© Ilienko A., Halata L., Dubchak O., Arieshkov M., Using a modified Merkle-Demgard hash function to increase authentication security // Ukrainian Scientific Journal of Information Security, 2024, vol. 30, issue 3, pp. 399-406. ©

DOI: 10.18372/2225-5036.30.20361

ПРОГРАМНИЙ МОДУЛЬ АВТЕНТИФІКАЦІЇ З ВИКОРИСТАННЯМ УДОСКОНАЛЕНОЇ СХЕМИ МЕРКЛЕ-ДЕМҐАРДА

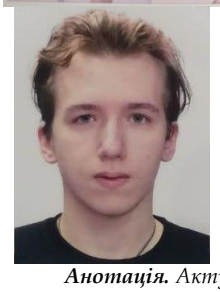
Анна Ільєнко, Лілія Галата, Олена Дубчак, Мирослав Арєшков

Державний університет «Київський авіаційний інститут»

ІЛЬЄНКО Анна Вадимівна, к.т.н., доцент
Рік та місце народження: 1987р., м.Венспілс, Латвія
Освіта: Національний авіаційний університет, 2009
Посада: завідувач кафедри кібербезпеки Державного університету «Київський авіаційний інститут»
Наукові інтереси: кібербезпека, інформаційна безпека, методи побудови та аналізу криптосистем, захищені інформаційні та комунікаційні системи і технології
Публікації: більше 200 наукових праць, серед них: монографії, посібник, статті у вітчизняних та міжнародних фахових виданнях, навчально-методичні комплекси дисциплін, матеріали і тези доповідей на конференціях, патенти на корисні моделі
E-mail: anna.ilienko@npp.kai.edu.ua
Orcid ID: *0000-0001-8565-1117*

ГАЛАТА Лілія Павлівна, доктор філософії за спеціальність 122 «Комп'ютерні науки»
Рік та місце народження: 1983р., м. Київ, Україна
Освіта: Національний авіаційний університет, 2005
Посада: доцент кафедри кібербезпеки Державного університету «Київський авіаційний інститут»
Наукові інтереси: кібербезпека, інформаційна безпека, проектування web-додатків, захищені інформаційні та комунікаційні системи і технології
Публікації: більше 70 наукових праць, серед них: статті у вітчизняних та міжнародних фахових виданнях, навчально-методичні комплекси дисциплін, матеріали і тези доповідей на конференціях, патенти на корисні моделі
E-mail: liliia.halata@npp.kai.edu.ua
Orcid ID: 0000-0002-7978-3954

ДУБЧАК Олена Вікторівна
Рік та місце народження: 1959р., м. Київ, Україна
Освіта: Київський інститут інженерів цивільної авіації, 1981
Посада: старший викладач кафедри кібербезпеки Державного університету «Київський авіаційний інститут»
Наукові інтереси: кібербезпека, інформаційна безпека, програмування, мережеві технології, захищені інформаційні та комунікаційні системи та технології
Публікації: понад 160 наукових праць, серед них: монографії, навчально-методичні видання, статті у вітчизняних та міжнародних фахових виданнях, навчально-методичні комплекси дисциплін, матеріали і тези доповідей на конференціях
E-mail: 3915922@npp.kai.edu.ua
Orcid ID: 0000-0001-9739-3960



АРЄШКОВ Мирослав Максимович
Рік та місце народження: 2002р., м. Горлівка, Донецька обл.
Освіта: Державний університет «Київський авіаційний інститут», 2024
Посада: здобувач кафедри кібербезпеки Державного університету «Київський авіаційний інститут»
Наукові інтереси: кібербезпека, інформаційна безпека, кібербезпека, інформаційна безпека, методи побудови та аналізу криптосистем.
E-mail: 6350028@stud.kai.edu.ua
Orcid ID: *0000-0001-8565-1117*

Анотація. *Актуальним завданням забезпечення безпеки персональних даних користувачів безлічі систем у всьому світі є вибір надійних методів захисту інформації, їх новизна, простота у впровадженні та оновленні, а також їх постійне вдосконалення. У цій статті проведено дослідження структур існуючих геш-функцій як складової процесу автентифікації, що у свою чергу є головною складовою систем захисту інформації. Основна увага приділяється розвитку структур у часі, загрозам, яким вони протистояли та шляхам забезпечення стійкості від різного роду атак. Запропонований новий погляд на розвиток структур побудови геш-функцій у системах автентифікації та створено рішення, що здатне конкурувати із існуючими стандартами при відносній простоті впровадження. Результат роботи може бути впроваджений до існуючих систем автентифікації організаціями та державними установами для підвищення рівня захисту інформації.*

Ключові слова *автентифікація, структура, модифікація, безпека, гешування, геш-функція, динамічна сіль, секретний перець.*

Постановка проблеми. Сучасні технології та існуючі системи захисту інформації забезпечують достатній рівень безпеки персональних даних, але складність цифрової інфраструктури зростає через різноманіття інтернет-платформ та онлайн-сервісів. Мобільні пристрої, інтернет речей, хмарні сервіси, сервери – усе це працює з персональними даними користувачів і потребує новітніх рішень з їх захисту.

