**UDC 003.26 : 004.056.53**

# AUTHENTICATION SCHEME ON FRACTAL SETS

***Denis Samoilenko***

*Implementation of authentication tools is the requirement for secured information system in open networks. Known authentication protocols are based on some secret knowledge (key or password) checking. Preferable scheme for checking procedure is client-server dialog without transferring of secret knowledge, even in ciphered form, also known as zero-knowledge protocol. Such protocols, as a rule, use mathematically complex problems without known simple solution for inverse calculation. This obscurity case a weakness of protocols – discovering solution of the problem impairs the secrecy of protocol. It is proposed the authentication schemes built on complex fractal sets including distant knowledge diagnostic methods. Fractal sets suitable for posed problem due to properties of finiteness and infiniteness combination. Property of finiteness allows set construction; infiniteness ensures multiple usage of the scheme. The algorithm of client-server communication is shown. Usage of authentication scheme could improve the network information resource security.*

***Keywords:*** *authentication, zero-knowledge protocol, fractal, fractal set, web-security.*

**Introduction**. As a rule, network information resources (NIR) provide different functionality for different users. To access additional services user should be authorized in NIR by authenticity verification. Such verification means checking user's secret knowledge – a key, password etc. If NIR contains information with restricted access the authentication function should be provided obligatorily.

In secure authentication schemes different communication protocols are used. One of the most perspective protocols is zero-knowledge protocol (ZNP) [1].

In such protocols in the communication channel transmits no information about the secret knowledge even in ciphered form. So, attacker has no possibilities for secret knowledge restoration by analysis of any number of transmitted packs.

The most popular zero-knowledge protocols use mathematically complex problems without known simple solution for inverse calculation. Complexity are usually derived from problem of graph isomorphism (Hamiltonian cycle in large graphs) or discrete log problem (in a given group). From the other side, the obscurity case a weakness of protocols – discovering solution of the problem impairs the secrecy of protocol.

Additional researches should be provided to find problems with granted complexity. In a present work it is proposed a protocol based on a problem of fractal sets comparison in a complex plane.

**Problem review.** In zero-knowledge protocols two main requirements should be combined. First one claims the possibility of several client-server communications (CSCs) during the session. In a perfect case amount of possible CSCs should be infinite.

The second requirement claims no possibility for attacker to restore a secret knowledge by analysis of any (finite) number of intercepted CSCs.

Thus objects that could form basis of ZNP should combine properties of finiteness and infiniteness. Finiteness allows segregation of finite secret knowledge for legal user. A property of infiniteness helps to solve a problem of protocol safety limit or increase the maximal number of CSCs with usage of the same secret knowledge.

One of the objects that combine aforementioned properties is a complex fractal set (FS). In work [2] FS is defined to be a set of complex points $X_0$ for which converges the iterative sequence

$$X_{k+1} = X_k^N + C, \qquad (1)$$

where $C$ – complex constant (consists of two real numbers – real and imaginary parts), $N$ – the power of sequence.

For the better presentation FS points could be placed on a Cartesian plane (means complex plane) forming fractal set image.

The idea of authentication with zero-knowledge is similar to the distant knowledge diagnostic problem. So in authentication schemes could be used some ideas adopted form this problem.

Proposed in work [3] method of distant knowledge diagnostics is focused on edge definition problem in semantic space. For the authentication problem the method could be easily changed with a procedure of full-space tracing defining the edge of intersected sets.

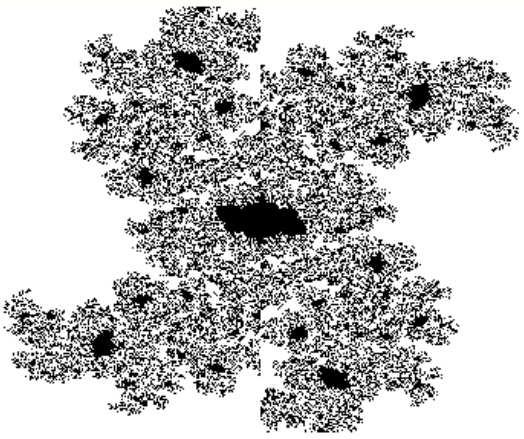
The knowledge (or knowledge field) in [3] is defined to be a part of semantic space with some distinct property. For the authentication problem it is enough that this part of space has had only Boolean property – belong or not belong to knowledge. Such situation is similar to credit/fail attestation form.

In work [4] it was proposed the usage of complex fractal images for authority protection on printed issues. It was described some fractal proper-

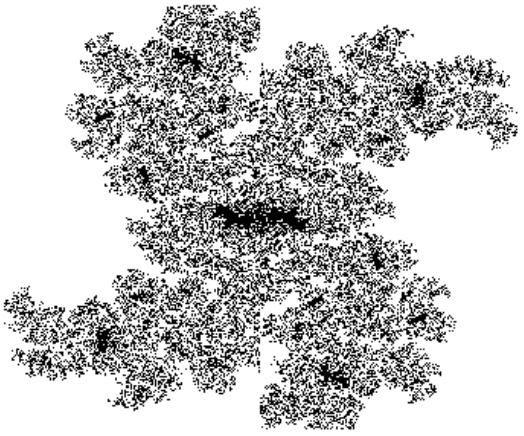
ties that could be useful in security improving. In combination with knowledge diagnostic methods such properties could form the basis of ZNP authentication scheme.

**Proposition.** The main idea of authentication consists in using knowledge diagnostic methods for respondent which knowledge field has a form of a FS. Server should check the "knowledge field" of client using dialog principle in form question (from server) – answer (from client).

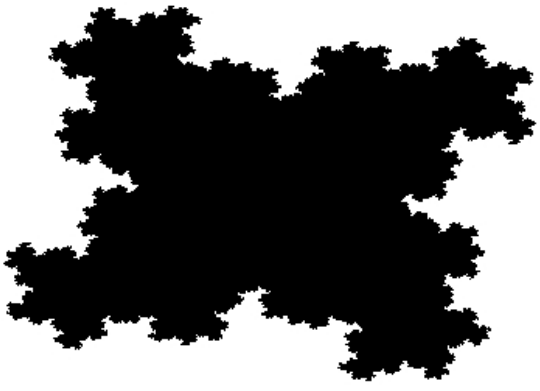
It is necessary to note that FS are extremely sensitive to the choice of parameters $C$ and $N$. On Fig. 1 FS images for the different values of $C$ and $N$ are shown.



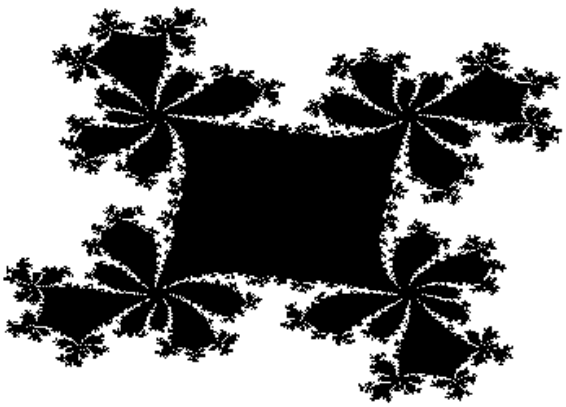
a)



b)



c)



d)

Fig. 1. Fractal set images with different parameters: a) $C$=(0.590; 0.420), $N$=3; b) $C$=(0.589; 0.420), $N$=3; c) $C$=(0.590; 0.420), $N$=4; d) $C$=(0.610; 0.450), $N$=4

The difference in 0.001 in only one $C$ component (in real part) produces changes that could be easy detected visually (Fig. 1 a-b). The difference in (0.02; 0.03) in $C$ with constant power index $N$ cause cardinal changes in shape and structure of fractal set image (Fig. 1 c-d).

