ISSN 2518-1467 (Online),
ISSN 1991-3494 (Print)

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ
ҰЛТТЫҚ ҒЫЛЫМ АКАДЕМИЯСЫНЫҢ
Абай атындағы Қазақ ұлттық педагогикалық университетінің

Х А Б А Р Ш Ы С Ы

ВЕСТНИК

НАЦИОНАЛЬНОЙ АКАДЕМИИ НАУК РЕСПУБЛИКИ КАЗАХСТАН
Казахский национальный педагогический университет имени Абая

THE BULLETIN

THE NATIONAL ACADEMY OF SCIENCES OF THE REPUBLIC OF KAZAKHSTAN
Abai Kazakh National Pedagogical University

PUBLISHED SINCE 1944

3 (403)

MAY-JUNE 2023

ALMATY, NAS RK

«Қазақстан Республикасы Ұлттық ғылым академиясының Хабаршысы».
ISSN 2518-1467 (Online),
ISSN 1991-3494 (Print).
Меншіктенуші: «Қазақстан Республикасының Ұлттық ғылым академиясы» РҚБ (Алматы қ.).
Қазақстан Республикасының Ақпарат және коммуникациялар министрлігінің Ақпарат комитетінде 12.02.2018 ж. берілген
№ 16895-Ж мерзімдік басылым тіркеуіне қойылу туралы куәлік.
Тақырыптық бағыты: *әлеуметтік ғылымдар саласындағы зерттеулерге арналған.*
Мерзімділігі: жылына 6 рет.
Тиражы: 300 дана.
Редакцияның мекен-жайы: 050010, Алматы қ., Шевченко көш., 28, 219 бөл., тел.: 272-13-19
http://www.bulletin-science.kz/index.php/en/

Типографияның мекен-жайы: «Аруна» ЖК, Алматы қ., Мұратбаев көш., 75.

«Вестник Национальной академии наук Республики Казахстан».
ISSN 2518-1467 (Online),
ISSN 1991-3494 (Print).
Собственник: РОО «Национальная академия наук Республики Казахстан» (г. Алматы). Свидетельство о постановке на учет периодического печатного издания в Комитете информации Министерства информации и коммуникаций и Республики Казахстан **№ 16895-Ж,** выданное 12.02.2018 г.
Тематическая направленность: *посвящен исследованиям в области социальных наук.*
Периодичность: 6 раз в год.
Тираж: 300 экземпляров.
Адрес редакции: 050010, г. Алматы, ул. Шевченко, 28, ком. 219, тел. 272-13-19
http://www.bulletin-science.kz/index.php/en/

Адрес типографии: ИП «Аруна», г. Алматы, ул. Муратбаева, 75.

Bulletin of the National Academy of Sciences of the Republic of Kazakhstan.
ISSN 2518-1467 (Online),
ISSN 1991-3494 (Print).
Owner: RPA «National Academy of Sciences of the Republic of Kazakhstan» (Almaty). The certificate of registration of
a periodical printed publication in the Committee of information of the Ministry of Information and Communications
of the Republic of Kazakhstan **No. 16895-Ж,** issued on 12.02.2018.
Thematic focus: *it is dedicated to research in the field of social sciences.*
Periodicity: 6 times a year.
Circulation: 300 copies.
Editorial address: 28, Shevchenko str., of. 220, Almaty, 050010, tel. 272-13-19
http://www.bulletin-science.kz/index.php/en/

Address of printing house: ST «Aruna», 75, Muratbayev str, Almaty.

BULLETIN OF NATIONAL ACADEMY OF SCIENCES
OF THE REPUBLIC OF KAZAKHSTAN
ISSN 1991-3494
Volume 3. Number 403 (2023), 203-217
https://doi.org/10.32014/2023.2518-1467.503

ӘОЖ 004.021;
ҒТАМР 14.35.07

© M. Serik*, D.Sh. Tleumagambetova, 2023
L.N. Gumilyov Eurasian National University, Kazakhstan, Astana.
E-mail: *danara1310@gmail.com*

METHOD IMPLEMENTATION OF CRYPTOGRAPHIC ALGORITHMS IN PYTHON

Serik Meruert — Doctor of Pedagogical Sciences, Professor of L.N. Gumilyov Eurasian National University, Kazakhstan, Astana
E-mail: *serik_meruerts@mail.ru*, *https://orcid.org/0000-0002-2801-432X;*
Tleumagambetova Danara — PhD Doctoral Student of L.N. Gumilyov Eurasian National University, Kazakhstan, Astana
E-mail: *danara1310@gmail.com*, *https://orcid.org/0009-0007-4221-3900.*

Abstract. These days, the most calculate influencing the political and financial component of national security is the degree of assurance of data and the data environment. Consequently, the issue of ensuring the security and integrity of confidential information when it is transferred from one system to another is always considered as one of the topical issues. Accordingly, it is necessary to consider new approaches and methods of ensuring information security in accordance with the intensive development of digital technologies. The main purpose of this article is to consider cryptographic algorithms (standard cryptographic algorithms and cryptographic hash functions) in the Python programming environment and to determine the importance of modern cryptographic hash functions. Cryptography is one of the most important tools for ensuring information security in the field of information security. This gives us the opportunity to protect confidential information from unauthorized access by encryption. Currently, along with cryptographic algorithms, cryptographic hash functions are effective tools for ensuring data security. Therefore, the article outlines the theoretical foundations, principles and types of hash functions as a basis for the upcoming practical work of students. Cryptography algorithms were taken as the main programming environment, because they are convenient to perform with the built-in cryptography library and pyperclip plugins in the Python programming environment. The article discusses the most commonly used standard reverse encryption algorithms, the Vigener cipher, the Caesar Cipher, multiplicative encryption. The research work was carried out by students of the

speciality "6B01511 - Computer science" of the L. N. Gumilyov Eurasian National University. The practical tasks described in the article algorithm for performing cryptographic hash functions, as well as the basics of automatic text encryption, allow students to master the principles of performing cryptographic algorithms and libraries used in cryptography in the Python programming environment, functions and independently create password hash functions.