Зловмисники також не стоять на місці і розробляють нові методи атак на системи автентифікації, що змушує діяти з ними наввипередки, створюючи нові схеми, структури, та цілі системи захисту інформації. Покращення вже існуючих методів може стати тим необхідним кроком, для відкриття чогось нового, більш стійкого до існуючих та можливих кіберзагроз.

Мета та постановка завдання

Метою даної статті є дослідження теоретичних основ застосування вдосконалених геш-функцій у процесах автентифікації користувачів, зокрема розгляду схеми Меркле-Демгарда з динамічною сіллю та секретним «перцем» для підвищення стійкості до атак грубою силою. Такий підхід спрямований на підвищення безпеки автентифікації користувачів шляхом унеможливлення попередньо обчислених атак (наприклад, атак райдужними таблицями) та зменшення ризику компрометації паролів у разі витоку даних. Це забезпечує унікальність і непередбачуваність гешу кожного паролю навіть при багаторазовому використанні, що значно підвищує захист від атак грубою силою. Запропонований програмний модуль може бути інтегрований у наявні інформаційні системи для забезпечення безпечної автентифікації користувачів. Його використання зменшить ризик несанкціонованого доступу до даних та облікових записів, що є критично важливим для сучасних кібербезпекових рішень.

Вступ. Автентифікація є однією з основних засобів захисту інформації та ключовою складовою системи безпеки. Дана процедура для користувача повинна бути простою, тому що по всьому світі від маніпуляцій даними у цифровому просторі залежать безліч людей, а ще більша кількість коротких сесій кожного з них навантажує ті чи інші системи й значно зменшує швидкість обробки необхідних користувачам даних , не кажучи вже про зручність користування. Одночасно з цим автентифікація має бути надійною, щоб запобігти помилок у роботі, а головне – захистити персональні дані користувачів.

Вибір та реалізація механізмів забезпечення цілісності та справжності інформаційних ресурсів у сучасних автоматизованих системах є одними з важливих етапів проектування та розробки підсистем захисту інформації. Це пов’язано з постійним зростанням послуг, які надаються різними мережними службами. Більшість послуг, які надаються при відсутності фіксованих мережевих адрес клієнтів та їх особливостей. У зв’язку з чим, ризик порушення цілісності та автентичності інформації збільшується.

Головними орієнтирами у розробці нових рішень забезпечення захисту інформації є правові засади [1-2], а також чітке розуміння термінології і вимог [3-4]. При модифікації вже існуючих рішень, необхідно розуміти шляхи розвитку даного питання та проблеми, що були вирішені у минулому [5], щоб правильно оцінити важливість внесення змін та порівняти результат. Не менш важливою є оцінка існуючих загроз, шляхів їх попередження та протидії у майбутньому [6-8].

Виклад основного матеріалу дослідження

Гешування є надійним інструментом забезпечення безпеки доступу до персональних даних і відіграє важливу участь у процедурі автентифікації безлічі систем [9].

Під поняттям геш-функції будемо розуміти односторонню криптографічну функцію, яка приймає вхідні дані довільного розміру та перетворює їх на фіксований за довжиною хеш-код. Гешування паролів є важливою складовою систем захисту інформації. Для підвищення безпеки при гешуванні може використовуватись сіль — випадкове значення, яке додається до пароля перед його хешуванням. Це допомагає захистити дані від атак грубою силою та атак на основі готових таблиць хешів (rainbow tables). Задля підвищення стійкості зберігання паролів та інших конфіденційних даних до атак грубою силою використовуються два типи солей — статична та динамічна.

Під поняттям статична сіль будемо розуміти фіксоване значення (константа), яке додається до кожного пароля перед його передачею до геш-функції. Основна перевага статичної солі — простота реалізації та відсутність необхідності зберігати додаткові дані для кожного запису. Зазвичай значення статичної солі визначається адміністратором системи один раз і прописується в конфігураційних файлах або коді додатка [10].

