Original Research Article: full paper

RESEARCH OF COLLISION PROPERTIES OF THE MODIFIED UMAC ALGORITHM ON CRYPTO-CODE CONSTRUCTIONS

Serhii Yevseiev✉
Department of Cyber Security and Information Technology[1]
serhii.yevseiev@hneu.net

Alla Havrylova
Department of Cyber Security and Information Technology[1]

Olha Korol
Department of Cyber Security and Information Technology[1]

Oleh Dmitriiev
Department of Flight Operations, Aerodynamics and Flight Dynamics
Flight Academy of National Aviation University
1 Dobrovolskogo str., Kropyvnytskyi, Ukraine, 25005

Oleksii Nesmiian
Department of Mathematical and Software of Automated Control Systems
Ivan Kozhedub Kharkiv National Air Force University
77/79 Sumska str., Kharkiv, Ukraine, 61023

Yevhen Yufa
Department of Radio-Technical and Special Troops
National Defence University of Ukraine named after Ivan Cherniakhovskyi
28 Povitroflotskyi ave., Kyiv, Ukraine, 03049

Asadi Hrebennikov
Department of Technical Information Protection Systems
Information Security Institute
State University of Telecommunications
7 Solomenska str., Kyiv, Ukraine, 03110

[1]*Simon Kuznets Kharkiv National University of Economics*
9-A Nauky ave., Kharkiv, Ukraine, 61166

✉**Corresponding author**

Abstract

The transfer of information by telecommunication channels is accompanied by message hashing to control the integrity of the data and confirm the authenticity of the data. When using a reliable hash function, it is computationally difficult to create a fake message with a pre-existing hash code, however, due to the weaknesses of specific hashing algorithms, this threat can be feasible. To increase the level of cryptographic strength of transmitted messages over telecommunication channels, there are ways to create hash codes, which, according to practical research, are imperfect in terms of the speed of their formation and the degree of cryptographic strength. The collisional properties of hashing functions formed using the modified UMAC algorithm using the methodology for assessing the universality and strict universality of hash codes are investigated. Based on the results of the research, an assessment of the impact of the proposed modifications at the last stage of the generation of authentication codes on the provision of universal hashing properties was presented. The analysis of the advantages and disadvantages that accompany the formation of the hash code by the previously known methods is carried out. The scheme of cascading generation of data integrity and authenticity control codes using the UMAC algorithm on crypto-code constructions has been improved. Schemes of algorithms for checking hash codes were developed to meet the requirements of universality and strict universality. The calculation and analysis of collision search in the set of generated hash codes was carried out according to the requirements of a universal and strictly universal class for creating hash codes.

Keywords: UMAC, crypto-code constructions, hybrid crypto-code constructions, collision, method of versatility.

DOI: 10.21303/2461-4262.2022.002213

1. Introduction

When transmitting information over telecommunication channels, hashing of transmitted messages is used, which is usually carried out using manipulation detection codes (to control data integrity) and message authentication codes (to confirm the authenticity of data). When using a reliable hash function, it is computationally difficult to create a fake message with the same hash code (MAC code – message authentication code), as the original message. However, these threats can be realized due to the weaknesses of specific hashing algorithms, signatures, or errors in their implementations.

The existing hashing algorithms for verifying the authenticity of received messages when working in the post-quantum period do not have the necessary cryptographic resistance to hacking [1–4], so the problem arises of creating new algorithms or modifying existing ones. These cryptographic algorithms must have not only a higher degree of cryptographic strength, but also sufficient efficiency. Also, hashing algorithms that are resistant to hacking from quantum computers require large enough power resources and a large number of operations to calculate the hash code [1, 4–6].

One of the features of hash codes is the even distribution of collision properties, which allows the use of these MAC codes as identifiers of data contained in large databases (DB). This feature can increase the level of data security when working with them remotely.

There are a number of approaches to creating hash codes to increase the cryptographic strength of transmitted messages over telecommunication channels, but practical studies that allow comparing the resulting variants of MAC code, which suggest the imperfection of the proposed hashing options in terms of the speed of their formation and the degree of cryptographic strength.

Therefore, the aim of this research is to research the collisional properties of hashing functions generated using the modified message authentication code based on universal hashing UMAC (message authentication code based on universal hashing) algorithm based on the methodology for assessing universality and strict universality, as well as assessing the impact of the proposed modifications at the last stage of generating authentication codes on ensuring properties of universal hashing.

