DOI 10.52209/1609-1825_2022_4_449 UDC 004.056.5

Analysis of Hash Functions and Their Application in Electronic Digital Signature

[1,2]**USSATOVA Olga,** *PhD, Senior Researcher, uoa_olga@mail.ru,*
[2,3]***BEGIMBAYEVA Yenlik,** PhD, Senior Researcher, enlik_89@mail.ru,*
[4]**USSATOV Nikita,** *student, usatov.nikita2242@gmail.com,*
[1]*NPJSC «Almaty University of Power Engineering and Telecommunications named after Gumarbek Daukeyev», Kazakhstan, Almaty, A. Baitursynova Street, 126/1,*
[2]*Institute of Information and Computational Technologies, Kazakhstan, Almaty, Shevchenko Street, 28,*
[3]*NCJSC «Kazakh National Research Technical University named after K.I. Satpayev», Kazakhstan, Almaty, Satpayev Street, 22a,*
[4]*Turan University, Kazakhstan, Almaty, Satpayev Street, 16a,*
**corresponding author.*

Abstract. *The hash function used is known for implementing algorithmic solutions in programming. Hash functions encrypt and optimize work with processed and stored data, as well as in various operating systems, ranking data to ensure their integrity. The application of the hash function is quite extensive. The usage of the hash function allows the possibility to solve almost all problems of protecting electronic information, from ensuring the authenticity of subjects and objects of information interaction to introducing uncertainty into the operation of means and objects of protection. The article deals with hash functions used in electronic digital signature (EDS) algorithms. Modern hashing methods and the areas in which they are applicable are described. The hash function is used to secure data. These functions can vary in bit depth, complexity, and cryptographic strength. Cryptographic hashing algorithms are given parameters, structures, and methods, as well as their scope. The work is devoted to analyzing hash functions and their application in the electronic digital signature. The article presents the results of calculating collisions of hashing algorithms, allowing you to choose the most optimal option for its application in the algorithm under consideration. The practical application of the hashing algorithm for the digital signature system, which can be used in systems and networks to transmit and store information, is considered.*

Keywords: *information security, hash function, collision, cryptographic hash function, electronic digital signature, asymmetric cryptosystems, data protection, key generation, attack, hash algorithms.*

Introduction

Hashing has various applications in computer science and algorithmic solutions, specifically for encrypting and optimizing data operations, manipulating files in operating systems, and personalizing data to ensure their integrity. The field of application of the hashing mechanism significantly varies: in the implementation of dictionary systems, building indexes in databases, building name tables in compilers, speeding up working with files in operating systems, organizing lists of links in browsers, organizing dictionaries in translation programs from one language to another, designing electronic signatures of electronic documents, implementation of basic operations in extensive data analysis systems and decision-making systems. Hashing successfully solves almost all tasks of protecting computer information, from ensuring the authenticity of subjects and objects of information interaction to introducing uncertainty into the operation of means and objects of protection. For data, reconciliation hashing allows verification of information for identity without original data.

The hash function for electronic digital signatures

In the international standard ISO / IEC 14888-1-2008 [1], the following definitions are given: electronic digital signature (EDS): a bit string obtained as a result of the signature generation process; the process of forming a signature: a process, the initial data of which are the message, the signature key and the parameters of the EDS scheme, and as a result a digital signature is generated; hash code: a string of bits that is the output of the hash function; hash function: a function that maps strings of bits to strings of bits of fixed length. The hash function should satisfy the following properties: for a given value of the function, it is difficult to calculate the initial data mapped to this value; for the given initial data, it is challenging to figure out other initial data mapped to the same function value; it is difficult to calculate any pair of

initial data mapped to the same value. Signature key and signature verification key definitions are also given in the standard ISO / IEC 14888-1-2008. A signature key is an element of personal data specific to the subject and used only by this subject in generating a digital signature. The signature verification key is a data item mathematically associated with a signature key and used by a relying party in verifying a digital signature. The electronic digital signature system consists of three parts. The first part is the algorithm for generating a key pair for the signature and its verification, the second part is the function of forming a signature, and the last one is the signature verification function. Generating the signature based on the document and the user's secret key calculates the signature itself.