Keywords: cryptography, algorithm, encryption, decryption, information security, hashing algorithms, hash functions

© М. Серік*, Д.Ш. Тлеумагамбетова, 2023
Л.Н. Гумилев атындағы Еуразия Ұлттық университеті, Қазақстан, Астана.
E-mail: *danara1310@gmail.com*

PYTHON ПРОГРАММАЛАУ ОРТАСЫНДА КРИПТОГРАФИЯ АЛГОРИТМДЕРДІ ЖҮЗЕГЕ АСЫРУ ӘДІСТЕРІ

Серік Меруерт — Л.Н. Гумилев атындағы Еуразия ұлттық университетінің профессоры, п.ғ.д., Астана, Қазақстан
E-mail: *serik_meruerts@mail.ru*, *https://orcid.org/0000-0002-2801-432X;*
Тлеумагамбетова Данара Шайкуалиевна — Л.Н. Гумилев атындағы Еуразия ұлттық университеті, «8D01511 – Информатика» білім беру бағдарламасы докторанты, Астана қ., Қазақстан
E-mail: *danara1310@gmail.com*, *https://orcid.org/0009-0007-4221-3900.*

Аңнотация. Бүгінгі таңда ұлттық қауіпсіздіктің саяси және экономикалық компоненттеріне әсер ететін негізгі фактор ақпарат пен ақпараттық ортаның қорғалу дәрежесі болып табылады. Сондықтан құпия ақпаратты бір жүйеден екінші жүйеге тасымалдау барысында оның қауіпсіздігін, тұтастығын қамтамасызетумәселесіәрдайымөзектімәселелердіңбіріболыпқарастырылады. Сол себепті цифрлық технологиялардың қарқынды дамуына сай ақпараттық қауіпсіздікті қамтамасыз етудің жаңа тәсілдерін, әдістерін қарастыруымыз қажет. Ғылыми мақаланың негізгі мақсаты- Python программалау ортасында криптографиялық алгоритмдерді(стандартты криптографиялық алгоритмдер мен криптографиялық хэш функцияларды) қарастыру және заманауи криптографиялық хэш функциялардың маңыздылығын анықтау. Криптография ақпараттық қауіпсіздік саласындағы ақпараттың қауіпсіздігін қамтамасыз ететін ең маңызды құралдардың бірі. Ол бізге құпия ақпаратты шифрлау арқылы рұқсатсыз қолжеткізуден қорғау мүмкіндігін береді. Қазіргі таңда криптографиялық алгоритмдермен қатар, криптографиялық хэш функциялары деректердің қауіпсіздігін сақтаудың тиімді құралдары болып табылады. Сондықтан мақалада білім алушыларға алдағы практикалық жұмыстарға негіз ретінде хэш функцияларының теориялық негіздері, орындалу принциптері мен түрлері баяндалған. Криптография алгоритмдерін Python программалау ортасында кіріктірілген cryptography кітапханасы мен pyperclip плагиндерімен орындау ыңғайлы және кез-келген қолданушыға қолжетімді болғандықтан

негізгі программалау ортасы ретінде алынды. Мақалада жиі қолданылатын қарапайым кері шифрлеу, Виженер шифрі, Цезар шифрі, мультипликативті шифрлеу алгоритмдері қарастырылады. Зерттеу жұмысы Л.Н.Гумилев атындағы Еуразия Ұлттық университетінің «6B01511- Информатика» білім алушыларымен орындалды. Мақалада көрсетілген практикалық тапсырмалар криптографиялық хэш функциялардың орындалу алгоритмін, сонымен қатар автоматты түрде мәтінді шифрлеу негізі білім алушыларға криптографиялық алгоритмдердің орындалу принциптері мен Python программалау ортасында криптографияда қолданылатын кітапханаларды, функцияларды жетік меңгеруге және құпия сөздердің хэш функцияларын өз бетінше құруға мүмкіндік береді.

Түйін сөздер: криптография, алгоритм, шифрлеу, дешифрлеу, ақпараттық қауіпсіздік, хэштеу алгоритмдері, хэш функциялары

© М.Серік*, Д.Ш.Тлеумагамбетова, 2023
Евразийский национальный университет имени Л.Н. Гумилева,
Казахстан, Астана.
E-mail: *danara1310@gmail.com*

МЕТОДЫ РЕАЛИЗАЦИИ КРИПТОГРАФИЧЕСКИХ АЛГОРИТМОВ В СРЕДЕ ПРОГРАММИРОВАНИЯ PYTHON

Серік Меруерт — д.п.н., профессор Евразийского национального университета им. Л.Н.Гумилева, г.Астана, Казахстан
E-mail: *serik_meruerts@mail.ru*, *https://orcid.org/0000-0002-2801-432X;*
Тлеумагамбетова Данара Шайкуалиевна — Евразийский национальный университет им. Л.Н.Гумилева, докторант по специальности «8D01511 – Информатика», г. Астана, Казахстан
E-mail: *danara1310@gmail.com*, *https://orcid.org/0009-0007-4221-3900.*

Аннотация. На сегодняшний день основным фактором, влияющим на политическую и экономическую составляющую национальной безопасности, является степень защищенности информации и информационной среды. Поэтому вопрос обеспечения безопасности, целостности конфиденциальной информации при ее передаче из одной системы в другую всегда рассматривается как один из актуальных вопросов. Соответственно, необходимо рассмотреть новые подходы, методы обеспечения информационной безопасности в соответствии с интенсивным развитием цифровых технологий. Основная цель научной статьи – рассмотреть криптографические алгоритмы (стандартные криптографические алгоритмы и криптографические хеш-функции) в среде программирования Python и определить важность современных криптографических хеш-функций. Криптография является одним из важнейших инструментов обеспечения безопасности информации в области информационной безопасности. Это дает нам возможность защитить конфиденциальную информацию от несанкционированного доступа путем шифрования. В настоящее время, наряду с криптографическими алгоритмами,

криптографические хеш-функции являются эффективными инструментами для обеспечения безопасности данных. Поэтому в статье изложены теоретические основы, принципы и виды выполнения хеш-функций в качестве основы для предстоящей практической работы обучающихся. Алгоритмы криптографии были взяты в качестве основной среды программирования, поскольку их удобно выполнять со встроенной библиотекой cryptography и плагинами pyperclip в среде программирования Python. В статье рассматриваются наиболее часто используемые стандартные алгоритмы обратного шифрования, шифр Виженера, Шифр Цезаря, мультипликативное шифрование. Исследовательская работа проведена обучающимися Евразийского национального университета им. Л.Н. Гумилева «6В01511 – Информатика». Практические задания, изложенные в статье алгоритм выполнения криптографических хеш-функций, а также основы автоматического шифрования текста позволяют обучающимся в совершенстве освоить принципы выполнения криптографических алгоритмов и библиотеки, используемые в криптографии в среде программирования Python, функции и самостоятельно создавать хеш-функции паролей.