2. Materials and methods

In their works [7–9], the authors proposed a method for constructing multilayer hashing functions using the UMAC algorithm as an example. It is based on a combination of multi-stage key universal hashing and the use of a symmetric block cipher. This algorithm uses sets of universal hash functions and provides provable security for the generated authentication code. The use of a pseudo-random pad on the last layer also gives additional cryptographic resistance to the code. Thus, universal hashing in a multi-layered UMAC design allows providing the same probability of hash-images formation for the entire set of key data used. In work [10] it is indicated that the obtained property ensures the safety of the algorithm.

Consider the features of the formation of a pseudo-random pad by already known cryptographically strong algorithms:

1) symmetric block cipher algorithm AES (Advanced Encryption Standard);

2) modular transformation RSA (Rivest, Shamir, Adleman) algorithm using loop functions;

3) keyless algorithms MASH-1 and MASH-2 using modular transformations (**Table 1**).

Generation of a pseudo-random pad with a cryptographically strong algorithm of a symmetric block cipher AES ensures the cryptographic strength of the UMAC algorithm at the level of the applied cryptoalgorithm. High efficiency indicators of code generation using this method are justified by the use of the bitwise addition operation for imposing pseudo-random pads. But at the same time, the UMAC algorithm loses an important property of the universality of hashing of the generated MAC codes, which also leads to a decrease in its collisional properties. This disadvantage is associated with the features of block symmetric encryption.

Therefore, the authors in [10] proposed to carry out universal hashing using modular transformations. Crypto resistance in this case provided such a property of the received MAC-codes as the impossibility of decrypting them in a computable time. But, with the advent of high-performance quantum computers, this property will not become an obstacle to hacking.

The next option for the formation of a cryptographically strong MAC-code can be called universal hashing based on modular transformations using the RSA algorithm using elliptic curves and the computational complexity of the problem of factoring large numbers. To form the final layer, cryptographically strong functions of strict universal hashing are used on the modular arithmetic of exponentiation using cycle functions [10]. But, as for the previous method of generating a MAC-code, the use of this option is limited by a number of significant drawbacks (**Table 1**).

Table 1

Analysis of algorithms for the generation of a pseudo-random pad

Algorithms for the formation of pseudo-random pads	Advantages	Disadvantages
AES	+ high performance indicators; + inappropriateness of hacking when using modular transformations;	– there is no guarantee of preservation of properties of universality; – maintains cryptographic strength with existing computing power;
RSA	+ burglary resistance in existing conditions is high;	– does not provide efficiency; – low post-quantum cryptographic strength;
MASH-1	+ easy to implement;	– not cryptographically secure; – does not provide universality;
MASH-2	+ provides versatility and cryptographic strength	– low speed of hash-code generation; – not all values provide universality

An improved method for generating MAC-codes based on the UMAC algorithm, despite the fact that it provides a high speed of forming the first layers, but is cryptographically weak on the last layer. Therefore, it was proposed [10] to strengthen it using a strict universal hashing scheme based on modular transformations using keyless algorithms MASH-1 and MASH-2. Hash codes generated in this way meet the properties of universality and maintain a given level of cryptographic strength, but only for a given level of computing power. Also, for using the MASH-1 algorithm, there are restrictions on the values of the exponentiation values – these are only even numbers. Despite the fact that when using the MASH-2 algorithm, both even and odd numbers can be used as exponentiation numbers, this option is not suitable for all primes specified at the beginning of the hashing procedure. All this leads to the fact that, despite the provision of universality and a given strength of the generated MAC-codes, the practical implementation has a low rate of generation of a pseudo-random pad.

With the advent of quantum computers, all of the options considered lose their level of cryptographic strength, since attackers can carry out Grover and Shor attacks, which make it possible to break ciphers according to the presented algorithms in a computable time [4, 6].

According to the NIST competition, the algorithm for forming an asymmetric block cipher with a McEliece public key on elliptic curves has reached the third round and is a candidate for post-quantum cryptography, since it is resistant to attacks using Shor's algorithm. The cryptographic strength of the authentication code, which is generated by this algorithm, can be enhanced by using a pseudo-random pad. This pseudo-random pad can be represented as a crypto-code construct (CCC) on elliptic curves.

Thus, to eliminate the identified drawback in already existing algorithms, it is proposed to use CCC on elliptic curves (EC), modified elliptic curves (MEC) McEliece and hybrid crypto-code constructions (HCCC) on defective codes (DC) [1] (**Fig. 1**).

In the proposed method for generating data integrity and authenticity control codes, the first transformation layers are proposed to be implemented with high-speed, but cryptographically weak universal hashing schemes traditional for the UMAC algorithm, and the last layer is proposed to be implemented using the developed secure (cryptographically strong) strictly universal hashing scheme based on algebraic geometric codes CCC. This algorithm for generating a hash code is cryptographically resistant to hacking in the post-quantum period, since the technical characteristics of quantum computers will only slightly reduce the number of operations required to crack this algorithm [5].