The greater changes produce difference in power $N$. For the parameters $N$=3 and $N$=4 results are shown on Fig. 1 (a-c).For the value of $N$=2 FS image consists from some several points so this image omitted from the Fig. 1 as non-informative.

Described property of FS justifies it's usage in ZNP authentication schemes for NIR. The property of finiteness provides by finite number of parameters, that uniquely defines FS: $C$ and $N$ in formula (1). Infiniteness follows from infinite number of points on a complex plane. Moreover, infinite number of points contains any finite part of the plane, so limitation of comparison area does not impair the safety of protocol.

Parameters $C$ and $N$ could be mentioned as a secret knowledge – the client's numerical password which forms knowledge field in FS form. Additive constant $C$ consists from two real numbers. The power index $N$ could be real or integer. To select the types of constants it is necessary to analyze some aspects.

As far as different computers could use different calculation technique, it is recommended the usage of integer numbers for $N$, because there is no precise method to compute $X^N$ where $N$ is not integer. Approximate methods depend on math support (co-processor, math libraries, programming language, data types, zero limit etc.). In that case the same formula could give different results on different computers.

Components of additive constant $C$ could not be integer, because the region of (1) convergence is limited by inequality $|C|<\sqrt{2}$. In this region there are only 9 points with integer coordinates. Thus $C$ should forms from real number.

In case of integer $N$ all calculations with $C$ components require only basic algebraic operations: addition/subtraction and multiplication/division. Such operations have similar precision on different computers limited by data type used for number presentation. In special cases algebraic calculations could be produced with any precision, specified in protocol.

According to proposed limitations the password $P$ is consist from three numbers – integer number $N$ and two real numbers Re $C$ and Im $C$:

$$P=(n, \text{Re } C, \text{Im } C). \quad (2)$$

The adopted method of knowledge diagnostic defines a content of CSCs. A client is mentioned to be a student and server – to be a teacher. Teacher has knowledge (password $P$) and he should check the student's knowledge. Additional requirement (ZNP) states that communication (examination) should not discover any information about $P$ directly.

**The scheme.** Taking into account expressed ideas the following authentication scheme is proposed:

*Goal*: Server should authenticate client by checking the knowledge of password $P$ in form (2) using ZNP.

1. *Server*: generates random complex point $X=(\text{Re } X, \text{Im } X)$ from region $|X|<1$ or any subregion (if it is reasonable).

2. *Server*: calculates the sequence (1) with $X_0=X$ and checks its convergence. Constants $C$ and $n$ are defined from the secret password $P$.

3. *Server*: sends to client the point $X$.

4. *Client*: performs action 2.

5. *Client*: sends to server result of calculation – does sequence converge or not.

6. *Server*: checks client's answer. In a right case performs actions 1-6 until probability limit was exceeded. In a wrong case authentication falls.

Probability limit could be estimated by formula $p_{lim}=2^{-n}$, where $n$ – the number of action 1-6 repetitions. In the estimation it is considered that on repetition twice decreases probability of impersonation.

**Features.** The authentication scheme could be produced with some features.

It is rather obvious that for different clients should be selected different passwords. But different values for constants entail some peculiarities.

As follows from Fig. 1 fractal sets could be relatively solid or sparse. For sparse FS random point $X$ could be selected from full fractal space. But for solid FS it is better to choose the part of space near the fractal edge. In a case of full space tracing for solid FS attacker could observe a difference between central and peripheral points and use it for attacks. So in addition to password $P$ it is recommended to store information about area of point's selection concretely for this $P$.

A possibility of infinite $X$ point's number producing could be worsened if some points $X$ will be repeated in different CSCs (possibly in different sessions). To avoid such collisions it is recommended to use a great precision of $C$ components or/and to provide storage of used points for all clients and all sessions during lifetime of password.

Let's consider that complex type is defined to be a program structure (all program fragments shown in C/C++ formalism):

```
typedef struct tagCOMP
{       float Re; // the real part
        float Im; // the imaginary part
} complex;
```

Necessary mathematical operations on complex number are defined by functions:

```
float mod(complex &x)        //modulus of complex number
{       return sqrt(x.Re*x.Re+x.Im*x.Im); }

float arg(complex &x) //argument of complex number
{       if (x.Re==0) if (x.Im>0) return acos(0); else return -acos(0);
        return atan(x.Im/x.Re);
}
```

In such case the next element in sequence (1) could be computed by the function:

```
complex Next(complex &X, int N)
{       float x,y;                      // variables for intermediate calculations
        x = mod(X);                     // modulus of X
        for(int i=1;i<N;i++)
        x=x*mod(X);                     // power for modulus: |X^N|=|X|^N
        y = N*arg(X);                   // power for argument: Arg(X^N)=N·Arg(X).
                                        // Here x and y – modulus and argument of X^N
        complex temp;                   // temporal complex variable
        temp.Re = x*cos(y)+C.Re;        // computing the result of X^N+C
```

```
    temp.Im = x*sin(y)+C.Im;    // considering that value of C is defined above
    return temp;                // returning the result
}
```

Convergence of sequence could be checked by the expression:

```
int N; complex X;                   //variables for X and N data
X.Re = …; X.Im = …; N = …;          //value of start point (X0) and sequence power
int k=1;                            //counter
do{ X=Next(X, N); k++;}  while((mod(X)<100)&(k<100));
```

There are two conditions for calculation aborting: mod(X)<100 means that X value is not great enough for conclusion about convergence; k<100 – limits the sequence. After the given fragment a convergence of sequence is defined by checking value of mod(X).

Limit values for mod(X) and (k) were used for computing fractal set images shown on Fig. 1. For the optimization of calculation complexity these values could be decreased. It could be shown [2] that there are no convergence points for $|C| > \sqrt{2}$. So this value could be used for the break condition in optimized code. For the optimal value of sequence member limitation further researches are necessary.

**Conclusions.** Authentication scheme based on zero-knowledge protocol is proposed as a combination of knowledge diagnostic methods and fractal sets comparison in a complex plane. The scheme is built on dialog principle and could perform authorization with given probability limit.

The proposed scheme has some features which should be accounted to avoid possible problems in technical realizations. It is shown some recommendations and program fragments allowed realization of authentication scheme.

Further researches could be based on analysis of different fractal sets in multi-dimension spaces and its applicability for selected problems. Additional researches could be related with optimization of calculation complexity of the scheme.

## REFERENCES


[1]. QUISQUATER J-J, GUILLOU L.C. and BERSON T.A. (1989) How to Explain Zero-Knowledge Protocols to Your Children., *Advances in Cryptology*. – CRYPTO'89: Proceedings 435: p. 628-631.

[2]. MANDELBROT B. (2004) Fractals and Chaos. Berlin: Springer.