Ключевые слова: криптография, алгоритм, шифрование, дешифрование, информационная безопасность, алгоритмы хеширования, хеш-функции

Кіріспе

Қазіргі таңда қоғамның барлық саласына енген ақпарат ағынының күрт өсуі оның қауіпсіздік шараларын жетілдіруді талап етеді. Қолданыстағы қауіпсіздік шаралары бүгінгі дамып келе жатқан салалардың алдыңғы қатарлы талаптарын орындау үшін бұрынғыданда маңызды талаптардың незігінде жұмыс істеуі қажет. Ақпараттық қауіпсіздікке қатысты негізгі мәселелер ресурстарға қауіпсіз және аутентификацияланған қолжетімділік болып табылатын қорғаныс сияқты салаларда қолданылуы қажеттілігі бар. Ақпараттық қауіпсіздік саласындағы осындай маңызды талаптарға жауап беретін бірден-бір бағыт криптография болып табылады. Бүгінгі таңда криптографияның көптеген алгоритмдерінің жүзеге асырылуы әр түрлі программалық құралдардың көмегімен шыңдалып, автоматтандырылу жүйесіне көшуде. Сондықтан олардың жүзеге асырылу принциптері мен қолданылу ерекшеліктерін түсіну мақсатында практикада көп қолданылатын криптографиялық алгоритмдерді қарастыру қажеттілігі туындады. Демек, зерттеуіміздің негізгі мақсаты криптография саласында қолданыстағы алгоритмдерді(стандартты алгоритмдер мен криптографиялық хэш-функцияларды) Python программалау ортасында жүзеге асыру мүмкіндіктерін қарастыру.

Криптография — бұл кодталған хабарламалар арқылы екі пайдаланушы арасындағы байланыс өнері. Криптография ғылымы бір тараптан екінші тарапқа берілетін құпия хабарламалардың қауіпсіздігін қамтамасыз етудің негізінде пайда болды. Криптография ақпараттық қауіпсіздікте танылған құпиялылық пен құпияны енгізу үшін хабарламаны жасыру өнері мен ғылымы ретінде анықталады (Баричев, 2011).

Криптографияны қолданудың алғашқы белгілі дәлелі б.з.д. 1900 ж.

Египеттегі дворян Хнумхотеп II-нің негізгі камералық қабірінде табылған. Өркениеттің дамуымен қатар криптография, шифрлеу саласы біртіндеп дамып, біздің заманымызда ақпаратты қорғаудың ең маңызды тәсіліне айналды (Sidhpurwala, 2023).

Криптография саласын Отандық және алыс-жақын шет ел зерттеушілері өздерінің еңбектерінде қарастырған. Атап айтқанда, Б.А. Фороузан (Криптография және желілердің қауіпсіздігі) (Фороузан, 2010), Л.Е. Бахаров (Ақпараттық қауіпсіздік және ақпаратты қорғау) (Бахаров, 2019), Н.Ж. Дүйсенов(Компьютерлік жүйелердегі ақпаратты қорғау) (Дүйсенов, 2019), В.Г. Грибунин (Криптография және цифрлық жүйелердің қауіпсіздігі) (Грибунин, 2011), С.Н. Борисова (Ақпаратты қорғаудың криптографиялық әдістері) (Борисова, 2018), Д. Михайлов (MS Excel ортасында криптография және криптоталдау) (Михайлов, 2022).

L.Ning-Bo «Overview on multi-key fully homomorphic encryption» мақаласында гомоморфты криптография жүйесінің типтік құрылысы, бұлттық ортада оның қолданылуы, қолдану барысында туындайтын мәселелері мен болашақ даму тенденциясын қарастырған (Ning-Bo, 2020). Ал, B.Onur, Gr. Matthew «Machine Learning Based Malware Detection on Encrypted Traffic: A Comprehensive Performance Study» мақаласында зиянды бағдарламалар ағынының деректер жиынында шифрланған зиянды бағдарламаларды анықтау үшін кеңінен қолданылатын машиналық оқыту және терең оқыту алгоритмдерінің жиынтығы бойынша кешенді зерттеу жұмысын жүргізген (Onur, 2020). Wegman M.N. «New hash functions and their use in authentication and set equality» мақаласында белгілі бір қасиеттері бар хэш функциялардың жаңа класстары мен қолданбаларын сипаттаған (Wegman, 2003). Manankova O. «Cryptanalysis the SHA-256 Hash Function using Rainbow Tables» атты зерттеу жұмысында SHA-256 хэш функциясына шабуылды ұйымдастыру үшін парольді және кемпірқосақ кестелерін жасау процесін баяндаған (Manankova, 2022). Бұл ғылыми зерттеу жұмыстарының барлығы криптографиялық алгоритмдердің қоғм өміріндегі маңыздылығын, алгоритмдерінің орындалу қағидаларын, жалпы теориялық негіздерін қалыптастыруға негіз болды.

Материалдар мен негізгі әдістер

Зерттеу жұмысымыздың мақсатын жүйелі түрде жүзеге асыру мақсатында теориялық зерттеу әдісі қолданылады. Ақпараттық жүйелердің криптография саласының негіздері және оның қолдану принциптері бізге бұрыннан таныс, дегенмен жаңа технологиялардың дамуына байланысты олар күннен-күнге жетілдіріліп келеді. Сол сияқты біз зерттеу жұмысымызда криптографияда қолданылатын алгоритмдерді заманауи құралдардың көмегімен білім алушыларға жетік түрде ұсынуды мақсат еттік. Ол үшін ең алдымен криптография саласы бойынша әдебиеттер, материалдар қарастырылды және нәтижесі бойынша заманауи криптографиялық алгоритмдерге талдау жасалды. «Ақпараттық қауіпсіздік» пәнінің мазмұнын қарастыру барысында стандартты шифрлеу алгоритмдерінің дәстүрлі жолмен талдау негіздері қарастырылатыны

анықталды. Сондықтан Л.Н. Гумилев атындағы Еуразия ұлттық университетінің «6В011100 – Информатика», «6В01511- Информатика мұғалімдерін даярлау» білім беру бағдарламасындағы «Ақпараттық қауіпсіздік» пәнінің мазмұнын заман талабына сай жетілдіруіміз қажеттілігі туындады. Ол үшін, криптография алгоритмдерін қарастыру үшін Python программалау ортасын негізге алдық. Программалау ортасында криптография алгоритмдерін жүзеге асыру ең алдымен криптография алгоритмдерін орындауды автоматтандырады және де білім алушыларға криптография бойынша қажетті кітапханаларды, функцияларды, командаларды үйренуге мүмкіндік береді.

Нәтижелері

Криптографияда қарапайым мәтін(алғашқы мәтін), шифрланған мәтін, дешифрленген мәтін терминдерінің жиі қолданылатынын білесіздер (Mao, 2005):

Қарапайым мәтіндік хабарлама - бұл оқылатын және барлық пайдаланушыларға түсінікті мәтін. Қарапайым мәтін – бұл криптографиядан өтетін хабарлама;

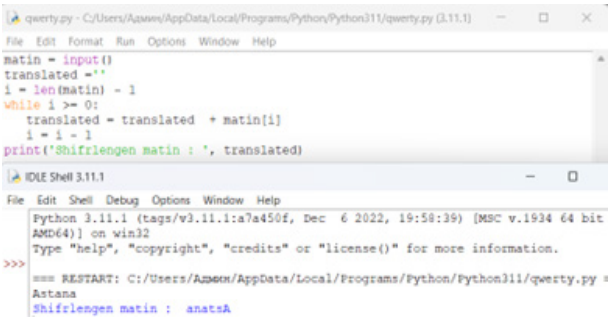
Шифрлық мәтін - кәдімгі мәтінге криптографияны қолданғаннан кейін алынған хабарлама;

Қарапайым мәтінді шифрланған мәтінге түрлендіру процесі шифрлау деп аталады. Оны кодтау деп те атайды;

Шифрланған мәтінді кәдімгі мәтінге түрлендіру процесі шифрды ашу деп аталады. Оны декодтау деп те атайды.

