УДК 004.056 DOI https://doi.org/10.35546/kntu2078-4481.2025.4.3.6

Т. О. ГРІНЕНКО
кандидат технічних наук, доцент,
доцент кафедри безпеки інформаційних технологій
Харківський національний університет радіоелектроніки
ORCID: 0000-0002-8251-8991

Д. М. МАЛЯРОВА
здобувач магістерського рівня кафедри безпеки інформаційних технологій
Харківський національний університет радіоелектроніки
ORCID: 0009-0003-2506-6970

О. П. НАРЄЖНІЙ
кандидат технічних наук,
доцент кафедри кібербезпеки інформаційних систем, мереж і технологій
Харківський національний університет імені В. Н. Каразіна
ORCID: 0000-0003-4321-0510

ПРОТОТИП ВЕБ-СЕРВІСУ КВАНТОВОГО ГЕНЕРАТОРА ВИПАДКОВИХ ЧИСЕЛ ДЛЯ ЗАХИЩЕНОГО ОБМІНУ ПОВІДОМЛЕННЯМИ

У даній роботі особливу увагу приділено криптографічному механізму HMAC як одному з найбільш ефективних засобів забезпечення автентичності та цілісності повідомлень. Досліджено принципи функціонування HMAC, описано його модифікацію HMAC-SHA-256, який визнано сучасним стандартом у безпечному обміні даними. Порівняння різних реалізацій HMAC виявило баланс між стійкістю та продуктивністю, що робить його придатним для використання у веб-сервісах з високими вимогами до продуктивності та безпеки, а саме, у веб-сервісі QRNG.

Обґрунтований вибір інструментів та середовища розробки прототипу веб-сервісу QRNG, наданий опис його основних компонентів та побудовані блок-схеми логіки автентифікації користувачів та обміну приватними повідомленнями. Надані результати тестування ключових функцій системи захисту за різними сценаріями, включаючи захист від підміни, помилкової автентифікації та обробку виняткових ситуацій.

Ефективне забезпечення захищеного обміну повідомленнями у веб-сервісі QRNG потребує комплексного підходу, що включає впровадження надійних криптографічних методів, надійне управління сесіями та ключами, а також застосування сучасних систем автентифікації. Такий підхід дозволяє не лише забезпечити високий рівень безпеки, але й підтримувати високу продуктивність веб-сервісу при високих навантаженнях, що є критично важливим для його стабільної роботи в сучасному цифровому середовищі.

Комплексний підхід щодо побудови захищеного обміну повідомленнями у веб-сервісі є необхідним для створення надійного і безпечного веб-сервісу QRNG, який буде здатним протистояти сучасним кіберзагрозам і забезпечувати безпеку даних.

Ключові слова: *QRNG, HMAC, SHA-256, TLS, WEB, автентифікація, веб-сервіс, захищений обмін, квантовий генератор випадкових чисел, клієнт-серверна архітектура, код автентифікації повідомлень, парольна автентифікація.*

T. O. HRINENKO
PhD, Associate Professor,
Associate Professor at the Department of Information Technologies Security
Kharkiv National University of Radio Electronics
ORCID: 0000-0002-8251-8991

D. M. MALIAROVA
Master's Degree Student at the Department of Information Technology Security
Kharkiv National University of Radio Electronics
ORCID: 0009-0003-2506-6970

O. P. NARIEZHNII
PhD, Associate Professor at the Department of Cybersecurity
of Information Systems, Networks and Technologies
V. N. Karazin Kharkiv National University
ORCID: 0000-0003-4321-0510

PROTOTYPE OF A QUANTUM RANDOM NUMBER GENERATOR WEB SERVICE FOR SECURE MESSAGE EXCHANGE

This work pays special attention to the HMAC cryptographic mechanism as one of the most effective means of ensuring message authenticity and integrity. The principles of HMAC operation are examined, and its modification HMAC-SHA-256 – recognized as a modern standard in secure data exchange – is described. A comparison of different HMAC implementations revealed a balance between security and performance, which makes it suitable for use in web services with high performance and security requirements, including the QRNG web service.