Однак цей підхід має суттєвий недолік: однакове значення солі для всіх паролів. Це робить систему вразливою, оскільки зловмисник може заздалегідь підготувати таблиці зі статичною сіллю для поширених паролів, що ставить під загрозу безпеку всієї системи. Через цей недолік статична сіль не вважається достатньо надійною для використання у сучасних системах безпеки, хоча її все ще застосовують у пошукових механізмах баз даних.

Динамічна сіль, на відміну від статичної, є унікальним значенням, яке генерується для кожного

користувача або навіть для кожного пароля окремо. Зазвичай у базах даних її зберігають разом із паролем або його гешем. Значення динамічної солі зазвичай генерується випадковим чином під час створення або оновлення пароля, наприклад, за допомогою криптографічно стійких генераторів випадкових чисел (CSPRNG).

Основна перевага динамічної солі — підвищена стійкість до атак із використанням попередньо обчислених таблиць (rainbow tables). При її застосуванні кожен пароль отримує унікальну сіль, яка може змінюватися під час роботи системи або при виконанні користувачем певних дій. Це значно ускладнює компрометацію паролів, оскільки для кожного запису зловмисникові потрібно виконувати окремі обчислення, що робить атаку ресурсомісткою.

Серед недоліків динамічної солі — необхідність збереження додаткових даних для кожного запису, що збільшує обсяг використаного місця у базах даних. Попри цей недолік, динамічна сіль є стандартом для зберігання паролів, оскільки забезпечує значно вищий рівень захисту від атак.

Разом із динамічною сіллю нерідко використовується статична змінна, що називається секретним «перцем» (pepper). На відміну від солі, вона додається на початок вхідного повідомлення та зберігається окремо від бази даних із паролями. Це, своєю чергою, забезпечує вищий рівень захисту, оскільки навіть у разі компрометації бази даних зловмисники не зможуть скористатися отриманою інформацією. Зазвичай значення секретного «перця» є глобальним для всієї системи та визначається адміністратором або автоматично генерується під час розгортання програмного забезпечення. Його зберігають у захищеному сховищі, наприклад у файлах конфігурації, апаратних модулях безпеки (HSM) або змінних середовища, щоб ускладнити його витік.

Такі відомі компанії, як Google, Microsoft та Apple, використовують у своїх системах комбінацію динамічної солі із секретним перцем для забезпечення максимально можливого рівня захисту персональних даних. Проте методи їхнього використання не є публічно доступною інформацією.

Далі розглянемо основні схемо формування геш-функцій, які використовуються для автентифікації та контролю цілісності. На сьогодні схема Меркле-Демгарда є найвідомішою основою для побудови криптографічних геш-функцій. До таких функцій належать MD5, SHA-1 та сімейство SHA-2, з якого для цієї роботи було обрано алгоритм SHA-256. Ця схема дозволяє створювати геш-значення для повідомлень будь-якої довжини, ітеративно застосовуючи блочний алгоритм.

Схема Меркле-Демгарда має такі властивості, як стійкість до колізій і захищеність від атак типу подовженого повідомлення. Однак її значним недоліком є вразливість до атак подовженого гешу, якщо геш-функція реалізована без модифікацій, таких як, наприклад, додавання доменних розширень [11].

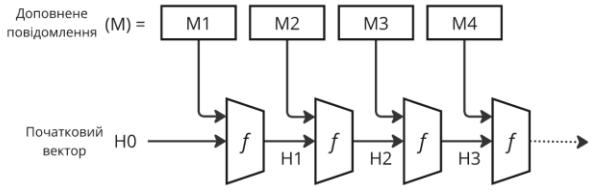


Рис.1. Схема Меркле-Демгарда

Наступною до розгляду візьмемо структуру Рабіна – що також є методом гешування, але, на відміну від минулої схеми, у побудові геш-функцій забезпечує криптографічну стійкість використовуючи блочні шифри. Разом із схемою Меркле-Демгарда ними використовується ітеративний підхід, тобто надіслане повідомлення розбивається на блоки, після чого по черзі обробляються.

Блочний шифр відіграє роль компресійної функції, де кожен блок шифрується прямим типом, використовуючи частину вхідного повідомлення як ключ, а попереднє геш-значення як вхідні дані.

Маємо наступну формулу:

$$H_{i+1} = E_{M_i}(H_i),$$

де H_i– ітеративне геш-значення; E – функція шифрування; M_i – і-тий блок доповненого повідомлення.