Криптографиялық алгоритмдердің көптеген түрлері бар. Атап айтсақ, Цезарь, Виженер, кері шифрлеу, мультипликативті шифрлеу, ашық кілтті, жабық кілтті және т.б. Ең көп қолданылатын криптографиялық алгоритмдердің орындалу принципін Python программалау ортасында жүзеге асырайық.

Кері шифр. Кері шифрлеу алгоритмі- мәтінді түрлендіру үшін мәтін жолын кері айналдыру үлгісін пайдаланады. Кері шифр сенімді шифр болып табылмайды, себебі хакер бастапқы хабарламаны алу үшін шифр мәтінін оңай бұза алады. Демек, кері шифр қауіпсіз байланыс арнасын қолдаудың жақсы нұсқасы ретінде қарастырылмайды. 2-суреттен оны программалау ортасында жүзеге асырылуы мен нәтижесін көре аламыз (Schneier, 2005).



Сур.2- Кері шифрлеу мысалы
(*Fig.2*- An example of reverse encryption)

Цезарь шифрі. Цезарь шифрінің алгоритмі ағымдық мәтіннің әрбір әрпі әліпбимен төмен орналасқан белгілі бір тұрақты саны бар әріппен ауыстырылу арқылы орындалады. Оны программалау тілінде жүзеге асыру тәсілі:

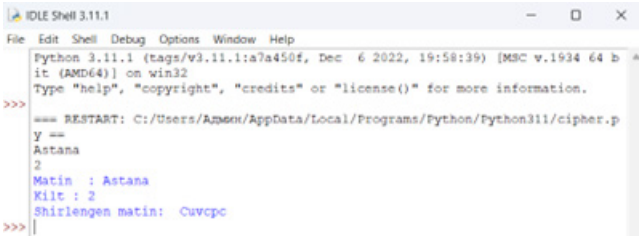
for i in range(len(text)): char = text[i]

if (char.isupper()): result += chr((ord(char) + s-65) % 26 + 65)

else: result += chr((ord(char) + s - 97) % 26 + 97)

return result

3-ші суреттен нәтижесін көруге болады.



Сур.3- Цезарь шифрін программалау ортасында жүзеге асыру нәтижесі
(*Fig.3*- The result of implementing the Caesar cipher in the programming environment)

Мультикативті шифр. Бұл жағдайда әріптерді шифрлеу үшін сандар қолданылады және 7 санына қалдықпен бөлу арқылы басқа әріпке шифрлеу әдісі жүзеге асырылады:

(алфавит номері * кілт) mod (алфавит саны)

1-ші кестедегідей Astana сөзін шифрлейік.

Кесте1-Шифрлеу процесі
Table 1- Encryption process

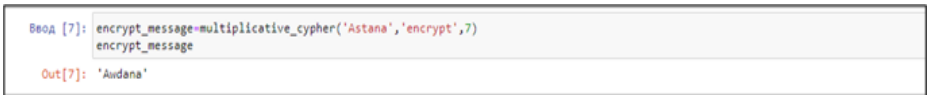
Символ	Номері	Кілт арқылы шифрлеу	Шифрленген символ
A	0	(0*7)%26=0	A
S	18	(18*7)%26=22	W
T	19	(19*7)%26=3	D
A	0	(0*7)%26=0	A
N	13	(13*7)%26=13	N
A	0	(0*7)%26=0	A

Мультикативті шифр алгоритмін программалық жүзеге асырайық (Reddy, 2022):

for char in text: if char in key_list:new_index=(char_dict[char]*key) % 26

cipher_message=cipher_message+inv_char_dict[new_index]

Нәтижесін 4-шы суреттен көруге болады.



Сур.4- Мультипликативті шифрлеу мысалы
(*Fig.4*- An example of multiplicative encryption)

Сурет пен кестедегі мәліметтерді салыстыратын болсақ, бірдей нәтиже көрсеткенін көре аламыз. Яғни, шифрлеу әдісінің дұрыс жүзеге асырылды деп толықтай айта аламыз. Мультипликативті шифрлеу ең негізгі артықшылығы ол 8 953 851 мәніне тең үлкен кілттермен жұмыс істеу мүмкіндігіне ие.

Виженер шифрі. Виженер шифрі алфавиттік мазмұнды шифрлауға арналған әдіс. Ол полиалфавитті ауыстырудың негізгі түрін пайдаланады. Полиалфавиттік шифр – ауыстыруға тәуелді, көптеген алмастыратын алфавиттерді пайдаланатын кез- келген шифр. Шифрлау циклiндегі әртүрлі фокустарда шифр жолдардың бірінен ретімен балама әріптерді пайдаланады. Әр нүктеде қолданылатын әріптер қайталанатын фразаға сүйенеді:

шифрлеу– (хабарлама+кілт) % 26

дешифрлеу – (шифрленген хабарлама-кілт+26)%26

```
def encryption(string, key): encrypt_text = []
for i in range(len(string)): x = (ord(string[i]) +ord(key[i])) % 26
x += ord('A')
encrypt_text.append(chr(x))
return("" . join(encrypt_text))
```

Қарастырылған мысалдардағы криптографиялық процесі екі жақты болып табылады, яғни белгілі бір кілттің көмегімен мәтінді шифрлеу және сол дешифрленген мәтінді қайта шифрлеу. Дегенмен, мұндай алгоритмдер қазіргі таңда ақпараттық технологиялардың дамуының нәтижесінде тиімсіз, қауіпсіз болып табылады. Қазіргі криптографияның негізгі сипаттамалары ол биттік тізбектерде жұмыс істейді және ол ақпаратты қорғау үшін математикалық алгоритмдерді пайдаланады, сонымен қатар құпиялылыққа қол жеткізу үшін қауіпсіз байланыс арнасына мүдделі тараптарды талап етеді. Сондықтан осындай мәселелерді шешу мақсатында криптографиялық хэш функциялар қолданыла бастады. Хэш функциялары кіріс ретінде ерікті жолдарды қабылдайды және кіріс деректеріне тәуелді тұрақты өлшемді шығыс деректерін шығарады. Хэш функциясының шығысын ескере отырып, кіріс деректерін алу ешқашан мүмкін болмауы керек. Қарапайым хэш функциясы- барлық кіріс байттарын қосу және 256 модулі бойынша нәтиже алу болып табылады (Рассел, 2012). Хэш функциясы криптографиялық тұрғыдан қауіпсіз болуы үшін бірдей хэш мәні бар екі хабарламаны табу немесе берілген хэш мәні бар хабарламаны табу өте қиын болуы керек.