The choice of tools and the development environment for the QRNG web service prototype is substantiated, and its main components are described. Block diagrams of the user authentication logic and private messaging processes are presented. The results of testing the system's key security functions under various scenarios are provided, including protection against spoofing, false authentication, and the handling of exceptional situations.

Effective secure message exchange in the QRNG web service requires a comprehensive approach that includes the implementation of reliable cryptographic methods, secure and reliable session and key management, as well as the use of modern authentication systems. Such an approach not only ensures a high level of security but also maintains the high performance of the web service under heavy loads, which is critically important for its stable operation in the modern digital environment.

A comprehensive approach to building secure message exchange within the web service is essential for creating a reliable and secure QRNG platform capable of withstanding modern cyber threats and ensuring data security.

Key words: *QRNG, HMAC, SHA-256, TLS, web, authentication, web service, secure communication, client–server architecture, message authentication code, password authentication.*

Постановка проблеми

Веб-сервіси стали ключовими інструментами обміну даними, організації бізнес-процесів і комунікації. Сучасні веб-сервіси обробляють значний обсяг персональних, фінансових та корпоративних даних, що робить їх привабливою мішенню для кіберзлочинців. Найпоширенішими атаками залишаються SQL-ін'єкції, міжсайтовий скриптинг (Cross-Site Scripting, XSS), підробка міжсайтових запитів (Cross-Site Request Forgery, CSRF), атака типу «людина посередині» (Man-In-The-Middle, MITM), атаки грубої сили, атаки на відмову в обслуговуванні (Distributed Denial of Service, DDoS-атаки) й інші [1].

Застосування веб-сервісів квантового генератора випадкових чисел (Quantum Random Number Generator, QRNG) стало критично важливим для забезпечення безпеки банківських систем, інтернету речей та інших високочутливих платформ. Наприклад, при шифруванні даних QRNG виступають як надійний інструмент для створення криптографічних ключів. Однак їх інтеграція в реальні системи породжує нові ризики, пов'язані з потенційними атаками на апаратному та програмному рівнях [2].

Ключовими проблемами, з якими стикаються веб-сервіси QRNG в контексті безпеки, є захист від атак типу перехоплення повідомлень (MITM-атаки) та повторного відтворення повідомлень (replay attacks). MITM-атаки можуть призвести до серйозних наслідків, оскільки перехоплені повідомлення можуть бути підроблені або використані для несанкціонованого доступу до конфіденційних даних. Повторне відтворення повідомлень (replay attacks), при якому зловмисники намагаються повторно використати перехоплені дані, є ще однією важливою загрозою, що вимагає ефективних заходів захисту.

Також серйозну загрозу для веб-сервісів QRNG становлять атаки на геш-функції, особливо при використанні алгоритмів, що можуть бути вразливими до колізій. Зловмисники можуть модифікувати геш-значення для зміни переданих даних, що ускладнює процес забезпечення цілісності інформації. Зокрема, такі атаки можуть бути спрямовані на порушення надійності системи автентифікації та перевірки цілісності даних.

Також важливою загрозою є слабкість традиційних методів автентифікації, таких як парольна автентифікація, що не здатні забезпечити належний рівень безпеки у випадку атаки зловмисників. Простота використання паролів і ймовірність їх компрометації або витоку робить цей метод уразливим до атак типу brute-force, фішинг, а також до атак на системи зберігання паролів. Сучасні зловмисники можуть також застосовувати атаки на системи зберігання паролів, що дозволяє їм отримувати доступ до захищених ресурсів без необхідності обходити інші механізми безпеки. У зв'язку з цим, виникає необхідність у впровадженні більш складних методів автентифікації, таких як багатофакторна автентифікація або коди автентифікації повідомлень, які гарантують більш високий рівень захисту.

Однак, навіть при використанні сучасних криптографічних алгоритмів, існує ще одна важлива проблема – управління секретними ключами. Ключі, що використовуються для криптографічного захисту даних, повинні бути надійно захищені від витоку або компрометації, оскільки компрометація ключа може призвести до втрати всієї безпеки системи. У зв'язку з цим, важливим є впровадження надійних механізмів для зберігання і обміну ключами, а також створення засобів їхнього автоматичного оновлення, щоб мінімізувати ризики, пов'язані з їх компрометацією.