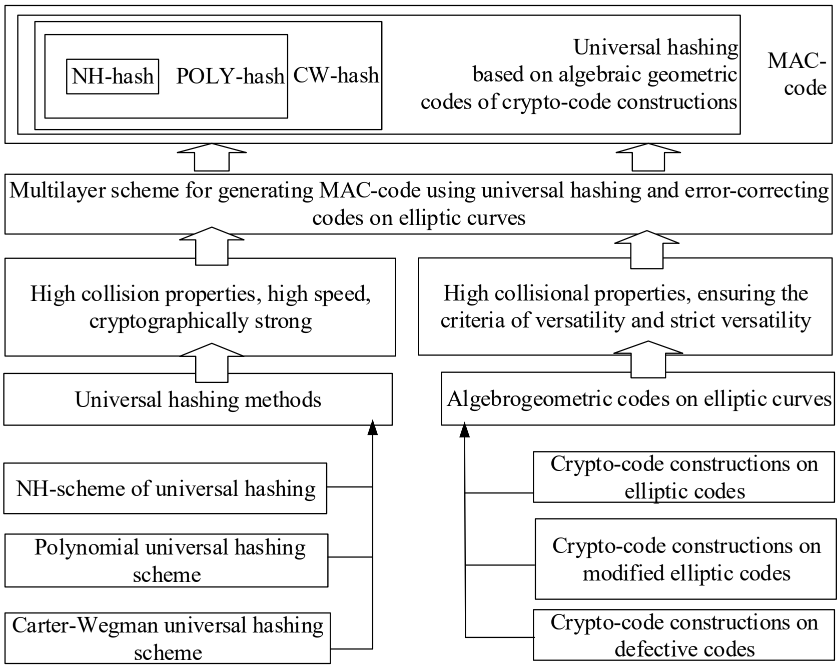
NH-hash
POLY-hash
CW-hash
Universal hashing based on algebraic geometric codes of crypto-code constructions
MAC-code
Multilayer scheme for generating MAC-code using universal hashing and error-correcting codes on elliptic curves
High collision properties, high speed, cryptographically strong
High collisional properties, ensuring the criteria of versatility and strict versatility
Universal hashing methods
Algebrogeometric codes on elliptic curves
NH-scheme of universal hashing
Polynomial universal hashing scheme
Carter-Wegman universal hashing scheme
Crypto-code constructions on elliptic codes
Crypto-code constructions on modified elliptic codes
Crypto-code constructions on defective codes

Fig. 1. Scheme of cascading generation of data integrity and authenticity control codes using the UMAC algorithm on the CCC

The pseudo-random pad can be represented by varieties that should equally ensure that the necessary transformations are performed and that the properties of universality are preserved by the UMAC algorithm.

3. Results

3. 1. Development of a practical algorithm for the implementation of the universality methodology and strict universality methodology of the generation of hash codes

In [1, 10], it was proposed to use the universality methodology and strict universality methodology to check hash codes for their collisional properties. Using this methodology, it is proposed to check hash codes generated using the modified UMAC algorithm. A formalized presentation of this methodology allows to develop a practical algorithm for its implementation. Let's consider schemes for checking hash codes for the possibility of collisions based on universality and strict universality criteria.

The algorithm for checking hash codes to comply with the rules of the universal class of hash functions is shown in **Fig. 2**.

The description of the algorithm for checking hash codes to meet the requirements of the universal class of hash functions consists in the implementation of the following steps of its implementation:

Step 1. Generating input messages $I_1, I_2,\ldots, I_j \in I$.

Step 2. Keys are generated $K_1, K_2,\ldots, K_j \in K$.

Step 3. For each input message I_i, using the keys K_j, their hash codes H_{ij} are generated.

Step 4. Conduct a sequential comparison of the received hash codes H_{ij} for the same key K_j for all input messages with each other based on the following condition: if the hash values coincide ($H_{ij}=H_{ij+1}$), which indicates a collision (L_j), then 1 is added to the collision counter:

$$L_j = \sum_{j=1}^{N} L_{j-1} + 1, \tag{1}$$

where L_{j-1} the previous value of the collision counter for the j key.

Step 5. Within one message for all keys, select the maximum collision value $L\max_i$.

