ISSN NO: 2770-0003
Date of Publication: 10-06-2022

Application Of Information Compression to Create New Hash Functional Algorithms of Rectangal Matrix Introduction

Akhmadaliev Shakhobidin Sharifovich
Kokand state pedagogical institute, senior teacher, e-mail: 356_qabul_2009@mail.ru
Hasanov Xayrullo Maxmudovich
Kokand state pedagogical institute, teacher, e-mail: xayrullo-xasanov@mail.ru
Botirov Muzaffar Mansurovich
Kokand state pedagogical institute, teacher, e-mail: qdpi_muzaffar@mail.ru

Annotation: *In article on the basis of results of research of properties of compression and single-sided rectangular matrixes the new algorithm of hashing function is developed.*

Keywords: *rectangular matrix, data blocks, compression, application methods, one-way property, algorithm, hash function.*

A hash function is a function that converts information of any length (in bits or bytes) to a fixed length (in bits or bytes) [1-3].

In cryptography, hash functions are used to solve the following problems:

- to control the integrity of data transmission or storage;
- to authenticate the data source.

To control the integrity of a data transmission or storage, the hash value of each data is calculated and this value is stored or transmitted along with the data. The user who receives the data calculates the hash value of the data and compares it with the control value. If these values do not match in the comparison, it means that the data has changed [1-3].

Hash functions are divided into two important types, key and keyless hash functions.

Unlike the control set, hash functions not only detect random errors that occur in data storage and transmission, but also alert you to active attacks by an opponent. The hash function has a secret key so that the competitor cannot calculate the hash value independently and successfully imitate or modify the data, and is known only to the parties transmitting and receiving the data. Such hash functions are called key hash functions.

Impersonations generated from key hash functions are used to prevent fabrication in impersonation attacks and to modify data in attacks of substitution type.

The issue of data source authentication arises in the exchange of information between non-trusted users of information and communication systems. It uses a digital signature scheme that allows you to authenticate the data source. Typically, information is compressed using a hash function that identifies an error code before a digital signature based on the user's secret key. In this case, the hash function will not have a secret key and its algorithm will be known to all users. Such a hash function is called keyless. The main requirement is that it is not possible to change a digitally signed document and select two different data with the same hash value.

Thus, the hash function $h: X \to U$ is said to be an easy-to-calculate one-way function that reflects the information M of arbitrary length to the value $h(M) = H$ of fixed length.

The hash function has the following requirements [1-3]:

1. Can be applied to text of arbitrary length.
2. Returns the value of the specified length at the output.
3. On an arbitrarily given x, h (x) is easy to calculate.
4. It is not possible to calculate x from the equation h (x) = N on an arbitrarily given H. (One-sided property)
5. For the resulting x and $y \neq x$ texts, $h(x) \neq h(y)$. (Collision tolerance property)

ISSN NO: 2770-0003
Date of Publication: 10-06-2022

It should be noted that the number of possible data is greater than the number of possible hash values, so each hash value corresponds to several sets of texts, i.e., a set of data with the same hash value.
When using key hash functions, they have the following basic requirements:
lack of manufacturing capacity;
the impossibility of modification.
The first requirement means that when a hash value is assigned, it will be difficult to select the information that matches it. The second requirement means that when data and its hash value are given, it will be difficult to select other data whose hash value is equal to it.
Sometimes, these two properties are combined into one stronger property - the computational tolerance property. This requirement indicates the complexity of selecting other data x, x ≠ x_i, i = 1, l for which the hash values are equal to one of the given data {x_1, x_2,…, x_l} with known hash values.
Complexity is the computational complexity of a problem that cannot be solved in real time using modern computing devices.
Key hash functions are used between trusted parties and they have a common secret key. Usually, in these circumstances, it is not required to protect the information and communication system from the fact that the other party has not received or changed the information. Therefore, collision tolerance is not required from key hash functions.
Attacks on key hash functions can occur, such as "imitation", ie the transfer of fake information on an empty channel, and the exchange of transmitted data for fake information.
The computational tolerance property implies that it is not possible to determine the key used in the hash function, while knowing the key makes it possible to calculate the hash value of any data. The reverse confirmation is not appropriate, because in some cases, without knowing the key in advance, the hash value is set to t