Іншим важливим аспектом є забезпечення продуктивності веб-сервісу QRNG у високонавантажених середовищах. Використання складних криптографічних операцій може негативно впливати на швидкість роботи сервісу, що є критичним в умовах високих навантажень і масштабованих рішень (наприклад, у веб-сервісах з великою кількістю одночасних сесій). Тому важливо знаходити баланс між високим рівнем захисту і необхідною продуктивністю системи, що дозволяє підтримувати її ефективність навіть при значних навантаженнях.

Враховуючі вказані загрози, основними напрямками, які потребують вирішення, для забезпечення захищеного обміну повідомленнями у веб-сервісі QRNG, є:

– вибір криптографічних алгоритмів, що є стійкими до атак на колізії та перехоплення даних. Вони повинні бути надійними та ефективними для використання у веб-сервісах, де критично важлива продуктивність;

– безпечне управління ключами, яке охоплює зберігання, обмін і оновлення ключів. Надійні механізми захисту ключів є важливою умовою забезпечення високого рівня безпеки;

– забезпечення продуктивності при високих навантаженнях, де необхідно балансувати між швидкістю виконання криптографічних операцій та рівнем захисту;

– розробка заходів протидії MITM-атакам і підробці повідомлень, які включають використання сучасних методів автентифікації, криптографічних механізмів перевірки цілісності і збереження конфіденційності при передачі даних.

Вирішення цих проблем потребує комплексного підходу, що поєднує технічні та організаційні заходи для забезпечення надійного захисту даних у веб-сервісі QRNG.

Аналіз останніх досліджень і публікацій

Аналіз наукових публікацій [1-5] та стандартів [6-10] показує, що проблема захищеного обміну повідомленнями у веб-сервісах залишається надзвичайно актуальною, особливо через зростаючу кількість кібератак. Аналіз відповідних стандартів дозволяє сформувати архітектуру сервісу QRNG з урахуванням актуальних вимог до захисту даних, зокрема щодо конфіденційності, цілісності та автентичності.

Протокол WebSocket, описаний у RFC 6455 [6] є основою для реалізації двостороннього зв'язку між клієнтом і сервером. Він підтримує постійне TCP-з’єднання, завдяки чому обмін повідомленнями відбувається без повторного встановлення HTTP-запитів, що суттєво знижує затримки під час передачі повідомлень. Оскільки базова версія WebSocket не має вбудованого шифрування чи автентифікації, у веб-сервісі передбачається використання захищеного з’єднання wss://, яке базується на протоколі TLS. Це дозволяє забезпечити конфіденційність і цілісність даних при їх передачі мережею, що є обов’язковою умовою для протидії атакам типу «людина посередині».

Міжнародний стандарт ISO/IEC 27002:2022 [7] надає рекомендації щодо впровадження контролю інформаційної безпеки, включаючи використання криптографічних методів для забезпечення цілісності та автентичності даних. У межах захищеного обміну повідомленнями актуальним є застосування механізму HMAC, що дозволяє перевірити, чи було повідомлення змінено під час передачі. Крім того, можливе збереження унікального ідентифікатора користувача у локальному сховищі, що дозволяє підтримувати сесійну взаємодію з сервером.

НД ТЗІ 2.5-010-03 [8] встановлює вимоги до технічних та організаційних заходів захисту інформації WEB-сторінки в мережі Інтернет. Нормативний документ містить вимоги до захищеності інформації та визначає необхідність використання криптографічних засобів при обміні даними та впровадження механізмів автентифікації, що повністю відповідає вимогам до безпечного обміну повідомленнями.