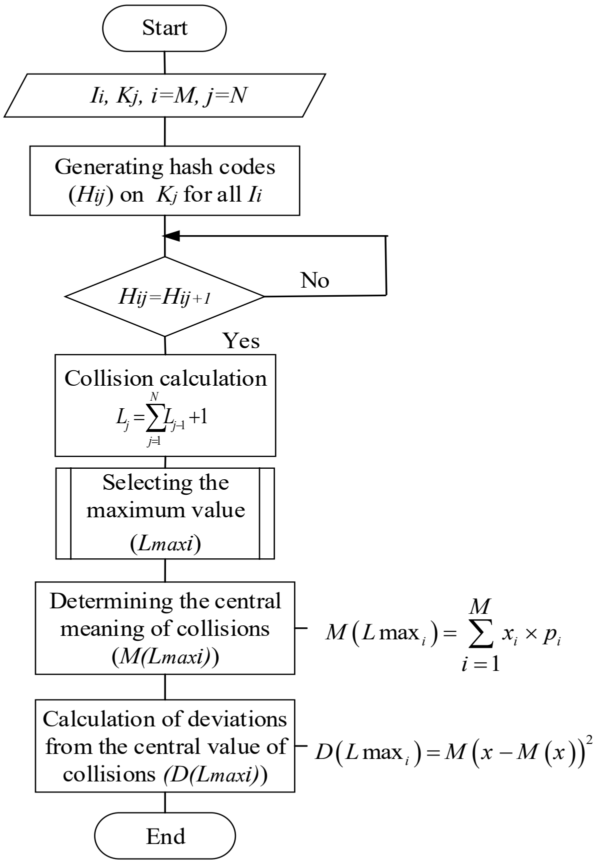
Start
Ii, Kj, i=M, j=N
Generating hash codes (Hij) on Kj for all Ii
Hij=Hij+1
No
Yes
Collision calculation $L_j=\sum_{j=1}^{N} L_{j-1}+1$
Selecting the maximum value (Lmaxi)
Determining the central meaning of collisions (M(Lmaxi))
$M(L\max_i)=\sum_{i=1}^{M} x_i \times p_i$
Calculation of deviations from the central value of collisions (D(Lmaxi))
$D(L\max_i)=M(x-M(x))^2$
End

Fig. 2. Algorithm for checking hash codes to meet the requirements of the universal class of hash functions

Step 6. Calculate the arithmetic mean or central value of the maximums of the number of collisions by finding their mathematical expectation $M(L\max_i)$:

$$M(L\max_i)=\sum_{i=1}^{M} x_i \times p_i. \tag{2}$$

Step 7. Calculate the value of the spread of the maximum collision values around their central value by finding the variance ($D(L\max_i)$) for the maximums of the number of collisions:

$$D(L\max_i)=M(x-M(x))^2. \tag{3}$$

The algorithm for checking hash codes to meet the requirements of a strictly universal class of hash functions according to the first criterion is shown in **Fig. 3**.

The description of the algorithm for checking hash codes to meet the requirements of a strictly universal class of hash functions according to the first criterion consists in the implementation of the following steps of its implementation:

Step 1. Generating one random input message I_{random}.

Step 2. Generating the hash code H_{random} of a random message I_{random}.

Step 3. Generating input messages $I_1, I_2,\ldots, I_j \in I$.

Step 4. Keys are formed $K_1, K_2,\ldots, K_j \in K$.

Step 5. For each input message I_i, using the keys K_j, their hash codes H_{ij} are generated.

Step 6. Conduct a sequential comparison of the received hash codes H_{ij} for the same key K_j for all input messages with each other based on the following condition: if the hash values coincide ($H_{ij}=H_{ij+1}$), which indicates a collision (L_j), then 1 is added to the collision counter (1).