1. $S = S_0, \ldots, S_{255}$ маълум сонлар блоки киритилади, бу ерда $0 \le S_l \le 255, l = \overline{0{,}255}$ (бу S блок фойдаланувчилар гуруҳи учун умумий бўлади). $H = 0$ ва $n = 1$ деб олинади.
2. М$_n$-хэшланувчи маълумот блоки киритилади:

$$T(0) = \mathrm{M}_n = t_0(0)t_1(1)\ldots t_{255}(0) \ \ (256 \text{ бит}).$$

3. $T(0)$ блокдан қуйидагича 32 та калит ҳосил қилиб олинади:

$$k_0 = t_0(0)t_1(0)\ldots t_7(0), k_1 = t_8(0)t_9(0)\ldots t_{15}(0), \rightleftarrows\rightleftarrows\ldots, \rightleftarrows\rightleftarrows\ k_{31} = t_{248}(0)t_{249}(0)\ldots t_{255}(0).$$

$T(0)$ блок чапга 29 бит циклик сурилади ва у $T(1)$ орқали белгиланади:

$$T(1) = T(0) <<< 29 = t_{29}(0)t_{30}(0)\ldots t_{255}(0)t_0(0)\ldots t_{28}(0) = t_0(1)t_1(1)\ldots t_{255}(1).$$

$T(1)$ блокдан қуйидагича 32 та калит ҳосил қилиб олинади:

$$k_{0+32\cdot1} = t_0(1)t_1(1)\ldots t_7(1), k_{1+32\cdot1} = t_8(1)t_9(1)\ldots t_{15}(1), \rightleftarrows\rightleftarrows\ldots, \rightleftarrows\rightleftarrows\ k_{31+32\cdot1} = t_{248}(1)t_{249}(1)\ldots t_{255}(1).$$

$T(1)$ блок чапга 29 бит циклик сурилади ва у $T(2)$ орқали белгиланади:

$$T(2) = T(1) <<< 29 = t_{29}(1)t_{30}(1)\ldots t_{255}(1)t_0(1)\ldots t_{28}(1) = t_0(2)t_1(2)\ldots t_{255}(2).$$

$T(2)$ блокдан қуйидагича 32 та калит ҳосил қилиб олинади:

$$k_{0+32\cdot2} = t_0(2)t_1(2)\ldots t_7(2), k_{1+32\cdot2} = t_8(2)t_9(2)\ldots t_{15}(2), \ldots, k_{31+32\cdot2} = t_{248}(2)t_{249}(2)\ldots t_{255}(2).$$

Юқоридагилар каби $T(3)$, $T(4)$, $T(5)$, $T(6)$ ларнинг ифодасидан қуйидаги калитлар ҳосил қилинади: $k_{0+32\cdot3}, \ldots, k_{31+32\cdot3}$, $k_{0+32\cdot4}, \ldots, k_{31+32\cdot4}$, $k_{0+32\cdot5}, \ldots, k_{31+32\cdot5}, k_{0+32\cdot6}, \ldots, k_{31+32\cdot6}$.
$T(6)$ блок чапга 29 бит циклик сурилади ва у $T(7)$ орқали белгиланади:

$$T(7) = T(6) <<< 29 = t_{29}(6)t_{30}(6)\ldots t_{255}(6)t_0(6)\ldots t_{28}(6) = t_0(7)t_1(7)\ldots t_{255}(7).$$

$T(7)$ блокдан қуйидагича 32 та калит ҳосил қилиб олинади:

$$k_{0+32\cdot7} = t_0(7)t_1(7)\ldots t_7(7), k_{1+32\cdot7} = t_8(7)t_9(7)\ldots t_{15}(7), \ldots, k_{31+32\cdot7} = t_{248}(7)t_{249}(7)\ldots t_{255}(7).$$

4. Бу ҳосил қилиб олинган бошланғич калитлар ва S блок ёрдамида қуйида келтирилган мураккаб бўлмаган акслантиришлар асосида қисм калитлар учун (32 байтдан иборат бўлган 8 та) 256 байтдан иборат калитлар ҳосил қилинади:
4.1. $i = 0$; $j = 0$;
4.2. $j = (j + S_i + k_i) \bmod 256$;
4.3. $P = S_i$; $Q = S_j$;