Структура Рабіна наведена на рис.2:

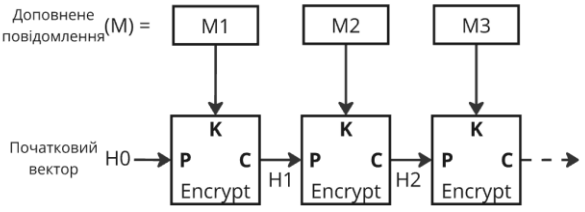


Рис.2. Структура Рабіна

Через свою структуру схема Рабіна в деяких аспектах є більш гнучкою, оскільки здатна включати зміни в порядку блоків або їх комбінації. При обробці окремих блоків вхідного повідомлення використовуються різні ключі, що, своєю чергою, підвищує стійкість до диференційного криптоаналізу.

Наступна структура має багато спільного зі схемою Рабіна. Схема Девіса-Мейєра також подає кожен блок вхідного повідомлення до блокового шифру як ключ, а отримане раніше геш-значення — як нове повідомлення для шифрування. Однак далі

вона має суттєву відмінність: наступним кроком виконується операція XOR вихідного геш-значення з попередньо отриманим геш-значенням (рис.3) [12]:

$$H_{i+1} = E_{M_i}(H_i) \oplus H_i,$$

де H_i – ітеративне геш-значення; E – функція шифрування; M_i – i-тий блок доповненого повідомлення.

Якщо порівняти дану структуру із структурою Рабіна, то варіант структури Девіса-Мейєра є функціонально кращим. Додаткова логічна операція XOR отримуваного геш-значення із попередніми результатами гешування значно підвищує стійкість до колізій і захищає від атак «зустрічі посередині».

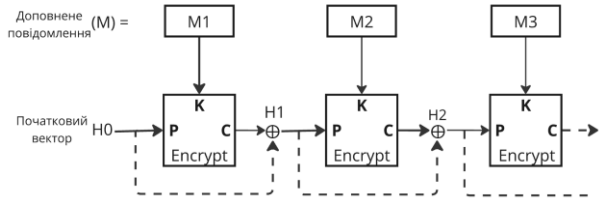


Рис.3. Структура Девіса-Мейєра

Схеми Меркле-Демгарда та Девіса-Мейєра схожі між собою лише тим, що використовують ітеративний підхід до побудови геш-функцій, але не можна сказати, що друга структура є прямим вдосконаленням першої. На відміну від Меркле-Демгарда від атак подовження гешу структура Девіса-Мейєра захищена. Дана структура є кращим варіантом застосування для цифрових підписів чи захисту комунікаційних каналів, тобто у випадках, коли потрібен високий рівень криптографічної стійкості. У ній використовується блочний шифр, що робить її особливо необхідною для спеціалізованих криптографічних протоколів, де важлива додаткова безпека. Далі розглянемо і порівняємо структуру Матіаса–Мейєра–Озіса, що є модифікованою версією стректури Девіса–Мейєра. Її принципово важливою відмінністю є специфічний спосіб обчислення геш-значення (рис.4). Основою цієї структури, спадково від попередника, є використання блочного шифру. Ключовою ж модифікацією є проведення логічної операції XOR блоку вхідного повідомлення із результатом шифрування:

$$H_{i+1} = E_{H_i}(M_i) \oplus M_i,$$

де H_i – ітеративне геш-значення; E – функція шифрування; M_i – і-тий блок доповненого повідомлення.

Кожен блок у структурі Матіаса–Мейєра–Озіса шифрується зворотнім типом шифрування, використовуючи у якості ключ попереднє геш-значення, а вхідними даними блочного шифру стає частина вхідного повідомлення.

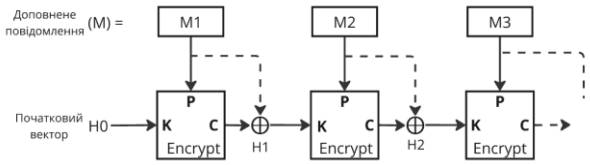


Рис.4. Структура Матіаса-Мейєра-Озіса

До підвищення стійкості вихідних геш-значень призвело рішення внести зміни у операцію XOR. У такому випадку структура є менш вразливою до атак на структуру компресійної функції. Схема Міагуччи-Пренеля, як і минулі розглянуті схеми, є варіантом побудови геш-функції на основі блочного шифру (рис.5). Її можна називати модифікацією структури Матіаса-Мейєра-Озіса. За допомогою зворотнього типу гешування блоки даних шифруються окремо один від одного, а роль ключа при цьому виконує попереднє геш-значення. Результат гешування на шляху до завершення геш-функції має пройти операцію XOR одразу і з блоком доповненого повідомлення, і з попереднім геш-значенням [13]:

$$H_{i+1} = E_{H_i}(M_i) \oplus M_i \oplus H_i,$$

де H_i – ітеративне геш-значення; E – функція шифрування; M_i – і-тий блок доповненого повідомлення [14-15].

