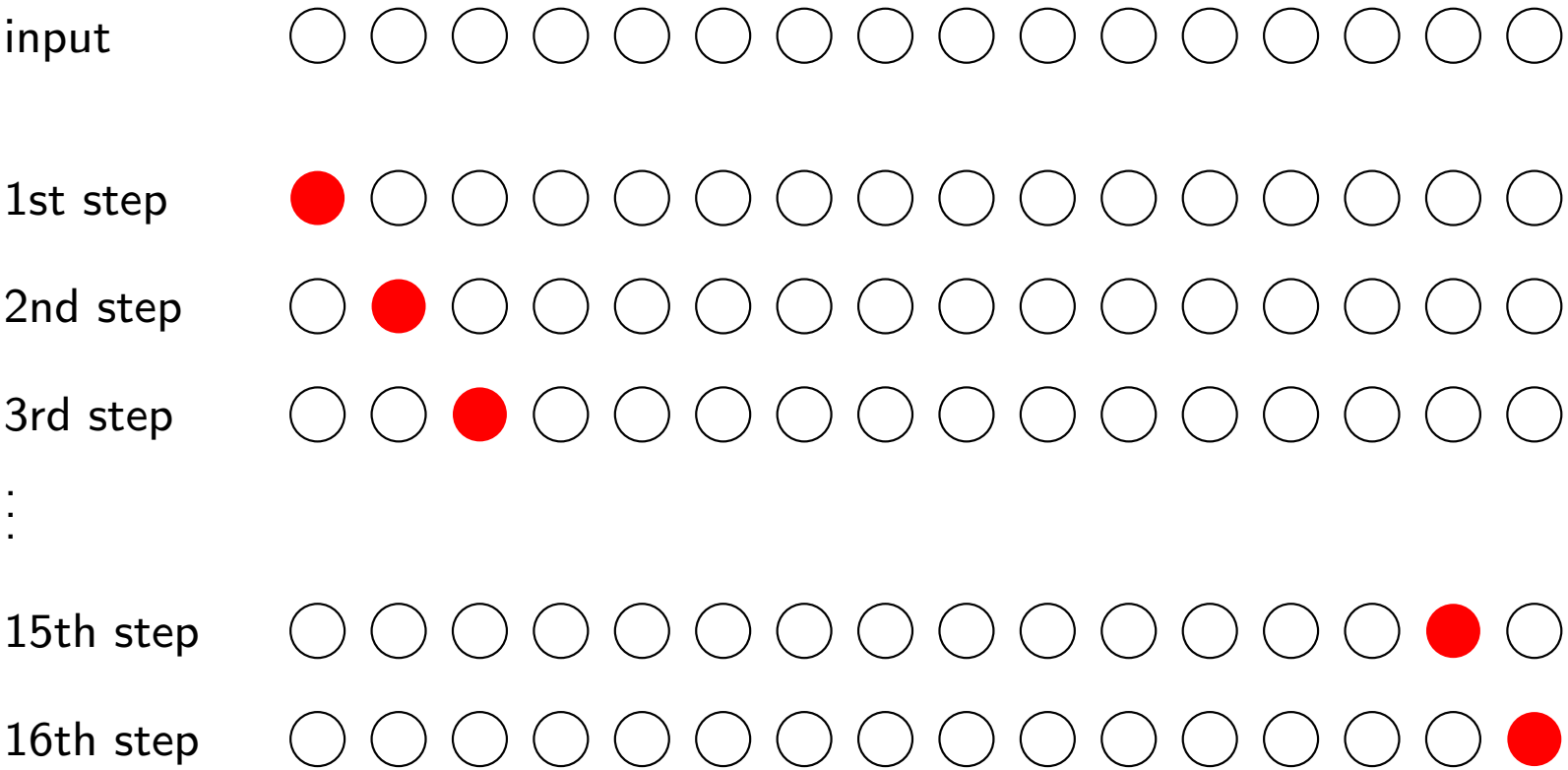
Cryptosystems based on Gluškov product (2013)

- The most simple versions of our block ciphers use 128 bit (16 bytes) long plaintext and ciphertext blocks.
- The automata network has 16 component automata.
- We use the same automaton for each component in the automata network.
- The component automata have
  - 1 byte long $(0 \ldots 255)$ states,
  - 2 bytes long $(0 \ldots 65535)$ input symbols,
  - and the transition matrix has 65536 lines, 256 columns, and $65536 \times 256 = 16777216$ elements. Each element is 1 byte long, this means the size of the transition matrix is 16 megabytes.