4.4. $S_i = Q;\ S_j = P;$
4.5. $i < 255$ шарт текширилади. Агар бу шарт бажарилса, $i = i + 1$ деб олиниб, 4.2. қадамга ўтилади, акс ҳолда кейинги қадамга ўтилади;
4.6. $i = 0;\quad j = 0;\quad l = 0;$
4.7. $i = (i + 1)\ mod\ 256;$
4.8. $j = (j + S_i)\ mod\ 256;$
4.9. $P = S_i;\ Q = Sj;$
4.10. $S_i = Q;\ S_j = P;$
4.11. $t = (S_i + S_j)\ mod\ 256;$
4.12. $k_l = S_t;$
4.13. $l = l + 1;$
4.14. $l < 256$ шарт текширилади. Агар бу шарт бажарилса, 4.7-қадамга ўтилади, акс ҳолда $K = k_0, k_1, \ldots, k_l, \ldots, k_{255}$ калитлар ҳосил қилиниши тугатилади.
5. $i = 0$.
6. $K = k_0\ k_1 \ldots k_{32}$ (256бит = 32байт) қисм калит билан $\mathrm{T}(0) = t_0(0) \ldots t_7(0) t_8(0) \ldots t_{15}(0) \ldots t_{248}(0) \ldots t_{255}(0)$ блок устида $\oplus$ амали бажарилади ва натижа $H(0) = h_0(0) h_1(0) \ldots h_{255}(0)$ деб белгиланади, яъни $K \oplus T(0) = H(0)$.
7. Ҳосил бўлган 256 битли блок тўртта 64 битли блокларга ажратилади
$X = h_0(0) \ldots h_{63}(0),\ Y = h_{64}(0) \ldots h_{127}(0),\ Z = h_{128}(0) \ldots h_{191}(0),\ W = h_{192}(0) \ldots h_{255}(0)$ ва қуйидаги мантиқий функцияларнинг қийматлари ҳисобланади:
$F(X; Y; Z; W) = (X \wedge Y) \vee (Z \wedge W);$
$G(X; Y; Z; W) = (X \wedge Z) \vee (Y \wedge W);$
$R(X; Y; Z; W) = X \oplus Y \oplus Z \oplus W;$

$$V(X; Y; Z; W) = (X \vee Y) \oplus (\overline{Z} \vee \overline{W}).$$

Бу ерда битлар бўйича мантиқий AND, OR, NOT, XOR амаллари мос равишда $\wedge, \vee, ^{-}, \oplus$ белгилари билан ифодаланган. Бу мантиқий функцияларнинг қийматлари 64 битли блоклар бўлади.
8. Юқорида берилган тўртта 64 битли блоклар конкатенация қилинади ва ҳосил қилинган 256 битли блок $L(0)$ орқали белгиланади:
$L(0) = F(X; Y; Z; W) || G(X; Y; Z; W) || R(X; Y; Z; W) || V(X; Y; Z; W) = l_0(0) l_1(0) \ldots l_{255}(0).$
9. $H(0)$ ва $L(0)$ конкатенация қилинади ва ҳосил қилинган 512 битли блок А(0) орқали белгиланади:

$$A(0) = H(0) || L(0) = h_0(0) h_1(0) \ldots h_{255}(0) l_0(0) l_1(0) \ldots l_{255}(0) = a_0(0) a_1(0) \ldots a_{511}(0)$$

10. А(0) блокни 1(бир) байтли блокларга ажратиб олинади:
$a_0(0) a_1(0) a_2(0) a_3(0)\quad a_4(0) a_5(0) a_6(0) a_7(0) = x_0,\quad a_8(0) a_9(0) a_{10}(0) a_{11}(0)$ $a_{12}(0) a_{13}(0) a_{14}(0) a_{15}(0) = x_1, \ldots\ , a_{0+8\gamma}(0) a_{1+8\gamma}(0) a_{2+8\gamma}(0) a_{3+8\gamma}(0)$
$a_{4+8\gamma}(0) a_{5+8\gamma}(0) a_{6+8\gamma}(0) a_{7+8\gamma}(0) = x_\gamma, \ldots\ ,\ a_{0+8\cdot 63}(0) a_{1+8\cdot 63}(0) a_{2+8\cdot 63}(0) a_{3+8\cdot 63}(0)$
$a_{4+8\cdot 63}(0) a_{5+8\cdot 63}(0) a_{6+8\cdot 63}(0) a_{7+8\cdot 63}(0) = x_{63}.$
деб белгиланади.