[3]. SAMOILENKO D. N. (2012) Knowledge diagnostics by search methods in the semantic space. *Electrotechnic and computer systems* 07 (83), p. 154-161

[4]. SAMOILENKO D., MIROSHNICHENKO O. and POPOV D. (2010) Fractal images use for holographic protection of polygraphist production. *Qualilogy of book* 2 (18), p. 77-81


## ЛИТЕРАТУРА


[1]. Quisquater, J-J. How to Explain Zero-Knowledge Protocols to Your Children [Текст]/Quisquater, J-J; Guillou, L.C.; Berson, T.A. // Advances in Cryptology. – CRYPTO'89: Proceedings 435: 628–631.

[2]. Mandelbrot B. Fractals and Chaos. [Текст] / Mandelbrot B. – Berlin: Springer. – 2004 – p. 38.

[3]. Samoilenko D. N. Knowledge diagnostics by search methods in the semantic space. [Текст] / Samoilenko D. N. // Electrotechnic and computer systems. – 2012. – № 07 (83). – p. 154-161.

[4]. Самойленко Д. М. Використання фрактальних зображень для голографічного захисту поліграфічної продукції [Текст] / Самойленко Д. М., Мірошниченко О. В. Попов Д. Д. // Квалілогія книги. – 2010. – № 2 (18). – с. 77-81.


## СХЕМА АУТЕНТИФИКАЦИИ НА ФРАКТАЛЬНЫХ МНОЖЕСТВАХ


Для построения защищенных информационных ресурсов в открытых сетях необходима реализация средств аутентификации. Известные протоколы аутентификации основываются на проверке некоторого секретного знания (ключа или пароля). Предпочтение отдается схемам проверки, построенным на диалоге клиента и сервера, осуществляемым без передачи секретного знания даже в зашифрованном виде – протоколам с нулевым разглашением. Как правило, такие протоколы используют математически сложные задачи с неизвестным решением обратной задачи. Эта неизвестность становиться причиной уязвимости протокола – нахождение решения задачи сведет на нет надежность протокола. Предложена схема аутентификации, построенная на комплексных фрактальных множествах используя в составе методы дистанционного оценивания знаний. Фрактальные множества применимы благодаря объединению свойств конечности и бесконечности. Свойство конечности позволяет построения самого множества, свойство бесконечности гарантирует возможность многократного использования схемы. Приведен алгоритм клиент-серверного взаимодействия. Применение схемы приведет к улучшению показателей безопасности сетевых информационных ресурсов.



**Ключевые слова**: аутентификация, протокол с нулевым разглашением, фрактал, фрактальное множество, сетевая безопасность.

## СХЕМА АВТЕНТИФІКАЦІЇ НА ФРАКТАЛЬНИХ МНОЖИНАХ

Для побудови захищених інформаційних ресурсів у відкритих мережах необхідна реалізація засобів автентифікації. Відомі протоколи автентифікації ґрунтуються на перевірці певних таємних знань (ключа чи пароля). Перевага надається схемам перевірки, побудованим на діалозі клієнта та сервера, який здійснюється без передавання секретного знання, навіть у зашифрованій формі, - протоколам з нульовим розголошенням. Як правило, такі протоколи використовують математично складні задачі з невідомим розв'язком зворотної проблеми. Ця невідомість виступає причиною вразливості протоколу – винайдення рішення задачі зведе нанівець надійність протоколу. Запропонована схема автентифікації, побудована на комплексних фрактальних множинах з використанням методів дистанційного оцінювання знань. Фрактальні множини зручні для використання завдяки поєднанню властивостей скінченності та нескінченності. Властивість скінченності дозволяє побудову самої множини, властивість нескінченності гарантує можливість багаторазового використання схеми. Наведено алгоритм клієнт-серверної взаємодії. Використання схеми дозволить покращити показників безпеки мережних інформаційних ресурсів.

**Ключові слова**: автентифікація, протокол з нульовим розголошенням, фрактал, фрактальна множина, мережна безпека.

**Samoilenko Denis,** PhD, docent of Ship Electrical Equipment and Information Security Department, National University of Shipbuilding after Admiral Makarov.
E-mail: denniksam@gmail.com
**Самойленко Денис Миколайович**, кандидат фізико-математичних наук, доцент, доцент кафедри електрообладнання суден та інформаційної безпеки. Національний університет кораблебудування імені адмірала Макарова.
**Самойленко Денис Николаевич**, кандидат физико-математических наук, доцент, доцент кафедры электрооборудования суден и информационной безопасности. Национальный университет кораблестроения имени адмирала Макарова.

**УДК 621.391:519.7**

## БЕЗКЛЮЧОВІ ГЕШ-ФУНКЦІЇ РЕГІСТРОВОГО ТИПУ

***Антон Олексійчук, Катерина Король***

*Безключові геш-функції відносяться до найважливіших криптографічних примітивів і застосовуються в сучасних системах шифрування, автентифікації, цифрового підпису, генерації ключів тощо. Незважаючи на помітний прогрес у розробці різноманітних атак на "конкретні" геш-функції, розуміння закономірностей, що лежать в основі зазначених атак, визначення умов їх застосовності та розробка методів оцінювання їх ефективності є предметом активних подальших досліджень. Метою статті є встановлення загальних умов, що визначають практичну стійкість широкого класу геш-функцій, які базуються на регістрах зсуву, відносно атак, спрямованих на побудування колізій їх стискувальних функцій. Показано, що задача побудування колізій зводиться до розв'язання автоматних рівнянь відносно двійкових невідомих, які задовольняють певним обмеженням. При цьому множини всіх розв'язків таких рівнянь (без урахування обмежень) мають простий алгоритмічний опис, що дозволяє перелічувати ці розв'язки в режимі реального часу.*
***Ключові слова:*** *безключова геш-функція, пошук колізій, скінченний автомат, нелінійний регістр зсуву, система автоматних рівнянь, MDx, SHA.*

**Вступ.** Безключові (криптографічні) геш-функції відносяться до найважливіших криптографічних примітивів. Вони застосовуються в сучасних системах шифрування, автентифікації, цифрового підпису, генерації ключів та звичайно виконують роль своєрідної ланки, що пов'язує окремі частини криптографічної системи (див., наприклад, [4, 5]). З появою у 2004 – 2005 роках потужних атак на окремі функції сім'ї MDx [13 – 16] суттєво підсилився інтерес фахівців до побудови нових видів геш-функцій, розробки методів їх криптоаналізу та обгрунтування їх стійкості відносно перспективних атак. Певним підсумком досліджень у цьому напрямі можна вважати прийняття у 2012 році нового стандарту гешування даних – алгоритму Keccak [8].

Незважаючи на помітний прогрес у розробці різноманітних атак на "конкретні" геш-функції, розуміння закономірностей, що лежать в основі зазначених атак, визначення загальних умов їх застосовності та розробка методів оцінювання їх ефективності є предметом подальших досліджень, спрямованих на створення загальної теорії побудови та аналізу геш-функцій. Потреби в та-

кої теорії відчуваються, зокрема, у зв'язку з наявністю великої кількості евристичних припущень та "ad hock-прийомів", що використовуються у сучасних атаках, більшість з яких базується, скоріше, на інтуіції розробників, ніж на загальних наукових засадах. Свідченням тому є сперечання та дискусії стосовно реальної ефективності або застосовності окремих атак, спростування чи перегляд раніше отриманих результатів (див., наприклад, [3]). Для розуміння причин, за якими окремі атаки є ефективніше (а окремі геш-функції – вразливіше) в порівнянні з іншими, необхідно проаналізувати властивості математичних об'єктів та конструкцій, на яких базуються сучасні алгоритми гешування. Дослідженню одного з найбільш відомих класів таких алгоритмів присвячено дану роботу.