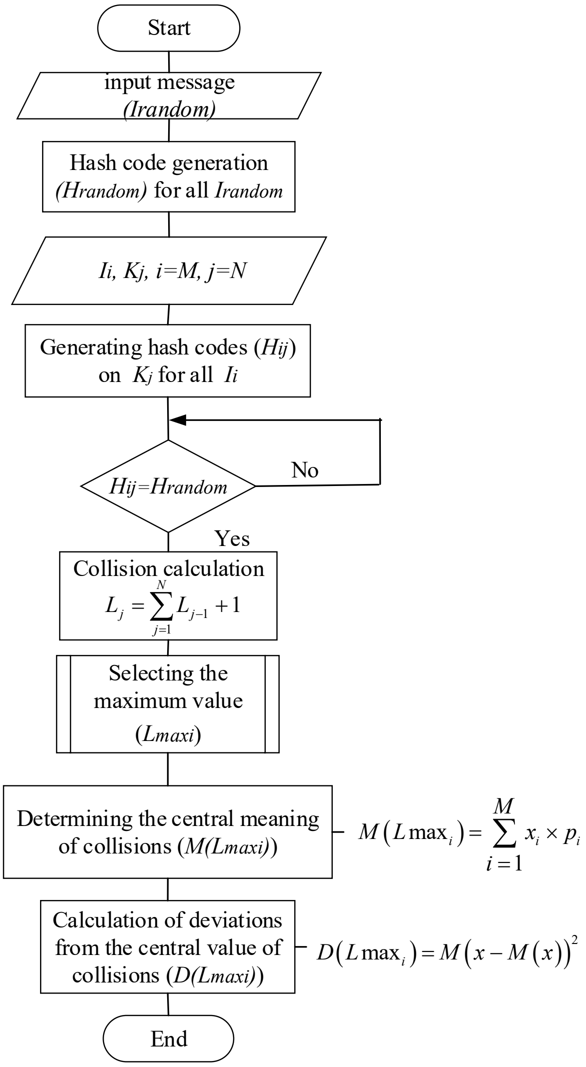
Start
input message (Irandom)
Hash code generation (Hrandom) for all Irandom
Ii, Kj, i=M, j=N
Generating hash codes (Hij) on Kj for all Ii
Hij=Hrandom
No
Yes
Collision calculation $L_j=\sum_{j=1}^{N}L_{j-1}+1$
Selecting the maximum value (Lmaxi)
Determining the central meaning of collisions (M(Lmaxi))
$M(L\max_i)=\sum_{i=1}^{M}x_i\times p_i$
Calculation of deviations from the central value of collisions (D(Lmaxi))
$D(L\max_i)=M(x-M(x))^2$
End

Fig. 3. Algorithm for checking hash codes to meet the requirements of a strictly universal class of hash functions according to the first criterion

Step 7. Within one message for all keys, select the maximum collision value $L\max_i$.

Step 8. Calculate the arithmetic mean or central value of the maximums of the number of collisions by finding their mathematical expectation $M(L\max_i)$ (2).

Step 9. Calculate the value of the spread of the maximum collision values around their central value by finding the variance ($D(L\max_i)$) for the maximums of the number of collisions (3).

The algorithm for checking hash codes to meet the requirements of a strictly universal class of hash functions according to the second criterion is shown in **Fig. 4**.

The description of the algorithm for checking hash codes to meet the requirements of a strictly universal class of hash functions according to the second criterion consists in the implementation of the following steps of its implementation:

Step 1. Generating two different random input messages $I_{random1}$ and $I_{random2}$.

Step 2. Generating hash codes $H_{random1}$ and $H_{random2}$ for each of the messages $I_{random1}$ and $I_{random2}$.

Step 3. Generating input messages $I_1, I_2,\ldots, I_j \in I$.

Step 4. Keys are formed $K_1, K_2,\ldots, K_j \in K$.

Step 5. For each input message I_i, using the keys K_j, their hash codes H_{ij} are generated.

Step 6. Conduct a sequential comparison of the received hash codes $H_{random1}$ and $H_{random2}$ for the same key K_j for all input messages with hash codes of two random messages: if the hash values coincide ($H_{ij} = H_{ij+1}$), which indicates a collision (L_j), then 1 is added to the collision counter (1).

Step 7. Within one message for all keys, select the maximum collision value $L\max_i$.

Step 8. Calculate the arithmetic mean or central value of the maximums of the number of collisions by finding their mathematical expectation $M(L\max_i)$ (2).

Step 9. Calculate the value of the spread of the maximum collision values around their central value by finding the variance ($D(L\max_i)$) for the maximums of the number of collisions (3).

Thus, the proposed schemes of algorithms make it possible to evaluate not only the fulfillment of the criteria for universality and strict universality of the obtained MAC-code, but also the level of security.