Several schemes for constructing a digital signature are based on symmetric encryption algorithms and asymmetric encryption algorithms. This scheme based on symmetric encryption algorithms provides for the presence of a third party in the system, an arbitrator whom both parties trust. Document authorization is because of its encryption with a secret key and its transfer to the arbiter. Symmetrical circuits are less common than asymmetric ones. Symmetric ciphers are based on block ciphers. The strength of symmetric schemes derives from the strength of the block ciphers used, the reliability of which is well understood. If the strength of the cipher turns out to be insufficient, it can be easily replaced with a more secure one with minimal changes in the implementation. Disadvantages of symmetric schemes: It is necessary to sign separately each bit of the transmitted information, which leads to a significant increase in the signature. The signature can be up to two orders of magnitude larger than the message. The keys generated for signing can only be used once since, after signing, half of the secret key is revealed [2]. EDS schemes based on asymmetric encryption algorithms are the most common and are widely used [3-4]. This popularity is due to the main advantage of asymmetric schemes over symmetric ones – only one party knows the secret key, so there is no need to send the sender's private key over a secure channel.

Two types of algorithms are used to construct digital signatures: the Cryptographic Hash Functions Algorithms and Digital Signature Algorithms [5]. The use of hash algorithms for EDS is considered. Figure 1 illustrates the general scheme of a digital signature. Since the documents to be signed have a variable volume, in EDS schemes, the signature is often placed not on the document itself but on the hash. Cryptographic hash functions are used to calculate the hash, which ensures that document changes are detected when the signature is verified. Hash functions are not part of the EDS algorithm, so any reliable hash function can be used in the scheme [6].

There is a huge list of Hash-Algorithms available with us to provide dedicated services (i.e., security) depending on how complex it is to break. A few renowned names are MD5, Blake, CRUSH, Grøstl, HAIFA, JH, Lake, SHA, Skein, Whirlpool etc., Keyed/ Un-keyed constructions having their own justifiable capabilities [7]. The best-known hashing algorithms are Message Digest 4 (MD4), Message Digest 5 (MD5), and Secure Hash Algorithm (SHA) [8]. Hash functions MD4 and MD5 generate fixed-length hash values of 128 bits. The American SHA hashing standard produces hash values that are 160 bits long. Compares cryptographic hashing algorithms are given in table.

The term Cryptographic-Hash-Function [9] has been used in Computer Science and IT, which refers to a function that compresses a message m of arbitrary length to a message of fixed length h,