Метою статті є встановлення загальних умов, що визначають практичну стійкість широкого класу геш-функцій, які базуються на регістрах зсуву, відносно атак, спрямованих на побудування колізій їх стискувальних функцій. Оскільки зазначені (а також багато які інші) геш-функції реалізуються за допомогою скінченних автоматів, виявляється природним використовувати для їх аналізу саме теоретико-автоматні методи.

В п. 1 описано конструкцію Меркля-Дамгорда [7, 9] побудування геш-функцій, стійких до колізій, на основі стискувальних функцій, що володіють аналогічною властивістю. В п. 2 введено до розгляду клас функцій регістрового типу, який включає в себе стискувальні функції окремих алгоритмів сім'ї MDx. Показано, що задача побудування колізій зазначених функцій зводиться до розв'язання автоматних рівнянь відносно двійкових невідомих, які задовольняють певним обмеженням. При цьому множини всіх розв'язків таких рівнянь (без урахування обмежень) мають простий алгоритмічний опис, що дозволяє перелічувати ці розв'язки в режимі реального часу. У висновках стисло сформульовано основні результати статті.

**1. Конструкція Меркля-Дамгорда**

Для будь-якого натурального $l$ позначимо $V_l$ множину двійкових векторів довжини $l$, $V_{\le l} = \bigcup_{i=1}^{l} V_i$. Далі під (*безключовою*) *геш-функцією* розуміється довільне відображення $H: V_{\le m} \to V_n$. Якщо це відображення задається за допомогою певного алгоритму, то останній називається *алгоритмом гешування*; він ототожнюється з відображенням $H$ і позначається тим самим символом.

Нехай $X, Y$ – скінченні множини, $h: X \to Y$ – довільне відображення. Говорять, що впорядкована пара $(x_1, x_2) \in X^2$ є *колізією* функції $h$ (або що повідомлення $x_1$ і $x_2$ *утворюють колізію* цієї функції), якщо $x_1 \ne x_2$ та $h(x_1) = h(x_2)$.

Більшість сучасних геш-функцій будується на основі так званої конструкції Меркля-Дамгорда, що запропонована в [9] і незалежно в [7]. Ця конструкція являє собою спосіб побудування геш-функції $H: V_{\le 2^L - 1} \to V_N$ за довільними відображенням $h: V_N \times V_M \to V_N$, вектором $s^{(0)} \in V_N$ та натуральним числом $L < M$.

Нехай $x \in V_{\le 2^L - 1}$ – вхідне повідомлення довжини $l$. Тоді алгоритм обчислення значення $H(x)$ визначається наступним чином:

1) сформувати доповнення повідомлення $x$: $\hat{x} = (x \| 1 \| 0^r \| \vec{l})$, де $\|$ – символ конкатенації, $\vec{l}$ – вектор довжини $L$, що є двійковим записом числа $l$, а $r$ дорівнює остачі від ділення числа $-(L + l + 1)$ на $M$ (іншими словами, $r$ – найменше невід'ємне ціле число, для якого довжина повідомлення $\hat{x}$ є кратною $M$);

2) представити $\hat{x}$ у вигляді: $\hat{x} = \hat{x}_0, ..., \hat{x}_{t-1}$, де $\hat{x}_i \in V_M$ для кожного $i \in \overline{0, t-1}$, обчислити значення

$$s^{(i+1)} = h(s^{(i)}, \hat{x}_i),\ i \in \overline{0, t-1}, \qquad (1)$$

та покласти $H(x) = s^{(t)}$.

Отже, згідно з конструкцією Меркля-Дамгорда, обчислення значень геш-функції $H$ виконується в два етапи. На першому з них здійснюється процедура доповнення (padding), тобто заміна вхідного повідомлення $x$ повідомленням $\hat{x}$. На другому етапі значення $H(x)$ отримується в результаті виконання ітераційної процедури відповідно до формули (1).

Відображення $h: V_N \times V_M \to V_N$ називають іноді *стискувальною функцією* алгоритму гешування $H$. Зауважимо, що це відображення можна розглядати як функцію переходів автомату без виходу з вхідним алфавітом $V_M$ та множиною станів $V_N$. Тоді значення $H(x)$ співпадає з фінальним станом цього автомату, що відповідає його початковому стану $s^{(0)}$ та вхідній послідовності $\hat{x} = \hat{x}_0, ..., \hat{x}_{t-1}$.

**Приклад 1.** Геш-функція SHA-1 (відповідно SHA-256) будується на основі конструкції Меркля-Дамгорда, де $M = 512$, $N = 160$ (відповідно $N = 256$), $L = 64$, а вектор $s^{(0)} \in V_N$ визначається

як певна константа. Стискувальна функція $h$ задається рівністю

$$h(s, x) = s + g(s, x),\ (s, x) \in V_N \times V_M, \quad (2)$$

де + є символом операції абелевої групи $(\mathbf{Z}_{2^{32}})^5$ (відповідно $(\mathbf{Z}_{2^{32}})^8$), а функція $g: V_N \times V_M \to V_N$ визначається за допомогою окремого алгоритму; див. нижче п. 2. Подібним чином будується також геш-функція SHA-512, де $M = 1024$, $N = 512$, $L = 128$, а символ + у формулі (2) позначає операцію абелевої групи $(\mathbf{Z}_{2^{64}})^8$ [10].

Добре відомо, що конструкція Меркля-Дамгорда дозволяє будувати стійкі до колізій геш-функції на основі стискувальних функцій, які володіють аналогічною властивістю. Для уточнення сенсу цього твердження розглянемо, поряд із зазначеним вище повідомленням $x$, інше повідомлення $x' \in V_{\le 2^L - 1}$ довжини $l'$. Позначимо $\hat{x}' = \hat{x}'_0, ..., \hat{x}'_{t'-1}$ його доповнення та покладемо

$$s'^{(j+1)} = h(s'^{(j)}, \hat{x}'_j),\ j \in \overline{0, t'-1}. \quad (3)$$

Припустимо, що $(x, x')$ – колізія функції $H$, причому $l \ne l'$. Тоді $\hat{x}'_{t'-1} \ne \hat{x}_{t-1}$ і на підставі рівностей (1), (3) повідомлення $(s^{(t-1)}, \hat{x}_{t-1})$ і $(s'^{(t-1)}, \hat{x}'_{t'-1})$ утворюють колізію функції $h$. Якщо ж $l = l'$, то $t = t'$ і колізію цієї функції можна знайти серед пар $((s^{(i)}, \hat{x}_i), (s'^{(i)}, \hat{x}'_i))$ таких, що $\hat{x}'_i \ne \hat{x}_i$, $i \in \overline{0, t-1}$.

Отже, в будь-якому випадку за відомою колізією $(x, x')$ функції $H$ можна побудувати колізію її стискувальної функції, використовуючи $O(l(x, x'))$ операцій (звернення до функції $h$ та порівняння на рівність елементів множини $V_M$), де $l(x, x')$ – максимум довжин слів $x$ та $x'$.