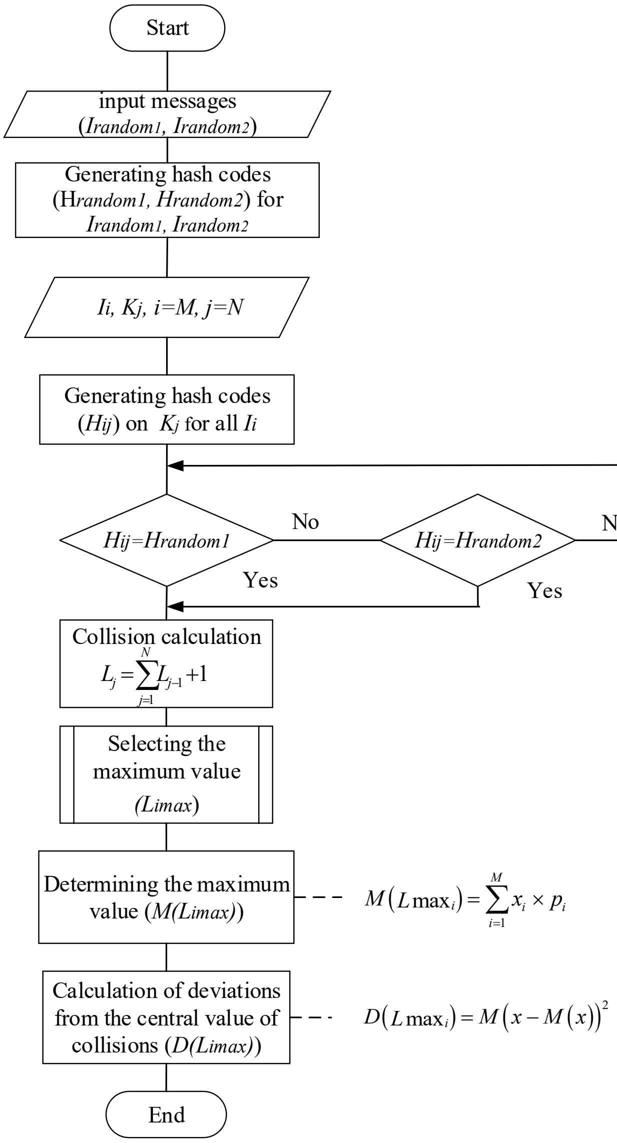

Fig. 4. Algorithm for checking hash codes to meet the requirements of a strictly universal class of hash functions according to the second criterion

3. 2. Software implementation of the algorithm for the occurrence of collisional properties of hash codes

An application that implements collision search in a set of hash codes generated using a modified UMAC algorithm was developed in an object-oriented programming language C#.

Using the reduced UMAC model (mini-UMAC), let's conduct a research of the collision properties of message authentication codes, which consists in an experimental assessment of the distribution of the number of collisions of the generated images. Reduced models are designed to investigate the main indicators of the efficiency of the cryptoalgorithm while maintaining its algebraic structure [10].

Since in the above UMAC scheme on the first layer (when generating a hash code) multiplicity of universal hashing functions are used, which are studied in detail in [1, 6–10], let's carry out statistical research only on the second layer when forming a pseudo-random pad and at the final stage of generating authentication codes (after completing the summation).

It is at these stages, according to the authors' assumption [10], that the properties of the universality of the generated authentication codes are violated. When conducting statistical research of the collisional properties of the generated values of hash codes for each experiment, the mathematical expectations $m(n_1)$, $m(n_2)$ and $m(n_3)$, variances $D(n_1)$, $D(n_2)$ and $D(n_3)$ were estimated, as well as for a confidence level $P_{confid}(|\tilde{m}(n_i) - m(n_i)| < \varepsilon) = 0.98$ were determined based on the calculated accuracies $\varepsilon_1 = t_{kp}(q(n_1)/\sqrt{n})$, $\varepsilon_2 = t_{kp}(q(n_2)/\sqrt{n})$ and $\varepsilon_3 = t_{kp}(q(n_3)/\sqrt{n})$ the corresponding extrema (lower and upper values) of the confidence intervals $(\tilde{m}(n_i) - \varepsilon; \tilde{m}(n_i) + \varepsilon)$. And besides $\tilde{m}(n_i)$ it is a natural estimate for the mathematical expectation mn_i of a random variable n_i, and $\tilde{D}(n_i)$ is an estimate of the variance of the random variable n_i.

The research was carried out on a sample of size $N = 10\ 000$ elements. To form each element of the sample, the maximum was calculated over a set of $M = 1\ 000$ tuples of elements. Thus, the total volume of the generated sets was $N{\cdot}M = 10^7$.

The obtained results of experimental studies are summarized in **Table 2**.

Table 2

Results of experimental researches of collisional properties of authentication codes generated using MASH1, MASH2, mini-UMAC MASH1, mini-UMAC MASH2, mini-UMAC AES and mini-UMAC CCC (at $P_{confid} = 0.98$)