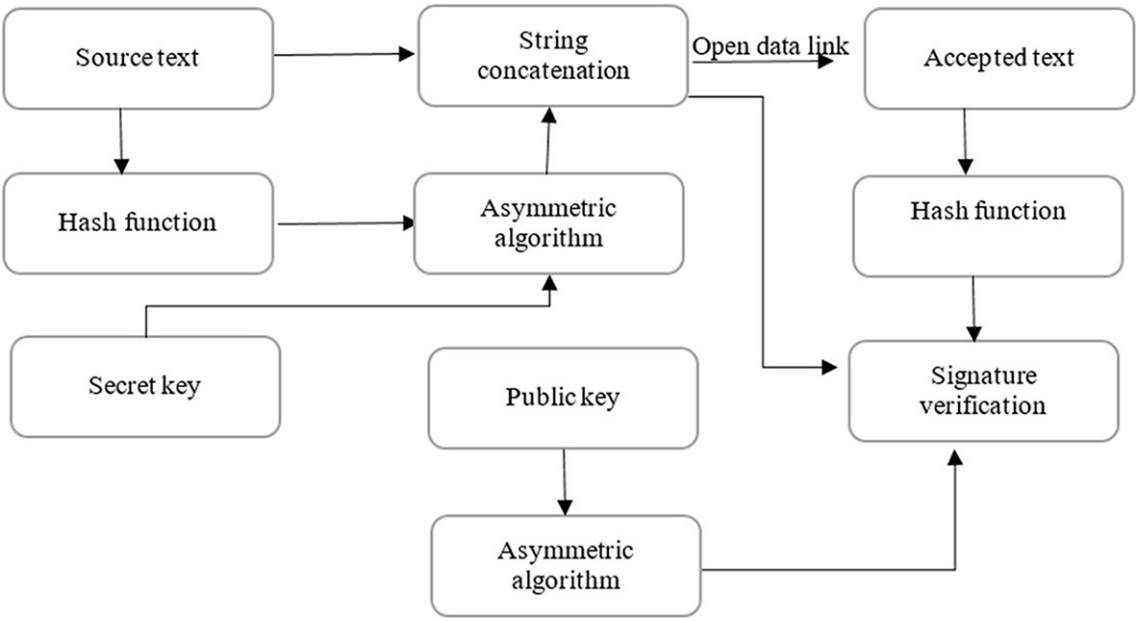

Figure 1 – General scheme of a digital signature

Comparison of cryptographic hash algorithms							
Hash functions	**Word length (bits)**	**Block length (bits)**	**Internal state length (bits)**	**Hash size (bit)**	**Number of rounds**	**Structure, method**	**Application area**
MD4	32	512	128 4×32	128	3 48 step	Merkle-Damgard structure.	Used in the MS-CHAP authentication protocols developed by Microsoft to perform authentication procedures for remote Windows workstations. It is the predecessor of MD5.
N-Hash	32		512 4×128	128	12/15	FEAL Block Encryption	Designed to solve the problem of information substitution on the way between two telephone users and speed up data retrieval.
SHA-1	32	512	160 5×32	160	80	Merkle-Damgard structure	Used in many cryptographic applications and protocols. Also recommended as a primary for government agencies in the US.
MD5	32	512	128 4×32	128	4 64 step	Merkle-Damgard structure	Designed to create «fingerprints» or message digests of arbitrary length and then verify their authenticity. It was widely used to check the integrity of information and store password hashes.
PANAMA	32	Stream based	544 17×32	Random	–	It is based on a finite state machine, consisting of two large blocks: 544 state bits and an 8192-bit buffer that works like a feedback shift register.	It was supposed to be used in encrypting or decrypting video to access some applications, in particular, «pay-TV». The developers' logic was that set-top boxes and digital TVs would use high-speed processors, and Panama would not load these processors too much during decryption.
SHA-512, SHA-384, SHA-512/256, SHA-512/224	64	1024	512 8×64	512/384/256/224	80	Merkle-Damgard structure.	US law allows for use in certain government applications, including use in other cryptographic algorithms and protocols, to protect unclassified information.
SHA-3 (Keccak)	64	1600		224/256/384/512	24	Built on the principle of a cryptographic sponge (Sponge)	Was designed to be very efficient in hardware but relatively slow in software.
STREEBOG	512	512	512	256/512	12		They are used in cryptographic methods of information processing and protection, including for the implementation of procedures for ensuring the integrity, authenticity, electronic digital signature (EDS) during the transmission, processing and storage of information in automated systems.

where h is message Digest. However, if it satisfies some additional requirements, it can be used for cryptographic applications and then known as Cryptographic Hash functions. $h=H(m)$. Where H: $\{0,1\}^* \rightarrow \{0,1\}^n$, where * is arbitrary length, n is fixed length.

One-way Hash Functions defined by Merkle, which is the base for Secure Hash Standards (SHS) evolved by NIST [9], i.e., a hash function H, must satisfy the following requirements:

1. H can be applied to Block of data of any length.
2. H produces a fixed-length output.
3. Given H and x, it is easy to computer Message Digest $H(x)$.
4. Given H and $H(x)$, it is computationally infeasible to find x.
5. Given H and $H(x)$, it is computationally infeasible to find x and x'' such that $H(x)=H(x'')$.

The first three requirements are a must for practical applications of a cryptographic hash function for «Message Authentication» and «Digital-Signatures».

In the second case, digital signature algorithms can be created using asymmetric ciphers. Moreover, there are two possible approaches to building a digital signature system. The second approach is that the digital signature is generated separately and sent along with the original message. EDS creation is based on the interconnection of the message content, the signature itself, and the key pair. When generating an EDS, the secret key of the sender of the message is used, and when verifying it, the public key of the sender is used. Regardless of the algorithm used, the general EDS scheme in the asymmetric encryption system can be represented as follows.

In the first step, the hash function h of the transmitted message M is calculated and a hash of the message $m=h(M)$ is generated. Using the A sender's secret key A_k and the EDS E generation algorithm, an encrypted hash value $C(m)$ of the message M is created. The package for the recipient B includes: message M, EDS $C(m)$ and the sender's public key K_A. The package B is transmitted to the recipient via an open communication channel. To verify the signature, the recipient B calculates the hash function $h(M)$ and receives a hash m', decrypts the EDS with the sender's public key, and receives the hash value m'. Next, a comparison is made between two hash functions m' and m. If these values match, then the signature is considered authentic. Otherwise the signature is rejected.