Жалпы хэш функция-бұл математикалық функция H: D → R,

Мұндағы, ерікті ұзындықтағы m сандық кіріс мәнін n >= 1 үшін d = {0,1}^* және R = {0,1}^N аймағында h сығылған шығыс сандық мәнді бейнелеу, яғни: h = H (m) (Rivest, 1995).

Криптографиялық хэш функциялары жалпы екі классқа бөлінеді: кілтсіз хэш функциялары, яғни манипуляцияны анықтау коды(MDC) немесе хабарламаның аутентификация коды (MAC) .

Барлық хэштеу алгоритмдері «енгізу» деп аталатын ақпарат бөлігін және хэштің күрделілігін, есептеу талаптарын және қосымша қауіпсіздік шараларын

анықтайтын параметрлерден тұрады. Хэштеу алгоритмінің орындалуы арнайы бір принципке негізделмейді және төмендегі қарастырылған паратмерлердің барлығын қамтымауы мүмкін. Деректерді криптографиялық хэштеу үшін мынадай жалпы мынадай параметрлер қолданылады:

Енгізу. Әрбір хэштеу алгоритмі хэшке енгізуді қажет етеді. Енгізу әдетте әртүрлі ұзындықтағы қарапайым мәтіндік жол болып табылады.

Тұз. «Тұз» хэштелгенге дейін кіріске қосылған немесе алдына қойылған таңбалар жолын білдіреді. Тұз кірістің ұзындығы мен күрделілігін арттыру арқылы хэш мәнін өзгертеді. Бұл сөздік шабуылдарынан немесе кемпірқосақ кестелерінен қорғауға мүмкіндік беретін әдіс.

Ұзындық. Ұзындық әзірленген хэш мәніндегі байттардың немесе HEX таңбаларының санын білдіреді.

Циклдер. Циклдар саны алгоритмнің хэштеу функциясы қанша рет орындалғанын немесе итерация санын сипаттайды. Қаншалықты уикл саны көп болған сайын хэш сенімді түрде жұмыс жасайды. Дегенмен, мұндай типтегі деректерді хэштеу үшін көбірек уақыт қажет болады. Циклдар саны жұмыс коэффициентімен, жұмыс коэффициенті болатын экспоненциалды қатынаспен анықталады.

Жұмыс факторы. Жұмыс коэффициенті хэштеу алгоритмінің орындалу санын білдіреді. Бұл әдетте жұмыс итерациясы сияқты 2-к дәреже ретінде көрсетіледі. Жұмыс коэффициенті неғұрлым жоғары болса, хакерге хэштеу алгоритмін бұзу қиынырақ болады. Бірақ, жұмыс коэффициенті неғұрлым жоғары болса, қолданбаның немесе оның аутентификация провайдерінің есептеу құны соғұрлым жоғары болады.

Ағындар. Ағындар алгоритм хэшті есептеу үшін пайдаланатын қатарлас ағындардың санын немесе параллелизм дәрежесін білдіреді.

Жадының сенімділігі. хэштеу процесінде пайдаланылатын жад көлемі. Алгоритмдер әдетте көптеген факторлармен бағаланады, олардың бірі «жадтың сенімділігі» деп аталады, ол берілген функцияны немесе әрекетті орындау үшін процессор мен жедел жадты пайдаланудың қаншалықты қажет екенін көрсетеді. Хэшинг алгоритмдері жадты тым қиын емес, бұзу оңай болмау арасындағы тепе-теңдікті сақтауы керек.

CPU. Хэшті жасау үшін қажет жад пен процессорды пайдалануды арттыратын шығын немесе жұмыс факторы. Кейбір скрипт енгізулері бұл параметрдің 2 дәрежесі болуын талап етеді.

Хэш функциялардың мынадай криптографиялық алгоритмдері бар:

SHA (Secure hash Algorithm) – ұлттық стандарттар және технологиялар институты (NIST) АҚШ-тың ақпаратты өңдеудің федералды стандарты (FIPS) ретінде жариялады. Мұнда алты түрлі хэш функция түрлері бар: SHA-0, SHA-1, SHA-224, SHA-256, SHA-384 және SHA-512. Алғашқы төртеуі 32 биттік сөздерге бөлінген 512 биттік хабарлама блоктарымен, ал соңғы екеуі 64 биттік сөздерге бөлінген 1024 биттік блоктармен жұмыс істейді. Аталған, криптоалгоритмдердің ішінен SHA-256 қауіпсіз болып саналады.

MD(message Digest)-RSA қауіпсіздігін қамтамасыз ету үшін Рональд Ривест әзірлеген және RFC 1321 интернет стандарты ретінде қабылданған MD2, MD4, MD5 және MD6 түрлерінен тұрады.

Whirlpool-Винсент Римен мен Пауло С. Л. М. Баррето әзірлеген бұл хэш функциясы Advanced Encryption Standard (AES)-тің өзгертілген нұсқасына негізделген.

Blake- Chacha ағындық шифрына негізделген алгоритм.

Хэш-функцияларымен жұмыс істеу барысында оқу үрдісінде пайдалануға болатын қарапайым хэш-функцияларын қарастырайық. Хэш функциясының модульдері мынадай айнымалыларды қолданылады:

digest_size- бүтін мән. хэштеу объектілері жасаған дайджест өлшемі. Сондай-ақ, бұл мәнді үлгі нысанын жасау және оған қайтарылған digest жолының ұзындығын алу арқылы алуға болады.

Объектілерді хэштеу үшін төмендегідей әдістерді қолдануға болады:

copy()- осы хэштеу нысанының жеке көшірмесін қайтарады. Бұл көшірмені жаңарту бастапқы нысанға әсер етпейді.

digest ()- бұл хэштеу объектісінің хэш мәнін 8 биттік деректері бар жол ретінде қайтарады. Бұл функция нысанды ешқандай жолмен өзгертпейді. Бұл функцияны шақырғаннан кейін нысанды жаңартуды жалғастыра аласыз.

hexdigest ()- бұл хэштеу объектісінің хэш мәнін он алтылық сандар түріндегі дайджест деректері бар жол ретінде қайтарады. Алынған жол digest () функциясы қайтарғаннан екі есе ұзын болады. Бұл функция нысанды ешқандай жолмен өзгертпейді. Осы функцияны шақырғаннан кейін нысанды жаңартуды жалғастыра аласыз.

update (arg)- бұл хэштеу нысанын arg жолымен жаңарту. Python 3.x: берілген аргумент байт буфері ретінде түсіндірілетін объект болуы керек.