Statistical characteristics of the experiment	MASH-1	MASH-2	mini-UMAC MASH-1	mini-UMAC MASH-2	mini-UMAC AES	mini-UMAC KKK
$\tilde{m}(n_1)$	7.09*	7.14*	1.965	1.968	1.096	1.166
$\tilde{m}(n_1)$	1.69	1.56	0.123	0.120	0.094	0.599
$\tilde{m}(n_1) - \varepsilon_1$	7.061	7.111	1.957	1.961	1.088	1.148
$\tilde{m}(n_1) + \varepsilon_1$	7.122	7.169	1.973	1.977	1.103	1.184
$\tilde{m}(n_2)$	1.013	1.014	2.629*	2.64*	1.532	1.161
$\tilde{D}(n_2)$	0.013	0.014	0.349	0.355	0.36	0.67
$\tilde{m}(n_2) - \varepsilon_2$	1.01	1.011	2.62	2.63	1.52	1.14
$\tilde{m}(n_2) + \varepsilon_2$	1.02	1.017	2.64	2.65	1.55	1.18
$\tilde{m}(n_3)$	1.0008	1.0002	0.237**	0.224**	0.0005**	0**
$\tilde{D}(n_3)$	$9993\cdot10^{-8}$	$9999\cdot10^{-8}$	0.184	0.177	$499\cdot10^{-6}$	0
$\tilde{m}(n_3) - \varepsilon_3$	1.00006	0.9994	0.227	0.214	–2.087	0
$\tilde{m}(n_3) + \varepsilon_3$	1.002	1.0009	0.247	0.234	0.001	0

** – natural estimates of mathematical expectations, according to which the number of collision values significantly exceeds their theoretical estimates;*

*** – natural estimates of mathematical expectations, according to which the number of collision values does not exceed their theoretical estimates.*

4. Discussion

Let's analyze the obtained results of statistical research of the collision properties of message authentication codes: let's compare the obtained results of the average estimates of the mathematical expectations $m(n_1)$, $m(n_2)$ and $m(n_3)$ of the number of hashing rules with theoretical estimates (with the number $P_{coll}{\cdot}|H|$ – for the first criterion, with the number $|H|/|B|$ – for the second criterion and the number $P_{coll}{\cdot}H$ – for the third criterion), in this case, equalities (5)–(7) in [10] must be satisfied, respectively.

Consider the first criterion by which the number of hashing rules is estimated for which there is a collision (coincidence of authentication codes) for two arbitrary input sequences. In accordance with theoretical estimates, this quantity is bounded from above by the number $P_{coll}{\cdot}|H|$. Let's concretize this (theoretical) estimate for authentication codes generated using the algorithms MASH-1, MASH-2, mini-UMAC MASH-1, mini-UMAC MASH-2, mini-UMAC AES, and mini-UMAC CCC.

The power of the key set for the algorithms MASH-1, MASH-2, mini-UMAC MASH-1, mini-UMAC MASH-2, mini-UMAC AES and mini-UMAC CCC is $|H|=2^{16}$, the cardinality of the set of generated authentication codes is also $|B|=2^{16}$. If to use the upper bound for the probability of collisions as the reciprocal of the power of the generated authentication codes $P_{coll}=2^{-16}$, let's obtain $n_1(x_1, x_2) \le P_{coll}{\cdot}|H| = 1$ [10].

The collisional properties of the MASH-1 and MASH-2 encryption algorithms are significantly inferior to this upper theoretical estimate. In fact, the number of collisions on them is more than 7 times higher than the theoretical limit. Codes generated by all other algorithms also do not meet the first universality criterion, since the number of collisions exceeds the specified limit. Consequently, the criterion of universality is not satisfied by any of the algorithms.

Let's consider the second criterion by which the number of hashing rules is estimated under which the value of the authentication code does not change for an arbitrary input sequence. In accordance with theoretical estimates, this value for authentication codes generated using all algorithms is bounded from above by the number $|H|/|B| = 1$ [10].

The experimental results obtained indicate that the collision properties of authentication codes generated using the mini-MASH-1 and mini-MASH-2 algorithms do not satisfy the second criterion, since the number of collisions for them exceeds the theoretical limit by almost 3 times, and for all the rest algorithms also observe an excess of the permissible number of collisions. Consequently, the first criterion of strict universality is not satisfied by any of the algorithms.

In accordance with the third criterion, the number of hashing rules is estimated under which the corresponding values of the authentication code do not change for two arbitrary input sequences. The theoretical estimate of this value for universal hashing is bounded from above by the number $P_{coll}{\cdot}|H|$, which when using the upper bound for the collision probability $P_{\mathbf{coll}}=2^{-16}$ let's obtain $n_3(x_1, x_2, y_1, y_2) \le P_{coll}{\cdot}|H| = 1$ [10].

The values given in **Table 1** indicate that the collisional properties of authentication codes generated using mini-MASH-1, mini-MASH-2, mini-UMAC AES, and mini-UMAC CCC satisfy the second criterion of versatility. The data obtained by calculation allows to assert that the use of the UMAC scheme on the CCC significantly improves the collision properties of the integrity control and authentication codes [11].

The elliptical form of hash codes when using post-quantum cryptography does not provide the required level of their cryptographic strength. Therefore, to form more cryptographically strong hash codes, it is necessary to use modified elliptical codes. Moreover, a higher level of cryptographic strength of authentication codes can be increased through the use of hybrid crypto-code constructions.