Application of hashing algorithms

Hash functions (hashing algorithms) successfully solve almost all problems of protecting computer information: ensuring the authenticity of subjects and objects of information interaction; formation of information integrity control codes; formation of an electronic digital signature; password systems of access control; authentication protocols for remote subscribers, etc. Hash functions are used for: the creation of electronic signatures, storing passwords in databases of security systems; within the framework of modern cryptography to generate unique keys online; checking the authenticity and integrity of the elements of the PC file system.

The practical application of the hashing algorithm for an asymmetric digital signature system, which can be used in systems and networks to transmit and store information, is considered. Figure 2 shows practical applications of the hash function in the digital signature algorithm. First, let's consider the application of the hash function in the asymmetric algorithm. Below the algorithm for the formation and verification of the digital signature is given:

Formation of EDS	EDS verification
1. starting the program;	1. starting the program;
2. enter the email to be signed;	2. entering a signed message and EDS;
3. key generation;	3. calculate the hash value;
4. calculate the hash value;	4. calculation of EDS;
5. calculation of eds according to the developed algorithm;	5. comparison of the calculated value of the EDS with the value of the received EDS;
6. sending a signed message;	6. displays a message about whether the EDS is genuine;
7. end of work.	7. end of work.

A multi-tiered complex cryptographic algorithm and additional security measures for protecting the communication channel are used to protect data. Many programming and server-side languages use special classes and functions that calculate hashes without difficulty but still use standard algorithms. Hashing algorithms are used in system software (software), blockchain technology and cryptocurrencies, archiving, applications, and database management systems (DBMS). The hash function must be fast to calculate, and its values must be evenly distributed over the domain of definition. When implemented programmatically, hash tables are used – essential data structures to store keywords and identifiers and compare strings, numeric sequences, etc. With a hash table, you can effectively implement an associative array. Hash tables must comply with the following properties: An operation in a hash table begins by computing a hash function given a key. The resulting hash value is an index into the original array. The number of array elements stored divided by the number of possible hash values is called the hash table fill factor and is an important parameter affecting the operation's average execution time. On average, search, insertion, and deletion operations should be completed in O(1) time. However, this estimate does not consider the possible hardware costs of rebuilding the hash table index associated with increasing the array size value and adding a new pair to the hash table. Hashing is proper when a wide range of possible values must be stored in a small amount of memory, and a fast, almost random-access method is needed. Hash tables are often used in databases, especially in language processors such as compilers and assemblers, where