5-ші суреттегідей қарапайым мәтінді SHA256 хэштеу функциясы көмегімен түрлендірейік.

```
from Crypto.Hash import SHA256
m = SHA256.new()
m.update('abc')
m.digest()
```

Сур.5- Мәтінді хэштеу мысалы
(*Fig.5*- An example of text hashing)

6-суретте көрсетілгендей, нәтижесінде 8 биттік және 16 сандар түрінде дайджест деректері бар жол беріледі.

```
In [ ]: >>> m.digest()
        '\xbax\x16\xbf\x8f\x01\xcf\xeaAA@\xde]\xae"#\xb0\x03a\xa3\x96\x17z\x9c\xb4\x10\xffa\xf2\x00\x15\xad'
        >>> m.hexdigest()
        'ba7816bf8f01cfea414140de5dae2223b00361a396177a9cb410ff61f20015ad'
```

Сур.6 – Нәтижесі
(*Fig.6*- Results)

Кез-келген мәтінді хэштеу алгоритмін қарастырайық. Мұндағы мәтін md5 алгоритмінің көмегімен жүзеге асырылды. Криптографиялық хэштеу функциясының басқа да түрін қолданылу үшін, 7-ші суреттегідей сәйкесінше md5 орнына sha1, sha224, sha128, sha256 жазу арқылы жүзеге асыруға болады.

Сур.7- Мәтінші хэштеу
(*Fig.7*- Text hashing)

Md5 алгоритмі көмегімен алынған мәтінді дешифрлеу барысында бастапқы мәтіннің алынбауын қарастыратын болсақ, нәтижесі бастапқы мәтінмен сәйкес келмейтінін аңғаруға болады. Демек, 8-суретте көрсетілгендей, хэш функциялар криптографиялық шифрлеуге қарағанда қауіпсіз екендігін аңғара аламыз.

Сур.8- Хэш тізбекті керу хэштеу
(*Fig.8*- Reverse hashing)

Сонымен хэштеу функциясын біз көбінесе мәліметтер қорындағы құпия сөздерді хэштеу үшін қолданылатындығы мәлім. Хэштеу процесінде қолданылатын маңызды негізгі ұғымдардың бірі salt(тұз) ұғымы. Криптографияда тұз - құпия сөз сияқты деректерді хэштейтін бір жақты функцияға қосымша кіріс ретінде пайдаланылатын кездейсоқ деректер. Тұздар құпия сөздерді сақтау кезінде оларды қауіпсіз сақтау үшін қолданылады. Тарихи түрде жүйеде тек криптографиялық құпия сөз хэш функциясына қолдау көрсетілді, бірақ уақыт өте келе қайталанатын немесе жалпы құпия сөздерді анықтауды болдырмау үшін қосымша қорғау шаралары әзірленді. Осындай алдын алу шараларының бірі – тұздау(salting).

Құпия сөздерді хэштеу кезінде тұзды пайдаланудың бірнеше себептері бар:

Инсайдерлік шабуылдардан қорғау. Тұздау инсайдерлердің құпия сөздерді ұрлауын қиындатады, өйткені тұз қолданылған соң деректерді(құпия сөздерді) оңай бұза алмайды.

Күш шабуылдарының алдын алу. Күш шабуылдары дұрыс құпия сөз

табылмайынша таңбалардың барлық ықтимал комбинацияларын сынауды қамтиды.

Сөздік шабуылдарының алдын алу. Тұздау арқылы, тіпті бір құпия сөзді бірнеше пайдаланушы пайдаланса да, олар әртүрлі хэштермен аяқталады.

Ережелерге сәйкестік. Көптеген деректерді қорғау ережелері құпия сөздердің қауіпсіз сақталуын талап етеді. Құпия сөздерді тұздау - бұл ережелерге сәйкес орындалғанын тексерудің тиімді жолы.

Python ортасында Bcrypt, Argon2, hashlib сияқты құпия сөздерді хэштеу алгоритмдері қолданылады. Bcrypt алгоритмі көмегімен «Astana» сөзін хэштеу үшін құпия сөзге salt кездейсоқ деректі қолданылып, хэштейміз. 4-ші суреттен «Astana» құпия сөзінің кездейсоқ дерек көмегімен хэштелу мысалы көрсетілген.

Сур.9- Bcrypt алгоритмі көмегімен деректі хэштеу
(*Fig.9*- Data hashing using the Bcrypt algorithm)

Bcrypt алгоритмін жүзеге асыру үшін bcrypt.gensalt(),bcrypt.hashpw() функциялары қолданылды. Мұндағы, bcrypt.gensalt() тұзды өндіру үшін пайдаланылады. Тұз ешқандай аргументтерді қажет етпейді және жалған кездейсоқ жолды қайтарады. Ал, bcrypt.hashpw() – ол дерекқорда сақталған соңғы хэшті әзірлеу үшін пайдаланылады. Тұз бен құпия сөзді байт-код түрінде бере аламыз. Ал, хэштеу нәтижесіндегі мәні хэштеу дұрыс орындалған жағдайда, ол хэш жолын экранға шығарады.

Python ортасында жиі қолданылатын алгоритмдердің бірі hashlib. Ол шифрланған пішімдегі кез келген өңделмеген хабарламаны хэштеуді өңдейтін көптеген әдістерді қамтиды. Бұл модульдің негізгі мақсаты жолдық типтегі деректі үшінші қолдануға қолжетімсіз болатындай хэштеу. hashlib алгоритмі хэш кестесін жасау үшін пайдаланылады. Хэш-кесте — жазбалар жиыны бойынша іздеуге арналған деректер құрылымы, сонымен қатар олардың әрқайсысы бірегей кілтпен анықталады. 6-шы суретте hashlib алгоритмі көмегімен деректі хэштеу мысалын қарастырайық.

```
import hashlib
password = 'Astana'
salt = "Sgz"
dataBase_password = password+salt
hashed = hashlib.md5(dataBase_password.encode())
print(hashed.hexdigest())
```

```
422a6d2c33ebb7df62609bc5895facc9
```

Сур.10- hashlib алгоритмі көмегімен деректі хэштеу
(*Fig.10*- Data hashing using the hashlib algorithm)

Қарастырылған алгоритм көмегімен деректерді хэштеу қағидасы мынадай түрде жүзеге асырылады: құпия сөзді тағайындау, тұзды тағайындау, тұзды құпия сөздің

соңына тіркеу, құпия сөзді хэштеу. Сонымен қатар, құпия сөздерді хэштеуге арналған argon алгоритмі қолданылады. Ол сөздік шабуылдары мен алдын-ала есептеу шабуылдары сияқты шабуылдардан қорғауға арналған. Python тілінде құпия сөзді хэштеу үшін Argon2 пайдалану үшін argon2-cffi кітапханасын пайдалануға болады. 11-ші суретте Argon2 алгоритмі көмегімен деректі хэштеу мысалы келтірілген.

```
import argon2
password = b'Astana'
hashed_password = argon2.hash_password(password)
print(hashed_password)
```

```
b'$argon2i$v=19$m=65536,t=3,p=4$6m9HDgy+UzGnoWw4Mn4w0w$w0IvJKQ6smEhs9yKk05wICeOW8rzB6sS7WPb7HQFhh0'
```