З іншого боку, як випливає безпосередньо з наведеного опису конструкції Меркля-Дамгорда, якщо повідомлення $(s^{(0)}, x_1)$ та $(s^{(0)}, x_2)$, де $x_1, x_2 \in V_M$, утворюють колізію функції $h$, то пара $(x_1, x_2)$ є колізією функції $H$.

Останнє твердження, певною мірою, зводить задачу побудування колізій геш-функції $H$ до знаходження колізій функції $h_{s^{(0)}}(x) = h(s^{(0)}, x)$, $x \in V_M$. Якщо при цьому функція $h$ визначається за формулою (2), то колізії функції $h_{s^{(0)}}$ співпадають з колізіями функції $g_{s^{(0)}}(x) = g(s^{(0)}, x)$, $x \in V_M$.

**2. Стискувальні функції, що базуються на регістрах зсуву, та відповідні їм системи автоматних рівнянь**

Опишемо загальну схему побудови стискувальних функцій, за якою будуються, зокрема, зазначені функції окремих алгоритмів сім'ї MDx. Спочатку введемо декілька допоміжних позначень.

Зафіксуємо натуральні числа $l, m, n \ge 2$, покладемо $M = ml$, $N = nl$. Позначимо символом + операцію додавання за модулем $2^l$ двійкових цілих чисел, що відповідають булевим векторам довжини $l$; символом $F_{l \times l}$ – множину матриць розміру $l \times l$ над полем $F = \mathbf{GF}(2)$, а символом $GL(l, 2)$ – групу оборотних матриць порядку $l$ над цим полем.

Для будь-яких функції $f: V_l^{n-1} \to V_l$ та послідовності $A = (A_1, ..., A_n)$, де $A_1, ..., A_n \in GL(l, 2)$, позначимо $\Re(f, A)$ регістр зсуву (РЗ) довжини $n$, що визначається як автомат без виходу з вхідним алфавітом $V_l$, множиною станів $V_l^n$ та функцією переходів

$$\phi(s, z) = (s_n A_n + z + f(s_1, ..., s_{n-1}), s_1 A_1, ..., s_{n-1} A_{n-1}), \quad (4)$$

де $s = (s_1, ..., s_n) \in V_l^n$, $z \in V_l$ (див. рис. 1, де показана схема регістру, що відповідає послідовності $I$ одиничних матриць $A_1, ..., A_n$ над полем $F$).

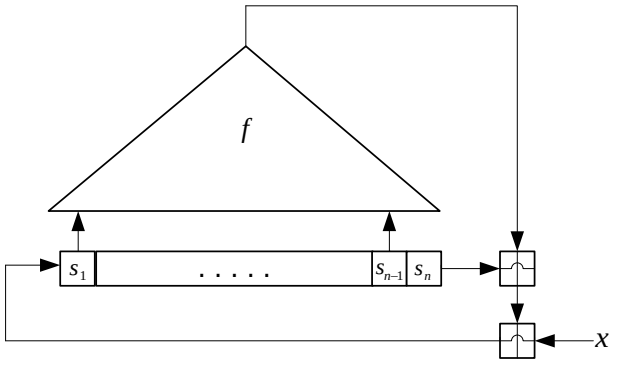



Рис. 1. Регістр зсуву $\Re(f, I)$

Для кожного $z \in R_l$ позначимо $\phi_z$ часткову функцію переходів РЗ $\Re(f, A)$: $\phi_z(s) = \phi(s, z)$, $s \in V_l^n$. Якщо РЗ знаходиться в початковому стані $s^{(0)}$, а на його вхід подається довільна послідовність $z_0, z_1, ...$ двійкових векторів довжини $l$, то стан регістру в $j$-му такті визначається за формулою $s^{(j)} = (\phi_{z_{j-1}} \circ ... \circ \phi_{z_0})(s^{(0)})$, $j = 0, 1, ...$.

Зауважимо, що на підставі формули (4) та оборотності матриць $A_1, ..., A_n$ кожна функція $\phi_z$ є підстановкою на множині $V_l^n$, і для будь-яких двох станів $s, s' \in V_l^n$ РЗ $\Re(f, A)$ існує точно одна послідовність $(z_1, ..., z_n) \in V_l^n$ така, що $s' =$

$(\phi_{z_1} \circ ... \circ \phi_{z_n})(s)$ (іншими словами, $\Re(f, A)$ є регулярним автоматом, будь-який стан якого є досяжним з будь-якого іншого стану за $n$ кроків, і притому єдиним способом).

Для означення класу стискувальних функцій регістрового типу введемо одне допоміжне відображення. Зафіксуємо натуральне число $T > m$, матриці $B_1, ..., B_m \in F_{l\times l}$, комутативну групову операцію $\dot{+}$ на множині $V_l$ та задамо відображення $G: V_l^m \to V_l^T$, вважаючи

$$G(x_0,...,x_{m-1}) = (x_0,...,x_{T-1}),\ (x_0,...,x_{m-1}) \in V_l^m, \quad (5)$$

де

$$x_{i+m} = x_i B_1 \dot{+} x_{i+1} B_2 \dot{+} ... \dot{+} x_{i+m-1} B_m,\ i \in \overline{0, T-m-1}. \quad (6)$$

Нарешті, зафіксуємо довільну групову операцію $*$ на множині $V_N$ та розглянемо стискувальну функцію

$$h(s, x) = s * g(s, x),\ (s, x) \in V_N \times V_M, \quad (7)$$

де

$$g(s, x) = (\phi_{x_{T-1}} \circ ... \circ \phi_{x_0})(s),\ (s, x) \in V_N \times V_M, \quad (8)$$

а вектор $(x_0,...,x_{T-1}) = G(x)$ задається за допомогою співвідношень (5), (6). Обчислення значень функції $h$ відбувається у три етапи. Спочатку за вхідним вектором $x \in V_M$ (або, що те ж саме, за набором $(x_0,...,x_{m-1}) \in V_l^m$) формується послідовність $x_0,...,x_{T-1}$. Далі вона подається на вхід регістру $\Re(f, A)$, переводячи його з початкового стану $s$ у фінальний стан $g(s, x)$. Нарешті, обчислюється значення $h(s, x)$ за формулою (7).

Зауважимо, що за описаною схемою будуються стискувальні функції окремих алгоритмів сім'ї MDx, лише з тією відмінністю, що функція $f$ у виразі підстановки $\phi_{x_j}$ (див. формули (4), (8)), взагалі кажучи, залежить від номера $j \in \overline{0, T-1}$. Приймаючи це до уваги, сформулюємо таке означення.

**Означення.** Нехай $M = ml$, $N = nl$, де $l, m, n$ – натуральні числа, більші за 1, $*$ – групова операція на множині $V_N$. Тоді відображення $h$ вигляду (7) називається *стискувальною функцією регістрового типу*, якщо існують натуральне число $T > m$, функції $f_j: V_l^{n-1} \to V_l$, $j \in \overline{0, T-1}$, і матриці $A_1, ..., A_n \in GL(l, 2)$, $B_1, ..., B_m \in F_{l\times l}$ такі, що

$$g(s, x) = (\phi_{x_{T-1}}^{(T-1)} \circ ... \circ \phi_{x_0}^{(0)})(s),\ (s, x) \in V_N \times V_M, \quad (9)$$

де вектор $(x_0,...,x_{T-1}) = G(x)$ визначається за формулами (5), (6), а

$$\phi_z^{(j)}(s) = (s_n A_n + z + f_j(s_1,...,s_{n-1}), s_1 A_1, ..., s_{n-1} A_{n-1}) \quad (10)$$

для будь-яких $s = (s_1,...,s_n) \in V_l^n$, $z \in V_l$, $j \in \overline{0, T-1}$.