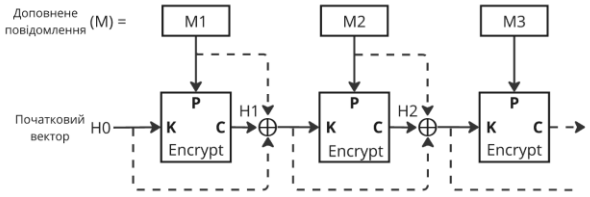


Рис.5. Структура Міагуччи-Пренеля

Поєднання XOR із обома вхідними даними (геш і блок повідомлення) забезпечує стійкість до атаки "зустрічі посередині", а також значно підвищує складність знаходження двох різних вхідних повідомлень з ідентичним геш-значенням. Крім цього, використання XOR із двома різними вхідними значеннями додатково заплутує зловмисника, що ускладнює проведення криптоаналізу. Виконання додаткових XOR операцій для обох вхідних значень знижує швидкість гешування порівняно з іншими розглянутими схемами, і особливо із класичною структурою Меркле–Демгарда.

Аналіз схем формування геш-функцій дозволив визначити підходи до їх удосконалення. Зокрема, було розглянуто модифіковану структуру Меркле-Демгарда, яка використовує динамічну сіль і секретний «перець» для підвищення стійкості до атак. Ця схема представлена на рис. 6.

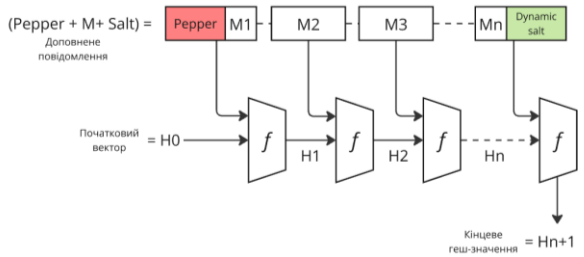


Рис.6. Модифікована структура Меркле-Демгарда

$$H_i = f(H_{i-1}\ (\mathrm{P} + (M_i + S))_i)$$

де Hi – ітеративне геш-значення; f – компресійна функція; Mi – і-тий блок доповненого повідомлення; S – динамічна сіль; P – секретний перець.

Далі наведемо зведену таблицю порівняння структур формування геш-функцій (табл. 1).

Таблиця 1

Порівняння структур формування геш-функцій

Структура	Компресійна функція	Формула розрахунку компресії	Захист від атак	Пошук колізії диф. криптоаналізом
Меркле–Демгарда	Будь-яка компресійна функція *f(H, M)*	$H_i = f(H_{i-1}, M_i)$	Захист від атак подовження повідомлення	Залежить від конкретної компресійної функції, зазвичай складно реалізувати
Рабіна	Прямий порядок, шифрування гешу *H* з ключем *M*	$H_{i+1} = E_{Mi}(H_i)$	Захист від атак подовження гешу	Може бути уразлива, оскільки відсутнє додаткове змішування результату
Девіса–Мейєра	Прямий порядок, шифрування гешу *H* з ключем *M*	$H_{i+1} = E_{Mi}(H_i) \oplus H_i$	Захист від атаки "зустрічі посередині"	Приблизно 2^{85} завдяки *XOR* з попереднім гешем
Матіаса–Мейєра–Озіса	Зворотній порядок, шифрування повідомлення *M* з ключем *H*	$H_{i+1} = E_{Hi}(M_i) \oplus M_i$	Підвищений захист за рахунок *XOR* з *M*	Приблизно 2^{85}, додатковий захист від *XOR* з *M*
Міагуччи-Пренеля	Зворотній порядок, шифрування повідомлення *M* з ключем *H*	$H_{i+1} = E_{Hi}(M_i) \oplus M_i \oplus H_i$	Підвищений захист за рахунок *XOR* з *M* та *H*	Близько до 2^{128} завдяки *XOR* з обома компонентами

Дане рішення призводить до того, що для формування кінцевого геш-значення необхідно здійснити більшу кількість операцій. На відміну від класичної структури Меркле-Демґарда, у нашому випадку функція гешування додатково має визначити статичне значення «перцю», що необхідно додати на початок уже доповненого вхідного повідомлення, а також згенерувати динамічну сіль, що додається у кінець. Крім цього наявність динамічної солі у алгоритмі не тільки збільшує час формування складного початкового значення перед передачею того до геш-функції, але і зобов'язує запускати її для генерації нового геш-значення при кожному новому сеансу користувача до системи.