11. Берилган $C_{4\times 8} = \begin{pmatrix} c_{11} & c_{12} & \ldots c_{18} \\ c_{21} & c_{22} & \ldots c_{28} \\ c_{31} & c_{32} & \ldots c_{38} \\ c_{41} & c_{42} & \ldots c_{48} \end{pmatrix}$ –матрица билан координаталари байтлардан иборат $x_{\gamma 8\times 1} = (x_{0+8\cdot\gamma}, x_{1+8\cdot\gamma}, x_{2+8\cdot\gamma}, \ldots, x_{7+8\cdot v})$ –векторларни $y_{\gamma 4\times 1} = (y_{0+4\cdot\gamma}, y_{1+4\cdot\gamma}, y_{2+4\cdot\gamma}, y_{3+4\cdot\gamma})$ – векторларга акслантирилиши натижасида 512 битли блок 256 битли блокка сиқилади ва сиқиш натижасида олинган блокни $T(1)$ орқали белгиланади:

$$C_{4\times 8} x_{\gamma 8\times 1} mod 256 = \begin{pmatrix} c_{11} & c_{12} & \ldots c_{18} \\ c_{21} & c_{22} & \ldots c_{28} \\ c_{31} & c_{32} & \ldots c_{38} \\ c_{41} & c_{42} & \ldots c_{48} \end{pmatrix} \times \begin{pmatrix} x_{0+8\cdot\gamma} \\ x_{1+8\cdot\gamma} \\ \ldots \\ x_{7+8\cdot\gamma} \end{pmatrix} mod\ 256 = \begin{pmatrix} y_{0+4\cdot\gamma} \\ y_{1+4\cdot\gamma} \\ y_{2+4\cdot\gamma} \\ y_{3+4\cdot\gamma} \end{pmatrix} \qquad y_{\gamma 4\times 1},\quad \gamma = 0,1, \ldots, 63;$$

бўлиб, $T(1) = y_0 y_1 y_2 y_3 \ldots y_{0+4\cdot\gamma}, y_{1+4\cdot\gamma}, y_{2+4\cdot\gamma}, y_{3+4\cdot\gamma} \ldots y_{252} y_{253} y_{254} y_{255}$

ISSN NO: 2770-0003
Date of Publication: 10-06-2022

12. $i = i + 1$.
13. $i < 8$ шарт текширилади. Агар бу шарт бажарилса, $T(0) = T(1)$, $k_0 = k_{0+32\cdot i}, \ldots, k_{31} = k_{31+32\cdot i}$ деб олинади ва 6-қадамга ўтилади, акс ҳолда кейинги қадамга ўтилади.
14. $H = H \oplus T(1)$.
15. $n = n + 1$.
16. $n \leq N$ шарт текширилади. Агар бу шарт бажарилса, 2-қадамга ўтилади, акс ҳолда H нинг қиймати берилган M матннинг хэш қиймати бўлади.

The analysis of the cryptographic properties of the new keyless hash-function algorithm proposed in the article and the construction of the block diagram is based on the data on the keyless hash function and rectangular matrix compression reflections given in the literature [3, 4].

References

1. Alferov A. P., Zubov A. Yu., Kuzmin A. S., Cheremushkin A. V. Basic cryptography: Uchebnoe posobie, 2-e izd. –M .: Gelios ARV, 2002.-480 p.
2. Schnayer B. Applied cryptography. Protocols, algorithms, source text in the letter Si. –M .: izdatelstvo TRIUMF, 2003 - 816 p.
3. Akbarov D.E. Cryptographic methods of information security and their application - Tashkent, "Brand of Uzbekistan", 2009 - 434 pages.
4. Khasanov P.F., Akbarov D. E., Axmadaliev Sh. Sh. Methods for creating new asymmetric algorithms based on existing computational complexities using parametric algebra operations. // Infokommunikatsii: Seti-Texnologii-Resheniya. -1 (9) / 2009. -s. 31-35.