**Приклад 2.** Стискувальна функція алгоритму SHA-1 є функцією регістрового типу. При цьому $l = 32$, $m = 16$, $n = 5$, $T = 80$, а матриці $A_1, ..., A_5$ у формулі (10) визначаються таким чином: $A_1 = A_3 = A_4 = A_5 = I_{32}$, $A_2 = C^{30}$, де $I_{32}$ – одинична матриця порядку 32, $C$ – матриця оператора циклічного зсуву на 1 біт ліворуч. Функція $f_j$ має наступний вигляд:

$f_j(s_1, s_2, s_3, s_4) = s_1 C^5 + \tilde{f}_j(s_2, s_3, s_4) + K_j$, $s_1, s_2, s_3, s_4 \in V_{32}$, де $K_j \in V_{32}$ – певна константа,

$$\tilde{f}_j(s_2, s_3, s_4) = \begin{cases} s_2 s_3 \oplus \overline{s_2} s_4, & 0 \le j \le 19; \\ s_2 \oplus s_3 \oplus s_4, & 20 \le j \le 39;\ 60 \le j \le 79; \\ s_2 s_3 \oplus s_2 s_4 \oplus s_3 s_4, & 40 \le j \le 59, \end{cases}$$

а операції булева множення, додавання та заперечення виконується над векторами $s_2, s_3, s_4$ покоординатно. Крім того, операція $\dot{+}$ у формулі (6) співпадає з покоординатним булевим додаванням на множині $V_l$, а відображення $G$ є лінійним над полем $F$: $G(x_0,...,x_{15}) = (x_0,...,x_{79})$, де $x_{i+16} = (x_i \oplus x_{i+2} \oplus x_{i+8} \oplus x_{i+13})C$, $i \in \overline{0, 63}$ [10].

Зауважимо, що для побудови стискувальних функцій можна використовувати регістри зсуву більш загального вигляду, що визначаються за допомогою довільних відображень множини $V_l^{n+1}$ у множину $V_l$.

Для будь-якого відображення $F_j: V_l^{n+1} \to V_l$ та послідовності $\tilde{A} = (A_1, ..., A_{n-1})$, де $A_1, ..., A_{n-1} \in GL(l, 2)$, позначимо $\tilde{\Re}(F_j, \tilde{A})$ автомат без виходу з вхідним алфавітом $V_l$, множиною станів $V_l^n$ та функцією переходів

$$\phi_z^{(j)}(s) = (F_j(z, s_1,...,s_n), s_1 A_1, ..., s_{n-1} A_{n-1}),$$

$$s = (s_1,...,s_n) \in V_l^n,\ z \in V_l,\ j \in \overline{0, T-1}. \quad (11)$$

Зрозуміло, що зазначений автомат співпадає з означеним вище РЗ $\Re(f_j, A)$ у випадку, коли $A = (\tilde{A}, A_n)$, $F_j(z, s_1,...,s_n) = s_n A_n + z + f_j(s_1,...,s_{n-1})$ для будь-яких $z, s_1,...,s_n \in V_l$, $j \in \overline{0, T-1}$. Отже, замінюючи у формулюванні означення функцій регістрового типу формулу (10) на формулу (11), отримаємо більш широкий клас стискувальних

функцій, який охоплює майже всі відомі алгоритми гешування сім'ї MDx (при цьому, як правило, відображення $F_j : V_l^{n+1} \to V_l$, $j \in \overline{0,T-1}$, є бієктивним за кожною змінною $z$, $s_n$ при будь-яких фіксованих значеннях решти).

**Приклад 3.** Стискувальна функція алгоритму MD5 будується на основі РЗ $\tilde{\Re}(F_j, \tilde{A})$, де $l = 32$, $m = 16$, $n = 4$, $T = 64$, $A_1 = A_2 = A_3 = I_{32}$, $F_j(z, s_1, s_2, s_3, s_4) = s_1 + (f_j(s_1, s_2, s_3) + s_4 + z + K_j)C^{\nu_j}$, $C$ – матриця оператора циклічного зсуву на 1 біт ліворуч, $K_j, \nu_j$ – певні константи,

$$f_j(s_1,s_2,s_3) = \begin{cases} s_1 s_2 \oplus \overline{s_1} s_3, & 0 \le j \le 15; \\ s_1 s_3 \oplus \overline{s_3} s_2, & 16 \le j \le 31; \\ s_1 \oplus s_2 \oplus s_3, & 32 \le j \le 47; \\ s_2 \oplus (s_1 \vee \overline{s_3}), & 48 \le j \le 63. \end{cases}$$

Зауважимо, що відображення $F_j$, $j \in \overline{0,63}$, є бієктивним за кожною змінною $z$, $s_4$ при фіксованих значеннях решти. При цьому відображення $G$, що визначає спосіб формування вхідної послідовності $x_0, ..., x_{T-1}$ за її початковим відрізком $(x_0, ..., x_{m-1}) \in V_l^m$, має вигляд, відмінний від (6), і задається наступним чином:

$$x_i = \begin{cases} x_{(1+5i) \bmod 16}, & 16 \le i \le 31; \\ x_{(5+3i) \bmod 16}, & 32 \le i \le 47; \\ x_{(7i) \bmod 16}, & 48 \le i \le 63. \end{cases}$$

Розглянемо зараз довільну стискувальну функцію регістрового типу $h : V_N \times V_M \to V_N$ і довільний вектор $s^{(0)} \in V_N$. Зрозуміло, що побудування колізій функції $h_{s^{(0)}}$ рівносильно розв'язанню системи рівнянь

$$(\phi_{x_{T-1}}^{(T-1)} \circ ... \circ \phi_{x_0}^{(0)})(s^{(0)}) = (\phi_{x'_{T-1}}^{(T-1)} \circ ... \circ \phi_{x'_0}^{(0)})(s^{(0)}), \quad (12)$$

$$(x_0, ..., x_{T-1}) = G(x_0, ..., x_{m-1}),$$

$$(x'_0, ..., x'_{T-1}) = G(x'_0, ..., x'_{m-1}) \quad (13)$$

відносно пари різних невідомих $x, x' \in V_M$. Зауважимо, що перше з цих двох рівнянь (відносно довільних наборів невідомих $(x_0, ..., x_{T-1})$, $(x'_0, ..., x'_{T-1}) \in V_l^T$, без урахування обмеження (13)) розв'язується тривіально. Це випливає з наступного твердження.

**Твердження.** Нехай $A_1, ..., A_n \in GL(l, 2)$, $f_j : V_l^{n-1} \to V_l$, а функція $\phi_z^{(j)}$ визначається за формулою (10), $z \in V_l$, $j \in \overline{0,T-1}$. Тоді для будь-яких $s, s' \in V_l^n$, $j_0, ..., j_{n-1} \in \overline{0,T-1}$ існує точно один вектор $(x_0, ..., x_{n-1}) \in V_l^n$ такий, що $s' = (\phi_{x_{n-1}}^{(j_{n-1})} \circ ... \circ \phi_{x_0}^{(j_0)})(s)$.