Далі наведемо блок-схему процесу автентифікації з впровадженням удосконаленої схеми Меркле-Демґарда (рис.7). Після аналізу міри модифікації структури Меркле-Демґарда, впливу солей на її захищеність та продуктивність, стає очевидною необхідність порівняння цієї структури з іншими, які можуть бути використані як основа для методів автентифікації (табл. 2).

Таблиця 2

Порівняння ефективності структур як основи для методу автентифікації.

Структура	Швидкість виконання 10 тис. циклів, мс	Довжина оброблюваного блоку, біт	Рівень захищеності вихідного геш-значення
Меркле-Демгарда	0.033	256	Середній
Рабіна	0.052	64/128	Середній
Девіса-Мейєра	0.066	128	Середньо-високий
Матіаса-Мейєра-Озіса	0.082	128	Високий
Міагуччи-Пренеля	0.098	128	Дуже високий
Модифікація Меркле-Демгарда	0.073	512	Високий

У таблиці 2 кількість операцій за цикл відображає операції, необхідні для формування геш-значення на кожному етапі. Всі розглянуті структури, окрім схеми Меркле-Демгарда, використовують блочне шифрування, яке є швидшим за асиметричне шифрування, що потребує додаткових обчислювальних ресурсів для обробки вхідних значень.

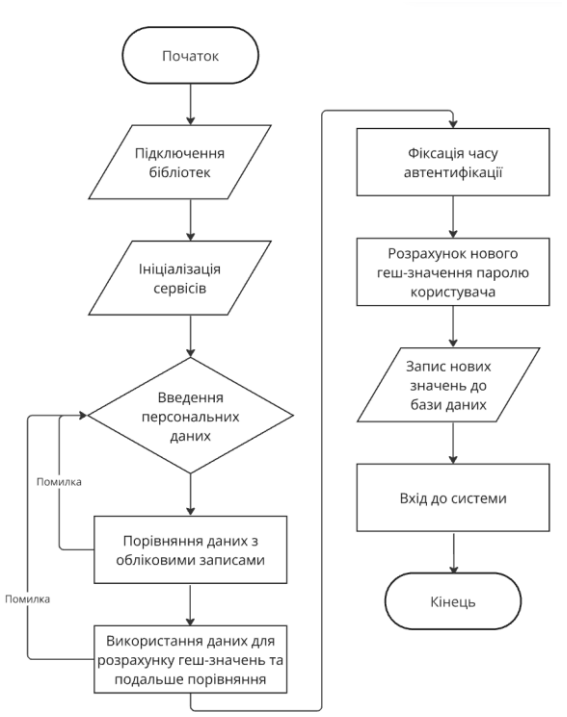


Рис.7. Блок-схема процесу автентифікації

Також на рис. 8 бачимо результат програмної реалізації розробленого рішення, а саме генерацію динамічної солі, секретного «перцю», їх подальше об'єднання із паролем у вхідне повідомлення, що проходить геш-функцію і перетворюється на геш-значення фіксованої довжини.

```
New login time = 13102024133405
pepper_2 = F
Pass_2 = 12345
last login_2 = 13102024133405
New pre_hash = F1234513102024133405
new_hash = c53950ba656b4819b6c5d1e682fe2abb514cfe97a7c84add558999aec7a948ec
```

Рис.8. Процес генерації геш-значення

Проведений аналіз ефективності різних структур для автентифікації дозволяє виділити ключові аспекти їхньої продуктивності та рівня захисту. При порівнянні за швидкістю виконання отримано такі результати: Найшвидшою є структура Меркле-Демґарда (0.033 мс), що робить її ефективною для застосувань із високими вимогами до швидкодії; Рабіна (0.052 мс) і Девіса-Мейєра (0.066 мс) мають помірну швидкість; Матіаса-Мейєра-Озіса та Міагуччи-Пренеля працюють повільніше (0.082 мс і 0.098 мс відповідно), що зумовлено складнішою структурою. Модифікація Меркле-Демґарда (0.073 мс) трохи поступається оригінальній версії за швидкістю, проте забезпечує вищий рівень захисту.