Сур.11- Argon2 алгоритмі көмегімен деректі хэштеу
(*Fig.11*- Data hashing using the Argon2 algorithm)

Кез-келген деректерді хэштеу барысында алынған хэштелген деректер мен бастапқы дерекке сәйкес келмейтін хэш функцияның қосындысы сәйкес болатын болса, онда сайтқа(мәліметтер қорына) оңай қол жеткізуге болады. Мұндай жағдайларды коллизия деп аталады. Демек, коллизия дегеніміз хэш нәтижесі тең болатын x және y кіріс деректері. Сондықтан хэш деректердің сенімділігі үшін кез-келген алгоритм үшін коллизия мәні 0-ге тең болу қажет.

Талқылау

Білім алушыларға деректерді хэштеу үшін қажетті қарастырылған стандартты алгоритмнің біреуін таңдау оның жад сенімділігі, орындалу уақыты, параметр номерін өлшеумен байланысты. Дегенмен, қысқаша түрде талдайтын болсақ, Argon2 – бұл жадты қажет ететін құпия сөзді хэштеу алгоритмі. Бұл оны желіден тыс кілттерді пайдалану үшін тиімді алгоритм болып табылады, дегенмен оның орындалуы үшін көп уақытты қажет етіледі, сондықтан веб-қосымшалар үшін онша қолайлы емес алгоритм болып табылады. bcrypt хэштеу уақытын 1 секундтан аз уақытқа жеткізе алады, бірақ ағындар, процессор немесе жад сенімділігі сияқты параметрлерді қамтымайды. hashlib шабуылдарға төтеп бере алатын сенімді алгоритм, бірақ сияқты есте сақтау күрделі болып табылады.

Практикалық жұмыс нәтижесінде тақырыпты бекіту мақсатында білім алушыларға Bcrypt, Argon2, hashlib алгоритмдерінің көмегімен қарапайым құпия сөзді хэштеу барысындағы оның орындалу алгоритмін анықтау ұсынылды. Құпия сөзді хэштеу функциясын жүзеге асыру үшін әдістемелік нұсқаулық ретінде мынадай программалық алгоритм ұсынылды: сәйкес кітапхананы шақыру, құпия сөзді енгізу, hashed= сәйкесхэш алгоритмі(матін), хэштелген алгоритмді шығару, тестілеу уақытын анықтау.

Қарастырылған зерттеу жұмысын ақпараттық технологиялар саласындағы кез-келген білім беру бағдарламаларындағы «Ақпараттық қауіпсіздік» пәнінде оқу материалы ретінде қолдануға болады.

Қорытынды

Жоғарыда қарастырылған кез-келген криптографиялық шифрлеу алгоритмдерінің артықшылығы мен кемшілігін аңғара аламыз. Мысалы, кері шифрлеу әдісінің қауіпсіз екендігін анықтадық. Сонымен қатар, білім алушыларға криптографиялық шифрлеу алгоритмдерінің орындалу алгоритмдерін жетік түсіну мақсатында криптографияда жиі қолданылатын кері шифрлеу, Цезарь шифрі, Виженер шифрі мен мультипликативті шифрлеу алгоритмдерін

Python программалау ортасында жүзеге асыру қарастырылды. Python тілін криптография үшін пайдалану C немесе C++ сияқты тілдерді пайдаланудан тиімді, себебі программалау тілінде криптография алгоритмдерін автоматты түрде жүзеге асыруға арналған қолжетімді кітапханалар жүйесі бар. Қарастырылған криптографиялық алгоритмдердің қаншалықты қол жетімді және түсіну оңай болғанына қарамастан, оның ақпараттық қауіпсіздікті қамтамасыз ете алмайтынын аңғардық. Осы мәселелерді шешу мақсатында білім алушыларға криптографиялық хэш функцияларды қарастыру ұсынылды, себебі олар біржақты ұйымдастырылады. Яғни, үшінші тарап криптографиялық хэш алгоритмді таба алу жағдайында да ол бастапқы(ағымдық) мәтінге қол жеткізе алмайды. Демек, бір криптографиялық алгоритмнің көмегімен әр түрлі нәтижеге жеткізуге болады. Бұл мәселе ең алдымен ақпараттың қауіпсіздігін қамтамасыз етеді. Әсіресе, деректерді хэштеу функциясы ақпараттық объектілерге аутентификациялау жағдайында қолдану тиімді, себебі мұндағы құпия сөздер хэштелген деректер түрінде сақталады. Ол үшін кез-келген криптографиялық хэш функциясын қолдану барысында құпия сөзге salt(тұз) кездейсоқ деректер жиыны қолдану қажет және мұндағы коллизия мәні 0-ге дейін теңестірілу қажет. Соның нәтижесінде ғана біз кез-келген аутентификация деректерінің ақпараттық қауіпсіздігін қамтамасыз ете аламыз. Қарастырылған зерттеу жұмысы білім алушыларға жетік түрде Python программалау ортасының көмегімен криптографияның стандартты алгоритмдер мен криптографиялық хэш-функцияларды жетік меңгеруге және заман талабына сай практикалық дағдыларды қалыптастыруға мүмкіндік береді.

ӘДЕБИЕТТЕР

Баричев С.Г., 2011 — Основы современной криптографии. - М.: Горячая линия – Телеком. 2011.- 45 с.

Бахаров Л.Е., 2019 — Информационная безопасность и защита информации.- Литрес. 2019.- 54 с.

Борисова С.Н., 2018 — Криптографические методы защиты информации. - Пенза : ПГУ. 2018. - 186 с.

Manankova O., 2022 — Cryptanalysis the SHA-256 Hash Function using Rainbow Tables, Indonesian Journal of Electrical Engineering and Informatics (IJEEI),4. Pp. 930–944.

Ning-Bo L., Tan-Ping Zh., 2020 — Overview on multi-key fully homomorphic encryption, Cryptologic Research,4. Pp.711–713.

Onur B., Matthew Gr., 2020 — Machine Learning Based Malware Detection on Encrypted Traffic: A Comprehensive Performance Study. 7th International Conference on Networking, Systems and Security. P. 47.

Reddy A., 1995 — Cryptography with Python [Электронды ресурс]. URL: *https://www.tutorialspoint.com/cryptography_with_python* (in Eng.)

Rivest R., 1995 — The MD5 Message-Digest Algorithm. - Network Working Group. RFC. 1995.-P. 1321

Schneier B., 2005 — Schneier on Security [Electronic resource]. URL: *http://www.schneier.com/blog/archives/2005/02/cryptanalysis_o.html* (in Eng.)