Федеральний стандарт FIPS 198-1 [9] визначає механізм HMAC як надійний спосіб автентифікації повідомлень на основі криптографічної геш-функції та секретного ключа. Додатково, спеціальна публікація NIST SP 800-224 [10] конкретизує технічні вимоги до реалізації HMAC, включаючи вимоги до довжини ключа (не менше 128 біт), використання сучасних геш-функцій, зокрема SHA-256, та неприпустимість обрізання тегу автентифікації.

Аналіз наукових джерел і нормативних документів підтверджує необхідність впровадження комплексних заходів захисту у веб-сервіси QRNG. Зокрема, застосування HMAC-SHA-256 у поєднанні з рекомендаціями ISO/IEC, FIPS та NIST дозволяє досягти високого рівня безпеки при обміні повідомленнями.

Формулювання мети дослідження

Метою дослідження є обґрунтування вибору алгоритму HMAC та його модифікації HMAC-SHA-256 для автентифікації повідомлень у відкритому мережевому середовищі; розробка прототипу веб-сервісу QRNG для захищеного обміну повідомленнями з використанням механізмів автентифікації та перевірки цілісності; тестування ключових компонентів системи захисту за різними сценаріями, включаючи захист від підміни, помилкової автентифікації та обробку виняткових ситуацій.

Викладення основного матеріалу дослідження

1. Прототип веб-сервісу QRNG

Схема апаратно-програмного комплексу QRNG наведена на рисунку 1. Основною складовою комплексу є QRNG, інтерфейсний модуль IEEE 488 та сервер, роль якого виконує Raspberry Pi.

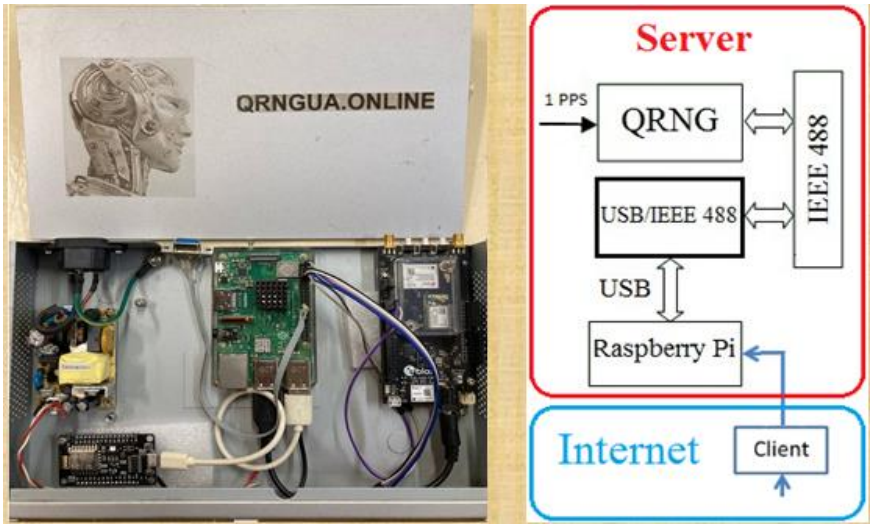

Рис. 1. Схема прототипу апаратно-програмного комплексу QRNG

Користувачі можуть отримати доступ до QRNG як джерела генерування випадкових послідовностей через сайт, де послідовність надається у вигляді рядка цифр розміром послідовності до 1024 чисел (рис. 2). Або через розроблений прототип веб-сервісу QRNG, який за допомогою запитів надає послідовності випадкових чисел у декількох форматах (рис. 3).

Рис. 2. Приклад сторінки сайту з QRNG

Варіант роботи у вигляді веб-сервісу розроблений за стандартом SOAP, а всі результати повертаються у вигляді JSON повідомлень. Доступ до веб-сервісу отримують тільки зареєстровані користувачі, які отримали особистий ключ доступу (api key). Веб-сервіс надає користувачу послугу – генерація випадкової послідовності чисел. Послідовність випадкових чисел надається у вигляді масиву, розмір масиву та формат елементів масиву задається користувачем при запиті. Розмір масиву може бути довільним, а формат елементів має два варіанти:

десяткові числа (uint8) або шістнадцяткові (hex8). Особистий ключ доступу (api key) обов'язково передається при кожному запиті.