Модифікація Меркле-Демґарда використовує 512-бітові блоки, що підвищує її стійкість до криптографічних атак. Завдяки цьому вона має високий рівень захисту, а підтримка довгих блоків робить її перспективною для застосування в системах із підвищеними вимогами до криптографічної стійкості. Модифікована структура Меркле-Демґарда має найкращий баланс між швидкодією та захищеністю. Хоча запропонована схема працює у 2.36 раза повільніше, ніж оригінальний Меркле-Демґард (за рахунок додавання динамічної солі та секретного «перцю»), вона забезпечує вищий рівень захисту при автентифікації кожного користувача. Крім того: вона підтримує 512-бітові блоки, що в 4 рази більше, ніж у Рабіна, і в 2 рази більше, ніж у класичній схемі Меркле-Демґарда. Має високий рівень захисту, що дозволяє їй конкурувати зі схемами Матіаса-Мейєра-Озіса та Міагуччи-Пренеля забезпечуючи швидкість виконання циклів у 1.12–1.34 рази швидше, при цьому збільшуючи розмір оброблюваного блоку в 4 рази.

Рівень захищеності геш-значення оцінюється за стійкістю до атак, таких як атака «Дня народження» та диференційний криптоаналіз. Удосконалена схема Меркле-Демґарда для формування геш-начененя має кілька важливих переваг: 1) Модифікація Меркле-Демґарда потребує меншої кількості шифрувань, оскільки вона додає сольові значення та секретний перець, що знижує обчислювальні витрати та дозволяє швидше генерувати хеші. 2) Динамічна сіль підвищує стійкість до атак типу райдужних таблиць. Секретний перець ще більше покращує стійкість до атак з відкритими текстами, оскільки він невідомий зловмиснику. 3) Використання як солі, так і перцю ускладнює атаки навіть при доступі до бази даних з геш-значеннями та динамічною сіллю, оскільки перець зберігається окремо. 4) Модифікація Меркле-Демґарда є більш стійкою до колізій, оскільки динамічна сіль ускладнює створення двох різних повідомлень з однаковим геш-значенням. Структури, що використовують шифрування, можуть бути менш стійкими до колізій через фіксовану функцію блочного шифру. 5) Відсутність складних операцій шифрування робить модифікацію Меркле-Демґарда простішою для реалізації, що корисно для швидкої обробки великих обсягів даних без витрат на складні криптографічні операції. Таким чином, модифікація Меркле-Демґарда може бути рекомендована для критично важливих систем автентифікації, де потрібен високий рівень захисту при відносно невисоких витратах ресурсів.

Висновки. У даній статті представлено комбіновану концепцію реалізації схеми автентифікації на базі модифікованої схеми Меркле-Демґарда. У процесі дослідження було проведено аналіз сучасних методів автентифікації, зокрема на основі алгоритмів гешування, що дозволило визначити загальні моделі та вимоги для подальшої

розробки модифікації. Також були проаналізовані сучасні алгоритми гешування для вибору оптимального варіанта під потреби безпечної автентифікації, що й дозволило обрати для роботи схему Меркле-Демгарда. Запропоновано модифіковану схему формування геш-функції на основі цієї структури з використанням динамічної солі та секретного «перцю».

Головна перевага вдосконаленої геш-функції – поєднання швидкості та підвищеного рівня захисту без значного ускладнення процесу обчислень, що дозволяє її інтегрувати в сучасні системи автентифікації, блокчейн-технології, криптографічні протоколи, захист даних у хмарних середовищах та IoT-рішення. Завдяки використанню динамічної солі та секретного «перцю» забезпечується стійкість до атак на довжину повідомлення, перебору хешів та підбору колізій, що робить її перспективною для застосування в корпоративних інформаційних системах, фінансових технологіях і системах моніторингу кіберзагроз.

Запропоноване рішення потребує більшої кількості операцій для формування кінцевого геш-значення порівняно з класичною схемою Меркле-Демгарда. Функція гешування визначає статичний «перець» та генерує динамічну сіль, що додається до вхідного повідомлення. Хоча складність структури зросла, сучасні технології забезпечують достатню швидкість виконання, що виправдовує додаткові витрати часу. Динамічна сіль підвищує безпеку, ускладнюючи атаки грубою силою та словникові атаки, оскільки забезпечує унікальність гешу для кожного користувача.

Генерація солі потребує додаткових обчислювальних ресурсів, але для сучасних систем це не є критичною проблемою. Однак у великих масштабованих системах може виникнути додаткове навантаження, особливо при високій кількості запитів. Використання динамічної солі збільшує обсяг даних для зберігання та може потребувати оптимізації бази даних для збереження швидкості доступу.

Попри можливе зниження продуктивності в системах із великим навантаженням, впровадження запропонованої модифікації значно підвищує рівень безпеки. Модуль автентифікації на базі модифікованої структури Меркле-Демгарда запобігає атакам повторного використання даних завдяки унікальним токенам у вигляді динамічної солі, що автоматично змінюється при кожній автентифікації. Крім того, модифікована структура гешування захищає від перехоплення даних і підвищує захист від атак на основі пошуку прообразів.

