Models of Pseudonondeterministic Cryptographic Transformations

Yurii Baryshev
Department of Information Protection
Vinnytsia National Technical University
Vinnytsia, Ukraine
yuriy.baryshev@gmail.com

***Abstract* — The analyses of possible approaches of cryptographic transformation performance from the automata theory point of view are presented. The choice of pseudonondeterministic approach for cryptographic transformations implementation is grounded. The models of encryption and hashing are presented as instance of the approach implementation.**

***Анотація* — Представлено аналіз можливих підходів до реалізації криптографічних примітивів з точки зору теорії автоматів. Обгрунтовано вибір псевдонедетермінованого підходу для реалізації криптографічних перетворень. Представлено моделі шифрування та гешування як приклад використання даного підходу.**

Keywords — automaton; model; ciphering; hashing; pseudonondeterministic.

Ключові слова — автомат; модель; шифрування; гешування; псевдонедетермінований.

I. Introduction

Modern approaches of cryptographic transformations development are based on the conception of their openness for the external research of these transformations. This peculiarity provides an opportunity for the scientific society to analyze developed algorithms. The latter allows correctness verification and boosts cryptography development due to active thoughts sharing thus allowing researches to avoid mistakes of others. Moreover, this feature allows customers of cryptographic tools to receive arbitrary expert views concerning quality and comparison analytics of the assets, they are going to purchase [1, 2].

Despite these positive features the approach grants similar abilities to intruders – they also are able to study cryptographic transformations and getting information concerning algorithms implementation drawbacks. For instance, such openness of hashing algorithms causes additional vulnerability embedding, which grants to intruders the ability to attack parallel hash functions such as cascaded ones and prepare attacks beforehand [3]. Hence intruders are able to start final and intermediate hash values yielding before they received by the authorized users from the message, that even could be unknown or not existing at the moment, when intruder begins attacking [3, 4]. These are critical for any hash functions but it is crucial for unkeyed hash functions, which are used for messages integrity checking, quick information searching and digital signatures yielding. Thus openness of cryptographic transformations provides vulnerabilities along with a set of positive features.

Same considerations are to be correct for ciphers designing. The openness of encryption algorithms allows intruders to perform their differential cryptanalysis gaining information concerning used key, which lowers number of sets are to be process to reverse the encryption [5]. Thus the initial difficulty of brute force attack (in case of the ideal cipher model) is reduced. The known approach of encrypting based on the well-studied mathematical problems such as discrete logarithm computation or large integers factorisation could make useless to perform such kind of the attack for the intruder [1]. But they are difficult to compute using modern computational platforms due to used operation and despite they provide certain level of protection which is theoretically proved, the one is much lesser than ideal computational infeasibility level.

That's why it is important to combine openness of algorithms for the external research and security of closed algorithms for an intruder analyses.

The goal of this research is infeasibility increasing of the cryptography transformations without losing their openness.

The following tasks are to be solved to reach the goal:

- cryptography transformations models analyses;
- development of the pseudonondeterministic cryptography transformations models;
- implementation of the models.

II. Automaton Models of Known Cryptographic Transformations

It is known, that each method could be described by both an algorithm and an automaton performing this algorithm. Therefore at this research it is proposed to use automaton models for cryptographic transformation formalization.

Consider these transformations from the intruder's point of view. The transformation is described by the following determined automaton [6]:

$$ADC = (OT, CT, k, IS, f(\cdot)), \quad (1)$$

where *ADC* – an deterministic automaton, which implements cryptographic algorithm; *OT* – all possible open texts; *CT* – all possible ciphertext blocks; *k* – key data; *IS* – intermediate (or internal) states $IS \subseteq CT$; $f(\cdot)$ – a function, which formalize known to an intruder cryptographic transformations .

Consider encryption data transformation [1]:

$$e_i = \varepsilon_k(m_i), \quad (2)$$

where e_i – a ciphertext block obtained at the *i*th iteration; m_i – a message (plaintext) block; $\varepsilon_k(\cdot)$ – an encryption, that uses key *k*.