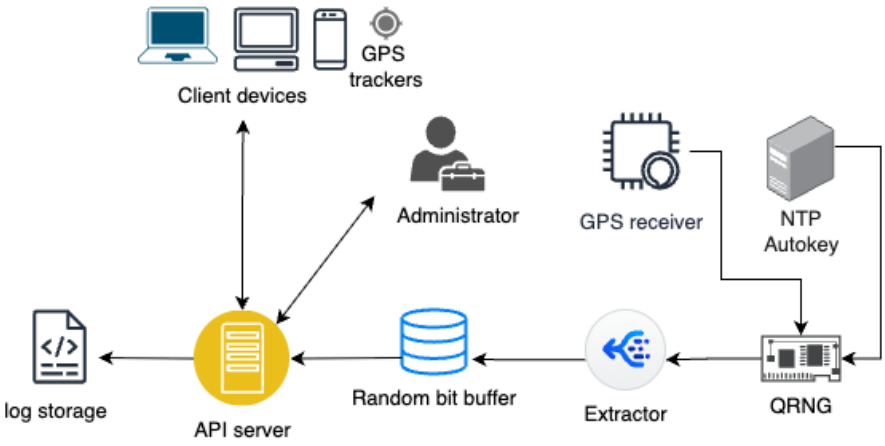

Рис. 3. Схема прототипу сервісу QRNG

2. Комплексний підхід щодо забезпечення безпеки обміну повідомленнями у веб-сервісі QRNG

Побудова моделі загроз та моделі порушника є одним із ключових етапів аналізу інформаційної безпеки, що дозволяє своєчасно виявити критичні вразливості веб-сервісу QRNG, оцінити потенційні ризики та обґрунтувати доцільність впровадження конкретних засобів захисту. Для забезпечення належної безпеки в веб-сервісі QRNG необхідно не тільки запобігати конкретним типам атак, але ще і виявляти слабкі місця в архітектурі та інфраструктурі сервісу. У роботі [11] досліджені типові архітектури таких сервісів, визначені потенційні вектори атак та профіль порушника, а також обґрунтовані та надані рекомендації щодо посилення інформаційної безпеки даного класу систем.

Ефективне забезпечення захищеного обміну повідомленнями у веб-сервісі QRNG потребує комплексного підходу, що включає впровадження надійних криптографічних методів, надійне управління сесіями та ключами, а також застосування сучасних систем автентифікації та захисту від атак типу MITM. Такий підхід дозволяє не лише забезпечити високий рівень безпеки, але й підтримувати високу продуктивність веб-сервісу при високих навантаженнях, що є критично важливим для його стабільної роботи в сучасному цифровому середовищі.

Ключовим елементом захисту є застосування криптографії – основного засобу забезпечення конфіденційності та цілісності даних. Одним із найпоширеніших механізмів захисту є TLS (Transport Layer Security), який забезпечує шифрування всіх переданих повідомлень, знижуючи ризик їх перехоплення або модифікації при передачі через незахищену мережу. Використання протоколу TLS дозволяє реалізувати захищене з'єднання між клієнтом і сервером, захищаючи дані від атак типу «людина посередині» (MITM). Однак для досягнення високого рівня безпеки важливо застосовувати сучасні версії TLS з підтримкою складніших механізмів автентифікації та шифрування, що додатково знижує ризики підслуховування та маніпуляцій з даними.

Також важливим інструментом є HMAC (Hash-based Message Authentication Code) – криптографічний механізм для перевірки автентичності і цілісності повідомлень. HMAC використовує секретний ключ у поєднанні з геш-функцією для створення спеціального коду автентифікації повідомлень. Це дозволяє точно перевірити, чи не було змінено передане повідомлення під час його доставки, забезпечуючи високу надійність захисту. Для сучасних веб-сервісів рекомендованим є використання HMAC-SHA-256 – модифікації HMAC, що використовує одну з найстійкіших геш-функцій сімейства SHA-2. Цей алгоритм забезпечує високу стійкість до атак на колізії та є ідеальним рішенням для збереження цілісності даних в умовах сучасних кіберзагроз.