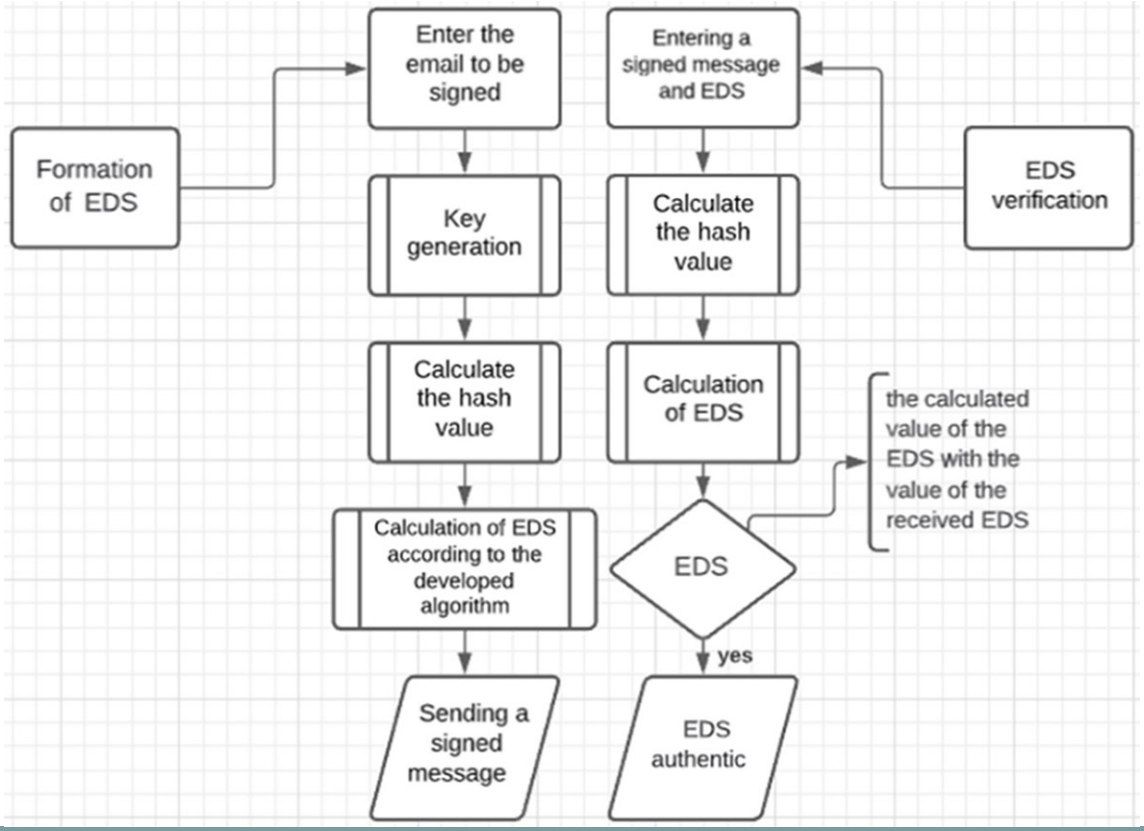

Figure 2 – Diagram of the electronic digital signature algorithm

they speed up the processing of an identifier table. The main problem in implementing a hash table is handling collisions. In case of collisions, finding a new place to store the keys that claim the same cell in the hash table is necessary. If a collision is detected, it is required to minimize their number and find solutions to eliminate them. If all the keys of the elements are known in advance, then it is possible to find some injective hash function that will distribute them among the cells of the hash table without collisions. Hash tables that use hash functions that do not need a conflict resolution mechanism are called direct address hash tables.

There are the following ways of resolving collisions [10]: the method of chains (external or open hashing) and the method of open addressing (closed hashing). The chaining method is a collision resolution technology consisting of elements of a set with equal hash values linked in a chain list. Unlike chain hashing, open addressing does not use lists, all entries are stored in the hash table itself. Using this method, each table cell contains a dynamic set element or NULL. Due to the computational complexity of the tasks on which the electronic digital signature is built, obtaining the private key of the algorithm is almost impossible, it is more likely to search for collisions of the first and second kinds, which is equivalent to existential and selective forgery, respectively. Since hash functions are used in EDS algorithms, finding collisions for the hash functions themselves is expedient [11]. Receiving two documents with the same signature or a collision of the second kind are considered. With a strong hash function, such an attack must be computationally complex. But these threats can be realized due to the weaknesses of specific hashing algorithms, signatures, or errors in their implementations [10].

To check the cryptographic strength of hash functions when searching for collisions, a method based on the «Birthdays» attack is considered. Calculation of the value of the time spent on the test in the event of random collisions with a probability of 0.001 of the considered hash function algorithms, indicating the search time in seconds. Check the translation here and paste it. Using this attack, finding a collision for an n-bit hash function will require an average of about $2^{\frac{n}{2}}$ operations. Therefore, an n-bit hash function is considered secure if the computational complexity of finding collisions for it is close to $2^{\frac{n}{2}}$. For a given hash function h, the goal of the attack is to find a collision of the second kind. To do this, h values are calculated for randomly selected blocks of input data until two blocks are found that have the same hash. The birthday attack succeeds if there is a pair $x_1 \neq x_2$ such that $x_1 \neq x_2$ but $h(x_1) = h(x_2)$. Thus, if the function $h(x)$ gives any of N different outputs with equal probability and N is large enough, then we expect to get a pair of different arguments $x_1 \neq x_2$ with $h(x_1) = h(x_2)$, after evaluating the function around $1.25 \cdot \sqrt{N}$ different arguments. An estimate of the number of hash operations for finding a collision of an ideal cryptographic hash function with an

output size of n bits is often written as $2^{\frac{n}{2}}$ rather than 2^n. A comparison of the time for calculating collisions of hash algorithms (SHA 256, SHA 512, SHA3 256, SHA3 512, STREEBOG 256, STREEBOG 512) with an indication of the collision search time per second are shown in figure 3.