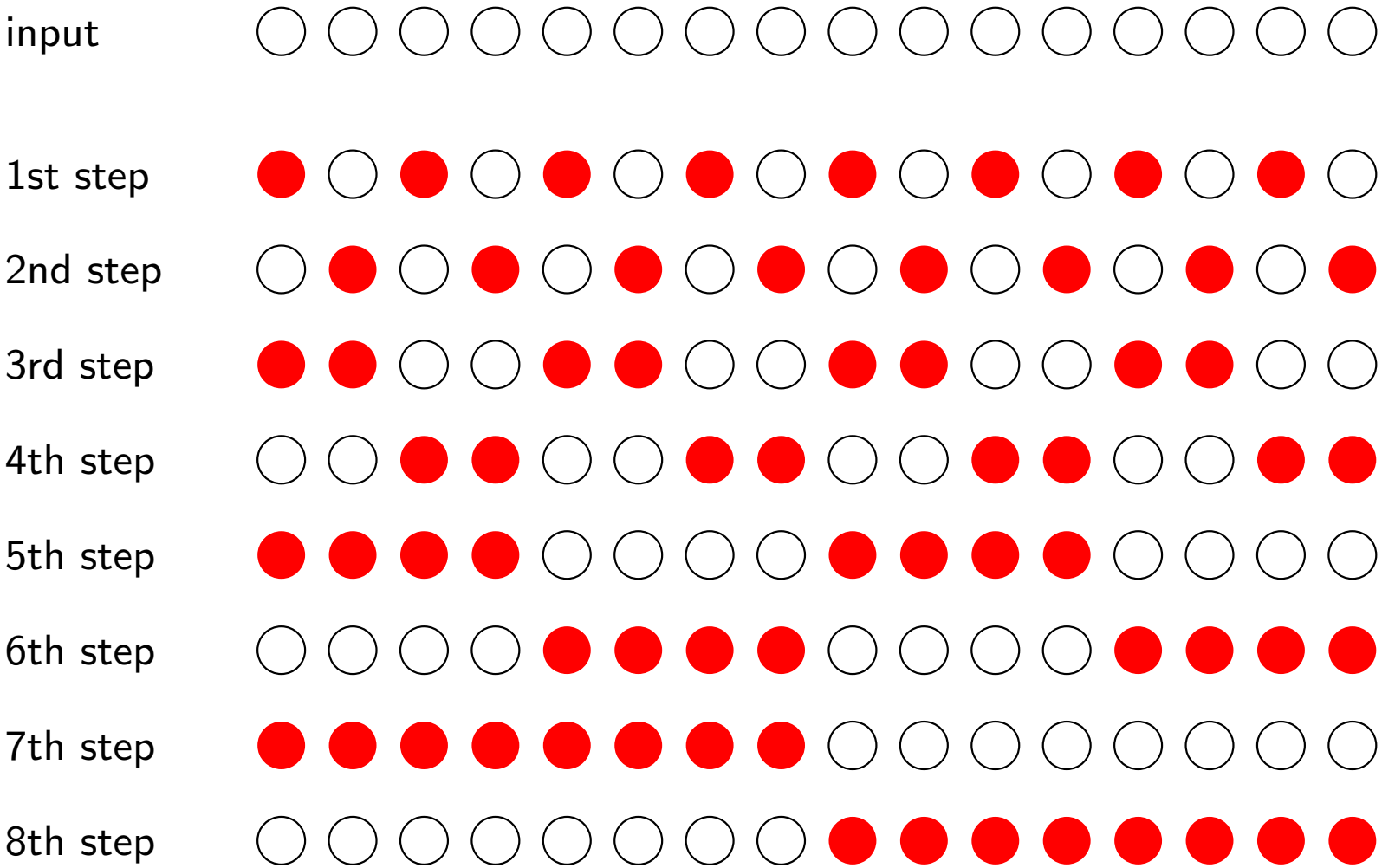
Encription – basic concept

1. Set up the states of the component automata with the plaintext block.
2. Calculate the 2 bytes long input for each component automaton in the following way:
   - the second byte of each input symbol is given by a pseudorandom generator,
   - the first byte of the input symbol is calculated from a neighbour state and the pseudorandom number of the same neighbour with the exclusive or (XOR) bit operation.
3. Run the automata network.
4. The letters of the ciphertext block are the transformed states of the component automata.