Therefore the automaton *DE*, which implements deterministic encryption, could be presented as one performed by the following automaton:

$$DE = (M, CT, k, IS, \varepsilon_k(\cdot)) . \quad (3)$$

According to the notion (3) the alphabet of the automaton *DE* is presented as a set of all possible data blocks $m_i \in M$. Therefore in case, when the data block consists of *n* bits, the power of the alphabet is 2^n . Consequently for any given *M* the following inequality is correct: $\|M\| \le 2^n$.

The notable feature of the model (3) is that it describes both stream and block ciphers, but in the case of the former ones $n = 1$. The role of automaton states is performed by ciphertext blocks *CT*. That's why *CT* in most cases could coincide with *IS* for block ciphers.

Therefore the function $\varepsilon_k(\cdot)$ implements the following mapping $M \times CT \to CT$. The variety of known ciphers causes that particular cipher may need latter assertion modifications for the automaton, that implements the cryptographic transformation, but nonetheless from the intruder's point of view the function $\varepsilon_k(\cdot)$ mapping remains at abovementioned form.

The hashing computation iteration model is called a hash construction. The classical one is Merkle-Damgaard construction [1, 7, 8]:

$$h_i = f(h_{i-1}, m_i) , \quad (4)$$

where h_i – the *i*th intermediate hash value, $i \in \{1, 2, \cdots, l\}$; m_i – the *i*th message block; $f(\cdot)$ – an irreversible reduction function.

According to the equation (4), the cryptography hashing transformation is described by an automaton model in the similar way as the ciphering one:

$$DH = (M, H, h_0, h_l, f(\cdot)) \quad (5)$$

where *DH* – an automaton, which implements deterministic hashing; *H* – a set of all possible intermediate hash values; h_0 – initialization vector in case of unkeyed hashing or key in case of the keyed hashing; h_l – message's hash value.

The alphabet of the *DH* is the set of all possible data blocks $m_i \in M$. The states of the automaton are presented by intermediate hash values. Due to fixed length of hash value for the certain message there is only one allowed state of the automaton, at which it stops, and this is final hash value of whole message h_l .

The drawback of both ciphering and hashing, which could be described for the intruder as models (3) and (5), is their predictability. The latter allows to design attacks based on the differential cryptanalysis and obtain information concerning the value of the used key certain bits [5]. Thus the preparation to the attack could be started even before the data is cryptographically transformed.

It is obvious, that this point can be avoided by nondeterministic cryptographic transformations, which are based on the nondeterministic automaton model:

$$ANDC = (OT, CT, k, IS, F) , \quad (6)$$

where *ANDC* – an automaton, which implements a nondeterministic cryptographic transformation; *F* – an unknown to an intruder cryptographic transformations.

The uncertainty of the performed action forces intruder to perform additional picking out while cryptanalitical attack designing, which obviously increases infeasibility of analyzed cryptographic algorithms. The implementation of the latter is impossible due to practical issues of nondeterministic automaton programming and repetition constraint of cryptographic transformations. That's why the pseudonondeterministic approach is proposed.

III. Automaton Models of Pseudonondeterministic Cryptographic Transformations

The key feature of pseudonondeterministic approach is hiding from the intruder round transformations. At the same time a cryptographic transformation algorithm stays open for the external research by community. This is achieved by making cryptographic algorithm as such performed by the deterministic automaton (1) for the person, who knows the key, and as transformation performed by the nondeterministic automaton (6) for the person, who doesn't know this key. To achieve it the following algebraic structure is proposed to formalize a subject

performing pseudonondeterministic ciphering algorithm:

$$APNDC = (OT, CT, k, IS, V, F_v), \qquad (7)$$

where *APNDC* – a subject (kind of automatons), which implements a pseudonondeterministic cryptographic transformation; V – a set of control vectors (unknown to an intruder); F_v – a set of known to an intruder cryptographic functions $f_{v_i}(\cdot) \in F_v$, whereby $v_i \in V$.