Summary

The EDS continually gains importance and has become a prominent tool in recent years, especially in the corporate world. Thus, the use of a digital signature preserves a high degree of information security and identity. A reliable unit of authenticity, confidentiality, integrity, and access control is provided using encryption methods and hash functions. Secure cryptographic hash functions have basic requirements, such as the inability to generate a message corresponding to a specific hash value and the inability to create two messages that produce the same hash value. Today, the use of hash functions in the EDS is relevant, and EDS is used to confirm data authenticity. In this article, hash functions used in digital signature algorithms were considered, and modern hashing methods and their areas of application were described. A comparative analysis of cryptographic hashing algorithms with the indication of parameters, structures, methods, and their scope are shown. The paper presents the results of calculating collisions of hashing algorithms: SHA 256, SHA 512, SHA3 256, SHA3 512, STREEBOG 256, STREEBOG 512. Based on the results, the most optimal variant for its application in the algorithm under consideration was chosen. In the research, the SHA 512 algorithm was selected and applied since it showed the longest collision search time. Finally, the practical application of the hashing algorithm for the digital signature system, which can be used in systems and networks to transmit and store information, is considered.

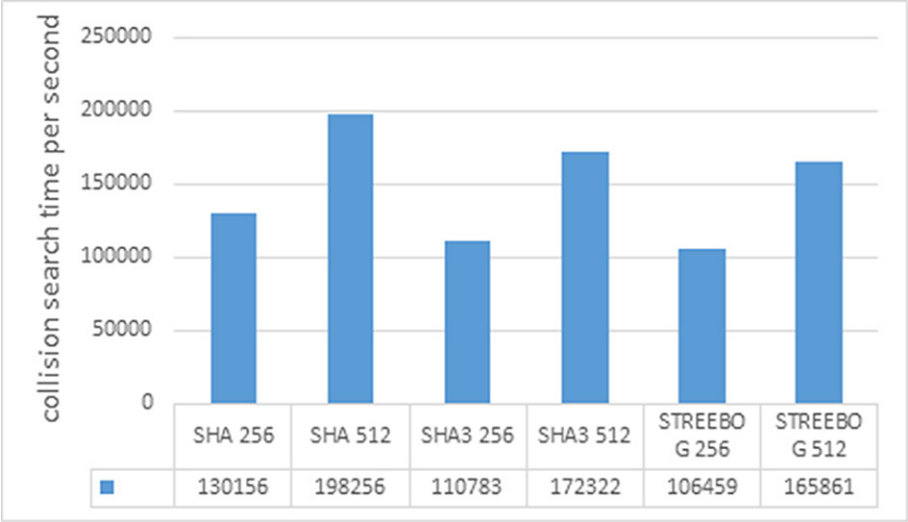

Figure 3 – Collision calculation of hash function algorithms

Acknowledgements

Research work was carried out within the framework of the project №OR11465439 «Development and research of hashing algorithms of arbitrary length for digital signatures and assessment of their strength», which is being implemented at the Institute of Information and Computational Technologies CS MES RK.

REFERENCES

1. International standard ISO / IEC 14888-1-2008 «Information technology. Protection methods. Digital signatures with an application» [Electronic resource]. URL https://www.iso.org/obp/ui/#iso:std:iso-iec:14888:-1:ed-2:v1:en: 06.07.2021.
2. EDS protocol [Electronic resource]. http://ezp20.ru/protokol-ecp: 06.07.2021.
3. Electronic digital signature [Electronic resource]. http://ezp20.ru/elektronnaya-cifrovaya-podpis-vikipediya: 06.07.2021.
4. Kalimoldayev, M.N., Biyashev, R.G., Nyssanbayeva, S.E., & Begimbayeva, Y.Y. (2016). Modification of the digital signature, developed on the nonpositional polynomial notations. Eurasian Journal of Mathematical and Computer Applications, 4 (2), 33-38.
5. A_Comprehensive_Study_on_Digital-Signatures_with_Hash-Functions [Electronic resource]. https://www.researchgate.net/publication/332752862: 16.10.2021.
6. Share of companies using digital signature in Spain from 2009 to 2016 [Electronic resource]. https://www.statista.com/statistics/463046/spain-share-of-companies-using-digital-signature/: 27.05.2021.