Sequential automata network



Two-phase automata network



Finite automaton

Definition

A deterministic finite automaton $\mathcal{A} = (Q, X, \delta)$ is a triple, consisting of the following:

- *Q – finite set of states,*
- *X – input alphabet,*
- *δ – transition function: $Q \times X \to Q$.*

Finite automaton

Example:

$$\mathcal{A} = (\{a, b, c\}, \{1, 2, 3\}, \delta)$$

δ	a	b	c
1	c	b	a
2	a	c	b
3	b	a	c

Ecryption with finite automaton

Example:

$$\mathcal{A} = (\{a, b, c\}, \{1, 2, 3\}, \delta)$$

δ	a	b	c
1	c	b	a
2	a	c	b
3	b	a	c

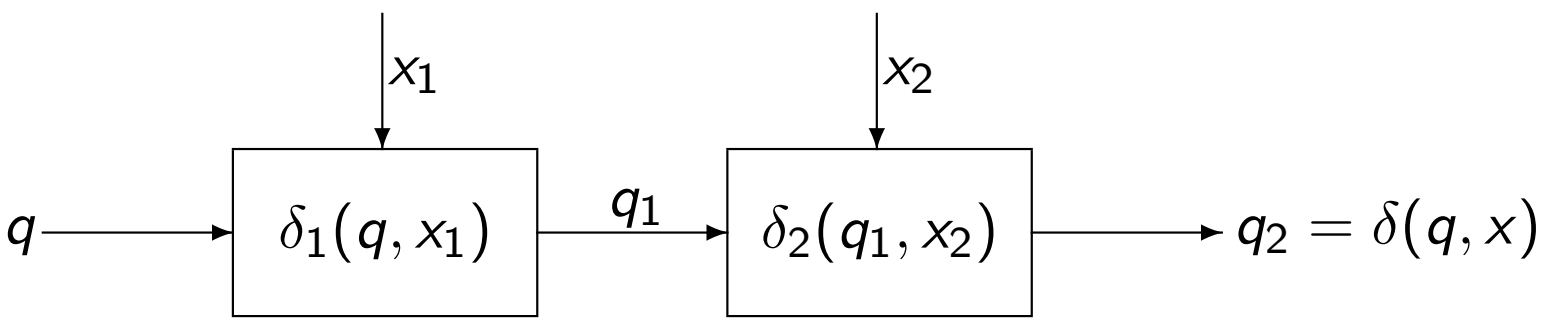
- the set of states is the plaintext and ciphertext alphabet $\{a, b, c\}$
- input letters are pseudorandom numbers $\{1, 2, 3\}$
- each row is a permutation of the states
- the key is the transition matrix

plaintext:	a	b	b	a	b	a	b	a
pseudorandom numbers:	1	2	1	2	3	1	3	3
ciphertext:	c	c	b	a	a	c	a	b