Sidhpurwala H., 2023 — A Brief History of Cryptography. - USA: Red Hat. 2023.- P.4.

Wegman M.N., 2003 — New hash functions and their use in authentication and set equality, Journal of computer and system sciences, 22: 265–2797

Грибунин В.Г., Мартынов А.П., Николаев Д.Б., Фомченко В.Н., 2011 — Криптография и безопасность цифровых систем. - Саров. 2011.- 411 с.

Дүйсенов Н.Ж., 2019 — Компьютерлік жүйелердегі ақпаратты қорғау.- Шымкент: ОҚМУ. 2019.- 12 б.

Мао В., 2005 — Современная криптография. Теория и практика.- Вильямс. 2005. - 768 с.

Михайлов Д., 2022 — Криптография и критоанализ с MS Excel.-Математика и информатика. 2022.-53–71 с.

Рассел Дж., 2012 — Tiger (хэш-функция). - М.: VSD. 2012. - 833 с.

Фороузан Б.А., 2010 — Криптография и безопасность сетей. - Интернет университет информационных технологий. 2010.- 741с.

REFERENCES

Barichev S.G., 2011 — Fundamentals of modern cryptography. - M.: Hotline – Telecom. 2011.- 45 p.

Bakharov L.E., 2019 — Information security and information protection.- Liters. 2019.- 54 p.

Borisova S.N., 2018 — Cryptographic methods of information protection. - Penza : PSU. 2018. - 186 p.

Duisenov N.ZH., 2019 — Komp'yuterlik zhuielerdegi akparatty korgau.- Shymkent: OKMU. 2019.- 12 p.

Forouzan B.A., 2010 — Cryptography and Network security. - Internet University of Information Technologies. 2010.- 741 p.

Gribunin V.G., Martynov A.P., Nikolaev D.B., Fomchenko V.N., 2011 — Cryptography and security of digital systems. - Sarov. 2011.- 411 p.

Manankova O., 2022 — Cryptanalysis the SHA-256 Hash Function using Rainbow Tables, Indonesian Journal of Electrical Engineering and Informatics (IJEEI),4. Pp. 930–944.

Mao V., 2005 — Modern Cryptography. Theory and practice.- Williams. 2005. - 768 p.

Mihajlov D., 2022 — Kriptografiya i kritoanaliz s MS Excel.-Matematika i informatika. 2022.- Pp. 53–71 c.

Ning-Bo L., Tan-Ping Zh., 2020 — Overview on multi-key fully homomorphic encryption, Cryptologic Research,4. Pp.711–713.

Onur B., Matthew Gr., 2020 — Machine Learning Based Malware Detection on Encrypted Traffic: A Comprehensive Performance Study. 7th International Conference on Networking, Systems and Security. P. 47.

Rassel Dzh., 2012 — Tiger (hesh-funkciya). - M.: VSD. 2012. - 833 p.

Reddy A. Cryptography with Python [Electronic resource]. URL: https://www.tutorialspoint.com/cryptography_with_python (in Eng.)

Rivest R., 1995 — The MD5 Message-Digest Algorithm. - Network Working Group. RFC. 1995.-P. 1321. Group. RFC,1995.-P. 1321

Schneier B. Schneier on Security [Electronic resource]. URL: *http://www.schneier.com/blog/archives/2005/02/cryptanalysis_o.html* (in Eng.)

Sidhpurwala H., 2023 — A Brief History of Cryptography. - USA: Red Hat. 2023.- P.4.

Wegman M.N., 2003 — New hash functions and their use in authentication and set equality, Journal of computer and system sciences, 22. Pp. 265–2797

МАЗМҰНЫ

ПЕДАГОГИКА

ЭКОНОМИКА

СОДЕРЖАНИЕ

ПЕДАГОГИКА

ЭКОНОМИКА

CONTENTS

PEDAGOGY

EKONOMICS

Publication Ethics and Publication Malpractice in the journals of the National Academy of Sciences of the Republic of Kazakhstan

For information on Ethics in publishing and Ethical guidelines for journal publication see http://www.elsevier.com/publishingethics and http://www.elsevier.com/journal-authors/ethics.

Submission of an article to the National Academy of Sciences of the Republic of Kazakhstan implies that the work described has not been published previously (except in the form of an abstract or as part of a published lecture or academic thesis or as an electronic preprint, see http://www.elsevier.com/postingpolicy), that it is not under consideration for publication elsewhere, that its publication is approved by all authors and tacitly or explicitly by the responsible authorities where the work was carried out, and that, if accepted, it will not be published elsewhere in the same form, in English or in any other language, including electronically without the written consent of the copyright-holder. In particular, translations into English of papers already published in another language are not accepted.

No other forms of scientific misconduct are allowed, such as plagiarism, falsification, fraudulent data, incorrect interpretation of other works, incorrect citations, etc. The National Academy of Sciences of the Republic of Kazakhstan follows the Code of Conduct of the Committee on Publication Ethics (COPE), and follows the COPE Flowcharts for Resolving Cases of Suspected Misconduct (http://publicationethics.org/files/u2/New_Code.pdf). To verify originality, your article may be checked by the originality detection service Cross Check http://www.elsevier.com/editors/plagdetect.

The authors are obliged to participate in peer review process and be ready to provide corrections, clarifications, retractions and apologies when needed. All authors of a paper should have significantly contributed to the research.

The reviewers should provide objective judgments and should point out relevant published works which are not yet cited. Reviewed articles should be treated confidentially. The reviewers will be chosen in such a way that there is no conflict of interests with respect to the research, the authors and/or the research funders.

The editors have complete responsibility and authority to reject or accept a paper, and they will onh accept a paper when reasonably certain. They will preserve anonymity of reviewers and promote publication of corrections, clarifications, retractions and apologies when needed. The acceptance of a paper automatically implies the copyright transfer to the National Academy of sciences of the Republic of Kazakhstan.

The Editorial Board of the National Academy of sciences of the Republic of Kazakhstan will monitor and safeguard publishing ethics.

Правила оформления статьи для публикации в журнале смотреть на сайте:

www: nauka-nanrk.kz
ISSN 2518–1467 (Online),
ISSN 1991–3494 (Print)
http://www.bulletin–science.kz/index.php/en

Заместитель директор отдела издания научных журналов НАН РК *Р. Жәлиқызы*
Редакторы: *М.С. Ахметова, Д.С. Аленов*
Верстка на компьютере *Г.Д. Жадырановой*

Подписано в печать 30.06.2023.
Формат 60x881/8. Бумага офсетная. Печать - ризограф.
40,0 п.л. Тираж 300. Заказ 3.

Национальная академия наук РК
050010, Алматы, ул. Шевченко, 28, т. 272-13-19