7. Arvind K. Sharma, Sudesh K. Mittal Cryptographic Keyed Hash Function: PARAS´U-256 // Journal of Computational and Theoretical Nanoscience Vol. 17, 5072-5084, 2020.
8. Begimbayeva Yenlik, Ussatova Olga, Biyashev Rustem, Nyssanbayeva Saule «Development of an automated system model of information protection in the cross-border exchange» // Cogent Engineering Journal, Birmingham, UK, no. 7, 2020. ISSN: 2331-1916, pp. 1-13. https://doi.org/10.1080/23311916.2020.1724597.
9. FIPS180-3, Secure Hash Standard (SHS), National Institute of Standards and Technology, US Department of Commerce, Washington D.C., 2008.
10. There are the following ways to resolve a collision // http://aliev.me/runestone/SortSearch/Hashing.html: 16.04.2022.
11. S.V. Zapechnikov Cryptographic protocols and their application in financial and commercial activities. – Moscow: Hotline-Telecom, 2007. – 320 p.

Хэш-функцияларды талдау және оларды электрондық цифрлық қолтаңбада қолдану

[1,2]УСАТОВА Ольга Александровна, *PhD, аға ғылыми қызметкер, uoa_olga@mail.ru,*
[2,3]*БЕГИМБАЕВА Енлик Ериковна, *PhD, аға ғылыми қызметкер, enlik_89@mail.ru,*
[4]УСАТОВ Никита Сергеевич, *студент, usatov.nikita2242@gmail.com,*
[1]«Ғұмарбек Дәукеев атындағы Алматы энергетика және байланыс университеті» КеАҚ, Қазақстан, Алматы, А. Байтұрсынұлы көшесі, 126/1,
[2]Ақпараттық және есептеуіш технологиялар институты, Қазақстан, Алматы, Шевченко көшесі, 28,
[3]«Қ.И. Сәтбаев атындағы Қазақ ұлттық техникалық зерттеу университеті» КеАҚ, Қазақстан, Алматы, Сәтбаев көшесі, 22а,
[4]«Тұран» университеті, Қазақстан, Алматы, Сәтбаев көшесі, 16а,
**автор-корреспондент.*

Аңдатпа. *Хеш функция бағдарламалауда алгоритмдік шешімдерді жүзеге асыру үшін кеңінен қолданылады. Хеш функциялары өңделген және сақталатын деректермен жұмысты шифрлау және оңтайландыру үшін, сондай-ақ әртүрлі операциялық жүйелерде, олардың тұтастығын қамтамасыз ету үшін деректерді персонализациялау үшін қолданылады. Хеш-функцияның қолданылуы айтарлықтай кең. Хеш-функцияны пайдалану субъектілер мен ақпараттық өзара әрекеттесу объектілерінің түпнұсқалығын қамтамасыз етуден қорғау құралдары мен объектілерінің жұмысына белгісіздік енгізуге дейін, электрондық ақпаратты қорғаудың барлық дерлік мәселелерін шешуге мүмкіндік береді. Мақалада электрондық цифрлық қолтаңба алгоритмдерінде қолданылатын хеш функциялары қарастырылады. Қазіргі заманғы хештеудің әдістері және олар қолданылатын салалар сипатталған. Хеш функциясы деректерді қорғау үшін пайдаланылады. Бұл функциялар сөз ұзындығы, күрделілігі және криптографиялық күші бойынша әртүрлі болуы мүмкін. Криптографиялық хештеу алгоритмдері параметрлерімен, құрылымдарымен және әдістерімен, сондай-ақ олардың қолданылу аясымен берілген. Жұмыс хеш-функцияларды талдауға және олардың электрондық цифрлық қолтаңбада қолданылуына арналған. Мақалада хештеу алгоритмдерінің соқтығысуын есептеу нәтижелері берілген, ол қарастырылып отырған алгоритмде оны қолданудың ең оңтайлы нұсқасын таңдауға мүмкіндік береді. Ақпаратты тарату және сақтау үшін жүйелер мен желілерде қолданылуы мүмкін электрондық цифрлық қолтаңба жүйесіне хештеу алгоритмінің практикалық қолданылуы қарастырылады.*

Кілт сөздер: *ақпараттық қауіпсіздік, хеш функция, коллизия, криптографиялық хеш функция, электрондық цифрлық қолтаңба, асимметриялық криптожүйе, деректерді қорғау, кілттерді генерациялау, шабуыл, хеш алгоритмдер.*