Temporal product

Definition

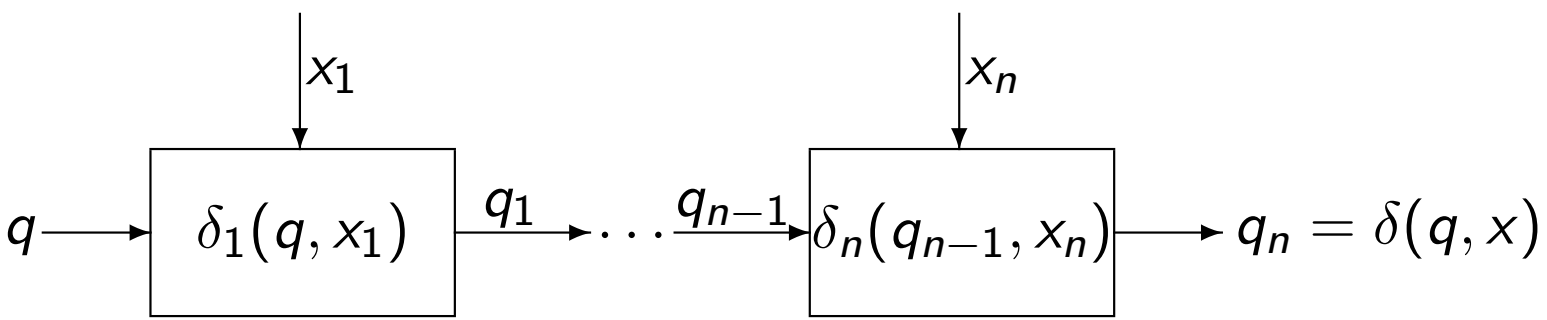
- *Let* $\mathcal{A}_i = (Q_i, X_i, \delta_i)$, $i = 1, 2$, *be automata having the common state set* $Q(= Q_1 = Q_2)$.
- *Take a finite non void set* X *and a mapping* ϕ *of* X *into* (X_1, X_2).
- *Then the automaton* $\mathcal{A} = (Q, X, \delta)$ *is a temporal product (t-product) of* $\mathcal{A}_1$ *by* $\mathcal{A}_2$ *with respect to* X *and* ϕ *if for any* $q \in Q$ *and* $x \in X$ $\delta(q, x) = \delta_2(\delta_1(q, x_1), x_2)$, *where* $(x_1, x_2) = \phi(x)$.



Generalized temporal product

Definition

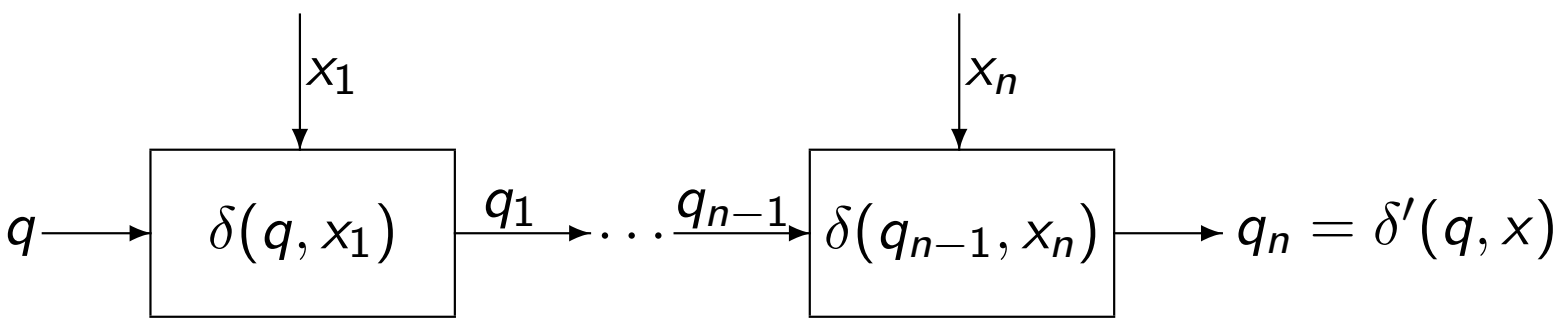
- *Let* $\mathcal{A}_i = (Q_i, X_i, \delta_i)$, $i = 1, \ldots, n$, $n > 0$ *be automata having the common state set* $Q (= Q_1 = \ldots = Q_n)$.
- *Take a finite non void set* X *and a mapping* $\phi : X \to \prod_{i=1}^{n} X_i$.
- *Then the automaton* $\mathcal{A} = (Q, X, \delta)$ *is a temporal product (t-product) of* $\mathcal{A}_1, \ldots, \mathcal{A}_n$ *with respect to* X *and* ϕ *if for any* $q \in Q$ *and* $x \in X$ $\delta(q, x) = \delta_n(\ldots \delta_2(\delta_1(q, x_1), x_2), \ldots, x_n)$, *where* $(x_1, \ldots, x_n) = \phi(x)$.