The approach used for model (7) obtaining is to be implemented for encryption and hashing. Therefore the automaton *PNDE*, which performs pseudonondeterministic encryption is formalized by the following set of six:

$$PNDE = (M, CT, k, IS, V, E), \qquad (1)$$

where E is a set of encrypting functions $\varepsilon_{v_i k}(\cdot) \in E$.

The pseudonondeterministic hashing is described by the following set of six:

$$PNDH = (M, H, h_0, h_l, V, F_v). \qquad (9)$$

The basic pseudonondeterministic hash construction is the following one [8]:

$$\begin{cases} h_i = f_{v_i}(h_{i-1}, m_i); \\ v_i = g(h_{i-2}) \end{cases} \qquad (10)$$

where $g(\cdot)$ is a function for hashing control vector generation.

Therefore implementation of the model (7) allowed to get the subjects (8) and (9), which perform pseudonondeterministic encryption and hashing respectively. Notably these subjects appear to an intruder as nondeterministic automaton (6), because he doesn't see the sixth parameter of the subjects. At the same time the subjects are open for their arbitrary effectiveness parameters estimation by community. Therefore this meets modern paradigm of cryptographic transformation.

IV. Conclusions

This research shows that model of the known approaches of cryptographic transformation designing can be considered as one performed by the deterministic automaton by the intruder. The approach causes additional vulnerabilities of developed transformations.

This drawback could be avoided by using nondeterministic automatons for cryptographic transformations performance, which is nearly impossible from the practical point of view. That is why the pseudonondeterministic approach was proposed. The approach makes subjects, those performs cryptographic transformations, look like nondeterministic automaton for the intruder, who doesn't know a key. At the same time the subjects are deterministic, so they can be implemented. Pseudonondeterministic encryption and hashing models were developed using the approach. According to the models the cryptographic transformation subjects looks like nondeterministic automaton for the intruder contrary to the known deterministic approach, which provides infeasibility only because the initial state of the automaton is unknown to the intruder.

Analysis of these models shows, that special vector generations functions are to be designed to implement these subjects. Therefore further research would be aimed to develop pseudonondeterministic cryptographic transformations methods as well software and hardware tools for their implementation.

References

[1] B. Schneier. *"Applied Cryptography: Protocols, Algorithms, and Source Code in C"*, John Wiley & Sons, Inc, 1996 p. 784.

[2] Лужецький В. А. Основи інформаційної безпеки : навчальний посібник / В. А. Лужецький, А. Д. Кожухівський, О. П. Войтович. – Вінниця: ВНТУ, 2013. – 221 с.

[3] J. J.Hoch, A. Shamir *"Breaking the ICE – Finding Multicollisions in Iterative Concatenated and Expanded (ICE) Hash Functions"*, 2006, p. 13. http://www.wisdom.weizmann.ac.il/~yaakovh/papers/hashpaper_submission.pdf

[4] J.Kelsey, T. Kohno. *"Herding hash functions and the Nostradamus attack"*, 2005, p. 18. http://archives.scovetta.com/pub/crypto/Nostradamus%20Attack.pdf

[5] C. Blondeau, G. Leander, K. Nyberg. Differential-Linear Cryptanalysis Revisited, 2014, p. 20, http://users.ics.aalto.fi/~blondeau/PDF/ FSE2014.pdf

[6] J. A. Anderson *"Discrete mathematics with combinatorics"*, Prentice Hall, Upper Saddle River, New Jersey, 2004, p. 960..

[7] P. Gauravaram, S. Hirose, D. Stebila *"Security Analysis of a Design Variant of Randomized Hashing"*, Proceedings ATIS 2017: Auckland, New Zealand, 2017, p. 14-22.

[8] V. Luzhetskyi, Y. Baryshev. "The Generalized Construction of pseudonondeterministic hashing", Computing, 2012, Vol. 11, Issue 3, p. 302-308.