**Доведення.** Нехай для простоти позначень $j_i = i$, $i \in \overline{0,n-1}$. Покладемо

$$s^{(i)} = (\phi_{x_{i-1}}^{(i-1)} \circ ... \circ \phi_{x_0}^{(0)})(s^{(0)}) = (s_1^{(i)}, ..., s_n^{(i)}), \ i \in \overline{0,n}. \quad (14)$$

Треба показати, що вектор $(x_0, ..., x_{n-1})$ однозначно визначається з системи рівнянь (14) за відомими значеннями $s^{(0)}$ та $s^{(n)}$.

З формул (10) та (14) випливають наступні рівності:

$$s_1^{(i)} = s_n^{(i-1)} A_n + f_{i-1}(s_1^{(i-1)}, ..., s_{n-1}^{(i-1)}) + x_{i-1},$$

$$s_2^{(i)} = s_1^{(i-1)} A_1, ..., s_{n-1}^{(i)} = s_{n-2}^{(i-1)} A_{n-2}, s_n^{(i)} = s_{n-1}^{(i-1)} A_{n-1}, \ i \in \overline{1,n}. \quad (15)$$

Використовуючи формули (15), отримаємо, що $s_n^{(n)} = s_{n-1}^{(n-1)} A_{n-1} = s_{n-2}^{(n-2)} A_{n-2} A_{n-1} = ... = s_1^{(1)} A_1 \cdots A_{n-1} = (s_n^{(0)} A_n + f_0(s_1^{(0)}, ..., s_{n-1}^{(0)}) + x_0) A_1 \cdots A_{n-1}$.

З останньої рівності можна однозначно відновити $x_0$ за відомими $s^{(0)}$ та $s^{(n)}$, а, отже, обчислити $s^{(1)}$ за формулою (14).

Далі, використовуючи формули (15), отримаємо, що $s_{n-1}^{(n)} = s_{n-2}^{(n-1)} A_{n-2} = s_{n-3}^{(n-2)} A_{n-3} A_{n-2} = ... = s_1^{(2)} A_1 \cdots A_{n-2} = (s_n^{(1)} A_n + f_1(s_1^{(1)}, ..., s_{n-1}^{(1)}) + x_1) A_1 \cdots A_{n-2}$.

Звідси однозначно відновимо $x_1$ та обчислимо значення $s^{(2)}$ за формулою (14).

Продовжуючи аналогічні міркування, однозначно відновимо усі координати вектора $(x_0, ..., x_{n-1})$. Твердження доведено.

**Наслідок 1.** За умовою твердження бінарна операція

$$(s, x) \mapsto (\phi_{x_{n-1}}^{(j_{n-1})} \circ ... \circ \phi_{x_0}^{(j_0)})(s), \ s, x = (x_0, ..., x_{n-1}) \in V_l^n$$

є квазігрупою на множині $V_l^n$ (тобто оборотна за кожним аргументом при фіксованому іншому).

**Наслідок 2.** Нехай $T = nt$. Тоді для будь-яких $s, s' \in V_l^n$ рівняння

$$(\phi_{x_{T-1}}^{(T-1)} \circ ... \circ \phi_{x_0}^{(0)})(s) = s'$$

має точно $2^{l(T-n)}$ розв'язків $(x_0, ..., x_{T-1}) \in V_l^T$, які знаходяться у взаємно однозначній відповідності з наборами $(s^{(n)}, s^{(2n)}, ..., s^{(t-1)n)}) \in V_N{}^t$.

Зауважимо, що отримане твердження та наслідки з нього залишаються справедливими й у більш загальному випадку, коли функція $\phi_z^{(j)}$ визначаються за формулою (11), а відображення $F_j : V_l^{n+1} \to V_l$ є бієктивним за кожною змінною $z$, $s_n$ при будь-яких фіксованих значеннях решти, $j \in \overline{0,T-1}$.

Таким чином, якщо $h: V_N \times V_M \to V_N$ є функцією регістрового типу, то для будь-якого вектора $s^{(0)} \in V_N$ множина всіх розв'язків $((x_0,...,x_{T-1}), (x'_0,...,x'_{T-1})) \in V_l^T \times V_l^T$ рівняння (12) має простий алгоритмічний опис, що дозволяє перелічувати ці розв'язки в режимі реального часу. Тому основна трудність побудування колізій функції $h_{s^{(0)}}$ полягає в знаходженні саме таких розв'язків, які задовольняють додатковому обмеженню (13).

Для розв'язання систем рівнянь (12), (13), в принципі, можуть бути застосовані відомі загальні методи, зокрема, ймовірнісний метод [2] або метод статистичних наближень скінченних автоматів [1, 6]. Зауважимо, що ідеї, на яких базуються останні два методи, використовуються при побудові так званих різнецевих атак на стискувальні функції алгоритмів гешування сім'ї MDx [6, 11– 16].

**Висновки.** Стискувальні функції майже усіх сучасних алгоритмів гешування сім'ї MDx будуються на основі певних регістрів зсуву над алфавітом потужності $2^l$, де, як правило, $l = 32$. Задача побудування колізій стискувальних функцій регістрового типу рівносильна розв'язанню систем автоматних рівнянь вигляду (12) при обмеженнях (13), причому множини всіх розв'язків таких рівнянь (без урахування обмежень) мають простий алгоритмічний опис, що дозволяє перелічувати ці розв'язки в режимі реального часу. Отже, стійкість зазначених геш-функцій відносно атак, спрямованих на побудування колізій, базується на складності знаходження хоча б одного елементу перетину двох скінченних множин, кожна з яких задається за допомогою простих алгоритмічних процедур. Фактичне значення стійкості кожного окремого алгоритму гешування з означеного класу залежить від властивостей функції зворотного зв'язку (вигляду (10) або (11)) відповідного регістру зсуву та відображення $G$ вигляду (5).

Однією з задач подальших досліджень є побудова систем автоматних рівнянь, що описують стискувальні функції більш складних алгоритмів гешування, подібних SHA-256 та SHA-512. Зокрема, викликає інтерес відповідь на запитання про справедливість для зазначених алгоритмів аналогу твердження, доведеного у даній статті.

## ЛІТЕРАТУРА


[1]. Бабаш А.В. Решение автоматных уравнений с искажениями в функции перехода автомата // Проблемы передачи информации. – 2002. – Т. 38. – Вып. 3. – С. 62-71.

[2]. Балакин Г.В. О вероятностном подходе к решению систем уравнений с целочисленными неизвестными // Дискретная математика. – 1995. – Т. 7. – Вып. 1. – С. 88- 98.

[3]. Карпунин Г.А., Ермолаева Е.З. Оценки сложности поиска коллизий для хэш-функции RIPEMD // Прикладная дискретная математика. Приложение. – 2012. – № 5. – С. 43- 44.

[4]. Фергюссон Н., Шнайер Б. Практическая криптография: Пер. с англ. – М.: "Вильямс", 2005. – 424 с.

[5]. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. – М.: Триумф, 2002. – 816 с.

[6]. Chabaud F., Joux A. Differential collision search attack on SHA-0 // Advances in Cryptology – CRYPTO'98, Proceedings. – Springer-Verlag, 1998. – P. 56- 71.

[7]. Damgård I B. A design principle for hash function // Advances in Cryptology – CRYPTO'89, Proceedings. – Springer-Verlag, 1990. – P. 416- 427.

[8]. ECRYPT II: Final hash function status report / http://www.ecrypt.eu.org/documents/D.SYM.11, 31 Jan., 2013.