Список літератури

[1] Закон України «Про Інформацію». URL:https://zakon.rada.gov.ua/laws/main/2657-12#Text

[2] Манжай О. В., Манжай І. А. Правові засади захисту інформації. Підручник. Харків, 2020. URL: https://univd.edu.ua/science-issue/issue/4315.

[3] Автентифікація. Інформаційні технології. URL: https://vue.gov.ua/Автентифікація_(інформаційні_технології)

[4] Технічні статті. Аутентифікація, авторизація та ідентифікація. URL: https://training.qatestlab.com/blog/technical-articles/authentication-authorization-and-identification.

[5] Євсеєв С. П., Йохов О. Ю., Король О.Г. Гешування даних в інформаційних системах: монографія. Харків: ХНЕУ, 2013. 312 с.

[6] Gavin Wright. Dictionary attack. URL: https://www.techtarget.com/searchsecurity/definition/dictionary-attack#:~:text=A%20dictionary%20attack%20uses%20a,that%20can%20vary%20by%20region.

[7] Dr. Nandhini Vineeth. Cryptography and Network Security. URL: https://bmsce.ac.in/Content/CS/NV_CNS_UNITV.pdf

[8] Rainbow table attack and how does it work. URL: https://nordvpn.com/ru/blog/what-is-rainbow-table-attack/?msockid=0169d861d2be65273675ca1cd3be6484

[9] A. Menezes, P. van Oorschot, and S. Vanstone Handbook of Applied Cryptography. Hash Functions and Data Integrity, CRC Press, 1996. URL: https://cacr.uwaterloo.ca/hac/about/chap9.pdf

[10] Salt in hashing. URL: https://www.nexusgroup.com/what-is-a-salt-hash/

[11] Структура Меркле-Демгарда. URL: https://uk.wikipedia.org/wiki/Будова_Меркла-Демгарда

[12] Davies-Meyer Construction and Collision Resistance URL: https://www.csa.iisc.ac.in/~arpita/Cryptography15/CT5.1.pdf

[13] Cryptographic Hash Functions and Attacks URL:https://pdfs.semanticscholar.org/b312/007ceb0918779c7cc4453489b6fc02bb0462.pdf

[14] Biserial Miyaguchi–Preneel Blockchain-Based Ruzicka-Indexed Deep Perceptive Learning for Malware Detection in IoMT URL: https://www.mdpi.com/1424-8220/21/21/7119

[15] Miyaguchi–Preneel Snefru Cryptographic Blockchain URL: https://www.ijirmps.org/papers/2022/3/1551.pdf

УДК 004.056.5(045)

Ilienko A., Halata L., Dubchak O., Arieshkov M., Using a modified Merkle-Demgard hash function to increase authentication security

Abstract. *The current task of ensuring the security of personal data of users of many systems around the world is the selection of reliable methods of information protection, their novelty, ease of implementation and updating, as well as their constant improvement. This article studies the structures of existing hash functions as a component of the authentication*

process, which in turn is the main component of information protection systems. The main focus is on the development of structures over time, the threats they have faced, and ways to ensure resilience against various types of attacks. A new perspective on the development of hash function construction structures in authentication systems is proposed and a solution is created that can compete with existing standards while being relatively easy to implement. The result of the work can be implemented into existing authentication systems by organizations and government agencies to increase the level of information protection.

Keywords: *authentication, structure, modification, security, hashing, hash function, dynamic salt, secret pepper.*

Ільєнко Анна Вадимівна, к.т.н., доцент, завідувач кафедри кібербезпеки Державного університету «Київський авіаційний інститут»
Ilienko Anna, Ph.D., Associate Professor, Head of the Department of Cybersecurity, State University «Kyiv Aviation Institute»

Галата Лілія Павлівна, доктор філософії, доцент кафедри кібербезпеки Державного університету «Київський авіаційний інститут»
Halata Liliia, PhD, associate professor of the Department of Cybersecurity, State University «Kyiv Aviation Institute»

Дубчак Олена Вікторівна, старший викладач кафедри кібербезпеки Державного університету «Київський авіаційний інститут»
Olena Dubchak, senior lecturer of the Department of Cybersecurity, State University «Kyiv Aviation Institute»

Арєшков Мирослав Максимович, магістр кафедри кібербезпеки Державного університету «Київський авіаційний інститут»
Arieshkov Myroslav, master of the Department of Cybersecurity, State University «Kyiv Aviation Institute»