Анализ хеш-функций и применение их в электронной цифровой подписи

[1,2]УСАТОВА Ольга Александровна, *PhD, старший научный сотрудник, uoa_olga@mail.ru,*
[2,3]*БЕГИМБАЕВА Енлик Ериковна, *PhD, старший научный сотрудник, enlik_89@mail.ru,*
[4]УСАТОВ Никита Сергеевич, *студент, usatov.nikita2242@gmail.com,*
[1]НАО «Алматинский университет энергетики и связи имени Гумарбека Даукеева», Казахстан, Алматы, ул. А. Байтурсынова, 126/1,
[2]Институт информационных и вычислительных технологий, Казахстан, Алматы, ул. Шевченко, 28,
[3]НАО «Казахский национальный исследовательский технический университет имени К.И. Сатпаева», Казахстан, Алматы, ул. Сатпаева, 22а,
[4]Университет «Туран», Казахстан, Алматы, ул. Сатпаева, 16а,
**автор-корреспондент.*

Аннотация. *Хеш-функция широко применяется для реализации алгоритмических решений в программировании. Хеш-функции используются для шифрования и оптимизации работы с обрабатываемыми и хранимыми данными, а также в различных операционных системах, персонализации данных для обеспечения их целостности. Применение хеш-функции весьма обширно. Использование хеш-функции позволяет решать практи-*

чески все задачи защиты электронной информации: от обеспечения аутентичности субъектов и объектов информационного взаимодействия до внесения неопределенности в работу средств и объектов защиты. В статье рассматриваются хеш-функции, применяемые в алгоритмах электронной цифровой подписи. Описаны современные методы хеширования и области, в которых они применимы. Хеш-функция используется для защиты данных. Данные функций могут различаться по разрядности, сложности и криптографической стойкости. Приведены криптографические алгоритмы хеширования с указанием параметров, структур и методов, а также область их применения. Работа посвящена анализу хеш-функций и применению их в электронной цифровой подписи. В статье приведены результаты расчета коллизий алгоритмов хеширования, которые позволят выбрать наиболее оптимальный вариант для ее применения в рассматриваемом алгоритме. Рассмотрено практическое применение алгоритма хеширования для системы цифровой подписи, который может быть использован в системах и сетях для передачи и хранения информации.

Ключевые слова: *информационная безопасность, хеш-функция, коллизия, криптографическая хеш-функция, электронная цифровая подпись, асимметричные криптосистемы, защита информации, генерация ключей, атака, хеш-алгоритмы.*

REFERENCES

1. International standard ISO / IEC 14888-1-2008 «Information technology. Protection methods. Digital signatures with an application» [Electronic resource]. URL https://www.iso.org/obp/ui/#iso:std:iso-iec:14888:-1:ed-2:v1:en: 06.07.2021.
2. EDS protocol [Electronic resource]. http://ezp20.ru/protokol-ecp: 06.07.2021.
3. Electronic digital signature [Electronic resource]. http://ezp20.ru/elektronnaya-cifrovaya-podpis-vikipediya: 06.07.2021.
4. Kalimoldayev, M.N., Biyashev, R.G., Nyssanbayeva, S.E., & Begimbayeva, Y.Y. (2016). Modification of the digital signature, developed on the nonpositional polynomial notations. Eurasian Journal of Mathematical and Computer Applications, 4 (2), 33-38.
5. A_Comprehensive_Study_on_Digital-Signatures_with_Hash-Functions [Electronic resource]. https://www.researchgate.net/publication/332752862: 16.10.2021.
6. Share of companies using digital signature in Spain from 2009 to 2016 [Electronic resource]. https://www.statista.com/statistics/463046/spain-share-of-companies-using-digital-signature/: 27.05.2021.
7. Arvind K. Sharma, Sudesh K. Mittal Cryptographic Keyed Hash Function: PARAS´U-256 // Journal of Computational and Theoretical Nanoscience Vol. 17, 5072-5084, 2020.
8. Begimbayeva Yenlik, Ussatova Olga, Biyashev Rustem, Nyssanbayeva Saule «Development of an automated system model of information protection in the cross-border exchange» // Cogent Engineering Journal, Birmingham, UK, no. 7, 2020. ISSN: 2331-1916, pp. 1-13. https://doi.org/10.1080/23311916.2020.1724597.
9. FIPS180-3, Secure Hash Standard (SHS), National Institute of Standards and Technology, US Department of Commerce, Washington D.C., 2008.
10. There are the following ways to resolve a collision // http://aliev.me/runestone/SortSearch/Hashing.html: 16.04.2022.
11. S.V. Zapechnikov Cryptographic protocols and their application in financial and commercial activities. – Moscow: Hotline-Telecom, 2007. – 320 p.