[9]. Merkle R.C. On way hash function and DES // Advances in Cryptology – CRYPTO'89, Proceedings. – Springer-Verlag, 1990. – P. 428-436.

[10]. National Institute of Standards and Technology (NIST). FIPS-180-2: Secure Hash Standard, http://www.itl.nist.gov/fipspubs.

[11]. Pramstaller N., Rechberger C., Rijmen V. Exploiting coding theory for collision attacks on SHA-1 // Cryptography and Coding 2005, Proceedings. – Springer-Verlag, 2005. – P. 78- 95.

[12]. Rijmen V., Oswald E. Update on SHA-1 // Topics in Cryptology. – CT-RSA, Proceedings. – Springer-Verlag, 2005. – P. 58- 71.

[13]. Wang X., Lai X., Feng D., Chen H., Yu X. Cryptanalysis of the hash functions MD4 and RIPEMD // Advances in Cryptology – EUROCRYPT'2005, Proceedings. – Springer-Verlag, 2005. – P. 1- 18.

[14]. Wang X., Yu H. How to break MD5 and other hash functions // Advances in Cryptology – EUROCRYPT'2005, Proceedings. – Springer-Verlag, 2005. – P. 19-35.

[15]. Wang X., Yu H., Yin Y.L. Efficien collisions search attacks on SHA-0 // Advances in Cryptology – CRYPTO'2005, Proceedings. – Springer-Verlag, 2005. – P. 1-16.

[16]. Wang X., Yu H., Yin Y.L. Finding collisions in the full SHA-1 // Advances in Cryptology – CRYPTO'2005, Proceedings. – Springer-Verlag, 2005. – P. 17-36.


## REFERENCES


[1]. Babash A. V. Solving automaton equations with distortions in the automaton transition function. Problems of Information Transmission, 2002, vol. 38(3), pp. 218-226.

[2]. Balakin G. V. On a probabilistic approach to solving systems of equations with integer-valued unknowns, Discrete Mathematics and Applications, 1995, vol. 5(1), pp. 43-51.

[3]. Kapurnin G. A., Ermolaeva E. Z. Estimates of collision resistance comlexity for the hash function RIPEMD. Discrete Applied Mathematics. Application. 2012, № 5, pp. 43- 44.
[4]. Ferguson N., Schneier B. Practical cryptography. John Wiley & Sons, 2003, 432 p.
[5]. Schneier B. Applied cryptography: Protocols, algorithms, and source code in C, 2nd Edition. John Wiley & Sons, 1995, 784 p.
[6]. Chabaud F., Joux A. Differential collision search attack on SHA-0. Advances in Cryptology – CRYPTO'98, Springer-Verlag, 1998, pp. 56-71.
[7]. Damgård I. B. A design principle for hash function. Advances in Cryptology – CRYPTO'89, Springer-Verlag, 1990, pp. 416-427.
[8]. ECRYPT II: Final hash function status report. http://www.ecrypt.eu.org/documents/D.SYM.11, 31 Jan., 2013.
[9]. Merkle R.C. On way hash function and DES. Advances in Cryptology – CRYPTO'89., Springer-Verlag, 1990, pp. 428-436.
[10]. National Institute of Standards and Technology (NIST). FIPS-180-2: Secure Hash Standard, http://www.itl.nist.gov/fipspubs.
[11]. Pramstaller N., Rechberger C., Rijmen V. Exploiting coding theory for collision attacks on SHA-1. Cryptography and Coding 2005, Springer-Verlag, 2005, pp. 78-95.
[12]. Rijmen V., Oswald E. Update on SHA-1. Topics in Cryptology. CT-RSA, Springer-Verlag, 2005, pp. 58-71.
[13]. Wang X., Lai X., Feng D., Chen H., Yu X. Cryptanalysis of the hash functions MD4 and RIPEMD. Advances in Cryptology – EUROCRYPT'2005, Springer-Verlag, 2005, pp. 1-18.
[14]. Wang X., Yu H. How to break MD5 and other hash functions. Advances in Cryptology – EUROCRYPT'2005, Springer-Verlag, 2005, pp. 19-35.
[15]. Wang X., Yu H., Yin Y.L. Efficien collisions search attacks on SHA-0. Advances in Cryptology – CRYPTO'2005, Springer-Verlag, 2005, pp. 1-16.
[16]. Wang X., Yu H., Yin Y.L. Finding collisions in the full SHA-1. Advances in Cryptology – CRYPTO'2005, Springer-Verlag, 2005, pp. 17-36.

## БЕЗКЛЮЧЕВЫЕ ХЭШ-ФУНКЦИИ РЕГИСТРОВОГО ТИПА

Безключевые хэш-функции относятся к наиболее важным криптографическим примитивам и применяются в современных системах шифрования, аутентификации, цифровой подписи, генерации ключей и т.д. Несмотря на заметный процесс в разработке различных атак на "конкретные" хэш-функции, понимание закономерностей, лежащих в основе указанных атак, нахождение условий их применимости и разработка методов оценивания их эффективности являются предметом активных дальнейших исследований. Цель статьи состоит в установлении общих условий, определяющих практическую стойкость широкого класса хэш-функций, основанных на регистрах сдвига, относительно атак, направленных на построение коллизий их сжимающих функций. Показано, что задача построения коллизий сводится к решению автоматных уравнений относительно двоичных неизвестных, удовлетворяющих определенным ограничениям. При этом множества всех решений таких уравнений (без учета ограничений) имеют простое алгоритмическое описание, что позволяет перечислять эти решения в режиме реального времени.
**Ключевые слова:** безключевая хэш-функция, поиск коллизий, конечный автомат, нелинейный регистр сдвига, система автоматных уравнений, MDx, SHA.

## KEYLESS HASH FUNCTIONS OF SHIFT REGISTERS TYPE

Keyless hash functions are one of the most important cryptographic primitives and are used in modern encryption, authentication, digital signature, keys generation systems, etc. Although conspicuous progress in developing of various attacks on "specific" hash functions, understanding of the principles underlying these attacks, determining the conditions of their applicability, and development of methods for their performance estimating is an active subject of further research. The goal of this paper is to establish the general conditions of practical security for a broad class of hash functions based on shift registers against collision search attacks on their compression functions. It is shown that the problem of building of collisions can be reduced to solving some automaton equations with binary unknowns satisfying certain constraints. Then the set of all solutions of these equations (without constraints) have simple algorithmic description that allows to enumerate these solutions in the real time-mode.
**Index Terms:** keyless hash function, collision search attacks, finite state automaton, nonlinear shift register, system of automaton equations, MDx, SHA.

**Олексійчук Антон Миколайович**, доктор технічних наук, професор Інституту спеціального зв'язку та захисту інформації НТУУ «КПІ».
E-mail: alex-crypto@mail.ru
**Алексейчук Антон Николаевич,** доктор технических наук, профессор Института специальной связи и защиты информации НТУУ «КПИ».
**Alekseychuk Anton,** Doctor of Technical Science, Professor of Institute of Special Communication and Information Security of NTUU «KPI».

**Король Катерина Вікторівна,** аспірантка Донецького національного університету.
E-mail: kate.lorok@gmail.com
**Король Екатерина Викторовна,** аспирантка Донецкого национального университета.
**Korol Kate,** post-graduate student of Donetsk National University.