5. Conclusions

The analysis of algorithms for the formation of the final layer of authentication codes, namely, a pseudo-random pad, in the course of which the advantages and disadvantages accompanying the formation of a hash code by already known methods were identified. These algorithms are not cryptographically resistant to hacking in the post-quantum period.

It is proposed to use different modifications of McEliece elliptic codes (EC, MEC, DC) on crypto-code constructions and hybrid crypto-code constructions to increase the cryptographic strength of MAC-codes.

Schemes of algorithms for checking hash codes to meet the requirements of universality and strict universality according to the first and second criteria have been developed in accordance with the requirements of universality and strict universality of hash functions.

The results of hashing messages were obtained and analyzed using a software application that searches for collisions in a set of hash codes: hash codes generated using the mini-UMAC MASH-1 ($\tilde{m}(n_3)=0.237$), mini-UMAC MASH-2 ($\tilde{m}(n_3)=0.224$), mini-UMAC AES ($\tilde{m}(n_3)=0.0005$) and mini-UMAC CCC ($\tilde{m}(n_3)=0$). And besides the use of the UMAC scheme on the CCC significantly improves the collision properties of the integrity control and authentication codes, because the value $\tilde{m}(n_3)$<1.

References

[1] Gavrilova, A., Volkov, I., Kozhedub, Y., Korolev, R., Lezik, O., Medvediev, V. et. al. (2020). Development of a modified UMAC algorithm based on cryptocode constructions. Eastern-European Journal of Enterprise Technologies, 4 (9 (106)), 45–63. doi: https://doi.org/10.15587/1729-4061.2020.210683

[2] PQC Standardization Process: Third Round Candidate Announcement (2020). Available at: https://csrc.nist.gov/News/2020/pqc-third-round-candidate-announcement

[3] Xia, L., Yu, X. H., Han, J. (2015). Design of motion control system of industrial robot based on UMAC. Journal of Hefei University of Technology (Natural Science), 38 (8), 1009–1012.

[4] Gorbenko, Y., Svatovskiy, I., Shevtsov, O. (2016). Post-quantum message authentication cryptography based on error-correcting codes. 2016 Third International Scientific-Practical Conference Problems of Infocommunications Science and Technology (PIC S&T). doi: https://doi.org/10.1109/infocommst.2016.7905333

[5] Grimes, R. A. (2020). Cryptography Apocalypse: Preparing for the Day When Quantum Computing Breaks Today's Crypto. John Wiley & Sons, Inc. doi: https://doi.org/10.1002/9781119618232

[6] Wang, L.-J., Zhang, K.-Y., Wang, J.-Y., Cheng, J., Yang, Y.-H., Tang, S.-B. et. al. (2021). Experimental authentication of quantum key distribution with post-quantum cryptography. Npj Quantum Information, 7 (1). doi: https://doi.org/10.1038/s41534-021-00400-7

[7] Krawczyk, H., Rogaway P. (2000). UMAC: Message authentication code using universal hashing. Available at: https://datatracker.ietf.org/doc/html/draft-krovetz-umac-00

[8] Krovetz, T. (2006). UMAC: Message Authentication Code using Universal Hashing. doi: https://doi.org/10.17487/rfc4418

[9] Krovetz, T. (2004). UMAC: Message Authentication Code using Universal Hashing. Available at: https://datatracker.ietf.org/doc/html/draft-krovetz-umac-02.html

[10] Yevseiev, S. P., Yokhov, O. Yu., Korol, O. H. (2013). Heshuvannia danykh v informatsiynykh systemakh. Kharkiv: Vyd. KhNEU, 312. Available at: http://www.repository.hneu.edu.ua/jspui/handle/123456789/6813

[11] Yevseiev, S., Ponomarenko, V., Laptiev, O., Milov, O., Korol, O., Milevskyi, S. et. al.; Yevseiev, S., Ponomarenko, V., Laptiev, O., Milov, O. (Eds.) (2021). Synergy of building cybersecurity systems. Kharkiv: PC TECHNOLOGY CENTER, 188. doi: https://doi.org/10.15587/978-617-7319-31-2

Received date 21.10.2021
Accepted date 03.12.2021
Published date 10.01.2022

How to cite: *Yevseiev, S., Havrylova, A., Korol, O., Dmitriiev, O., Nesmiian, O., Yufa, Y., Hrebennikov, A. (2022). Research of collision properties of the modified UMAC algorithm on crypto-code constructions. EUREKA: Physics and Engineering, 1, 34–43. doi: https://doi.org/10.21303/2461-4262.2022.002213*