Our temporal product

Definition

- *Let* $\mathcal{A}_i = (Q_i, X_i, \delta_i)$, $i = 1, \ldots, n$, $n > 0$ *be automata having the common state set* $Q(= Q_1 = \ldots = Q_n)$, *common input alphabet* $X_1 = \ldots = X_n$, *and common transition function* $\delta(= \delta_1 = \ldots = \delta_n)$.
- *Take a finite non void set* X *and a mapping* $\phi : X \to \prod_{i=1}^{n} X_i$.
- *Then the automaton* $\mathcal{A} = (Q, X, \delta')$ *is a temporal product (t-product) of* $\mathcal{A}_1, \ldots, \mathcal{A}_n$ *with respect to* X *and* ϕ *if for any* $q \in Q$ *and* $x \in X$ $\delta'(q, x) = \delta(\ldots\delta(\delta(q, x_1), x_2), \ldots, x_n)$, *where* $(x_1, \ldots, x_n) = \phi(x)$.



Example

Example:

$$\mathcal{A} = (Q, X, \delta),\ \mathcal{A} = (Q, X, \delta)^{-1},\ Q = X = \{0, 1, 2, 3\}$$

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

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

- the set of states is the plaintext and ciphertext alphabet $\{0, 1, 2, 3\}$
- input letters are pseudorandom strings over $\{0, 1, 2, 3\}$
- each row is a permutation of the states (latin square recommended)

plaintext:	0	1	2	3
pseudorandom strings:	11	21	30	31
ciphertext:	1	0	3	0

Key automaton

- Consider an automaton $\mathcal{A} = (Q, X, \delta)$ with $Q = X$, where for every $a, b \in Q$ $(a \neq b)$ and $x, y \in X$ $(x \neq y)$, $\delta(a, x) \neq \delta(b, x)$ 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 $\mathcal{A}^{-1} = (Q, X, \delta^{-1})$ be the automaton for which $\delta^{-1}(b, x) = a$ with $a, b \in Q,\ x \in X$ 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

1. Let $p_1 \dots p_k$ be the plaintext, where $p_1, \dots, p_k \in Q$.
2. Let $r_1, \dots, r_k \in X^+$ be random strings generated by the pseudorandom number generator starting by a seed r_0. (We note that $|r_0|, \dots, |r_k| = n$ holds for a fixed positive integer n.)
3. The ciphertext will be $c_1 \cdots c_k$ with $c_1 = \overrightarrow{\delta(p_1, r_1)}, \dots, c_k = \overrightarrow{\delta(p_k, r_k)}$.

Decryption

1. Let $c_1 \dots c_k$ be the ciphertext, where $c_1, \dots, c_k \in Q$.
2. Let $r_1, \dots, r_k \in X^+$ be the same random strings generated by the pseudorandom number generator starting by a seed r_0.
3. The decrypted plaintext will be $p_1 \cdots p_k$ with $p_1 = \overrightarrow{\delta^{-1}(c_1, (r_1)^R)}, \dots p_k = \overrightarrow{\delta^{-1}(c_k, (r_k)^R)}$, where $(r_i)^R$ is the reverse of r_i.

Summary

1. The ciphertext of the new stream cipher has the same length as the plaintext. (Dömösi)
2. Its security does not depend on unproved mathematical conjectures. (RSA)
3. It does not contain possible backdoors, like S-boxes. (DES, 3DES, AES)
4. The structure of the new cipher very simple.
   - It is easy to understand.
   - It uses only one operation.
   - It is very fast.
   - It requires very small and simple code.
   - It is easy to implement hardware solutions.
5. It is easy to create block cipher based on this stream cipher.

Plans

1. NIST test.
2. Differential cryptanalysis.
3. Post-quantum solution.


Guan, P. : Cellular Automaton Public-Key Cryptosystem. Complex Systems 1 (1987), 51-56.


Dömösi, P. : A novel cryptosystem based on finite automata without outputs. Proceedings of AFLAS 2008, Worlds Scientific, 2010, 23-32.


Dömösi, P. Nehaniv, C. : Algebraic theory of automata networks: An introduction. SIAM monographs on Discrete Mathematics and Applications, Volume 11, Philadelphia, PA, 2005.


Dömösi, P., Horváth, G. : A Novel Cryptosystem Based on Abstract Automata and Latin Cubes. Studia Scientiarum Mathematicarum Hungarica, Volume 52, Issue 2, (2015), 221-232.


Dömösi, P., Horváth, G. : A Novel Cryptosystem Based on Gluškov Product of Automata. Acta Cybernetica, Volume 22, (2015), 359-371.