Однак навіть використання найсучасніших криптографічних методів не може гарантувати повної безпеки, якщо управління сесіями та криптографічними ключами некоректно налаштовано. Адже компрометація секретних ключів або сесійних токенів може призвести до значних витоків даних і втрати контролю над системою. У зв'язку з цим важливим є використання апаратних модулів захисту для надійного зберігання і захисту криптографічних ключів, що забезпечує максимальний рівень безпеки. Крім того, важливим є впровадження процедур для регулярного оновлення ключів, що дозволяє знизити ризики від тривалого використання одного і того ж ключа.

Продуктивність при високих навантаженнях є важливим аспектом, оскільки криптографічні операції, зокрема шифрування та створення кодів автентифікації, можуть суттєва впливати на швидкість роботи веб-сервісу. Для підтримки високої продуктивності в умовах великих обсягів даних і значної кількості одночасних сесій важливо

оптимізувати криптографічні операції, застосовуючи паралельні обчислення або апаратне прискорення криптографії, що дозволяє мінімізувати затримки при забезпеченні високого рівня захисту.

Для запобігання несанкціонованому доступу до ресурсів веб-сервісу необхідно також застосовувати механізми автентифікації користувачів. Найпоширенішим методом є парольна автентифікація, однак цей метод має свої обмеження та вразливості, зокрема, він є схильним до атак типу brute-force, фішинг, а також витоків облікових даних. Тому багато веб-сервісів використовують багатофакторну автентифікацію (Multifactor Authentication, MFA), яка поєднує паролі з додатковими методами перевірки, такими як одноразові паролі, що генеруються за допомогою мобільних застосунків, або біометричні дані.

Іншим важливим аспектом є управління сесіями. Для запобігання несанкціонованому доступу до облікових записів веб-сервіси використовують сесійні токени, що зберігаються у захищених місцях, таких як cookie, доступні лише через захищене з'єднання. Крім того, важливо впроваджувати механізми автоматичного завершення сесії після певного часу неактивності, що допомагає знизити ризик несанкціонованого доступу до облікових записів.

Для захисту від атак MITM та забезпечення автентичності переданих даних використовуються SSL/TLS сертифікати. Вони підтверджують автентичність серверів і забезпечують шифрування переданої інформації, запобігаючи її перехопленню або модифікації. Використання сертифікатів дозволяє впевнитися в тому, що сервер, з яким взаємодіє клієнт, є дійсно тим, за кого себе видає, і що з'єднання між клієнтом і сервером є захищеним від зовнішніх атак.

Комплексний підхід щодо побудови захищеного обміну повідомленнями у веб-сервісі є необхідним для створення надійного і безпечного веб-сервісу QRNG, який буде здатним протистояти сучасним кіберзагрозам і забезпечувати безпеку даних.

3. Коди автентифікації повідомлень

У захищених веб-сервісах, для забезпечення автентичності повідомлень використовуються коди автентифікації повідомлень (Message Authentication Code, MAC), які дозволяють перевірити, що повідомлення дійсно надійшло від заявленого джерела та не було змінено в процесі передавання. Одним із ключових елементів побудови MAC-кодів є криптографічно стійка геш-функція. Саме криптографічно стійка геш-функція є необхідною умовою для побудови MAC-кодів, оскільки вона забезпечує стійкість до базових криптоатак, зокрема підбору прообразу та пошуку колізій.

На основі геш-функцій будуються ключові геш-коди, які на вхід приймають повідомлення та секретний ключ. Ці механізми дають змогу гарантувати цілісність і автентичність даних без залучення додаткових протоколів, базуючись на принципах симетричної криптографії [12].

MAC-функція h, повинна задовольняти таким умовам [12]:

– результат h(K;X) має фіксовану довжину n біт, X може бути довільної довжини, K – це секретний ключ;

– при наявності даних h і X (але з невідомим K), повинно бути складно визначити h(K;X) з імовірністю успіху значно більшою $1/2^n$.

Окремо можна виділити три підходи побудови MAC-кодів:

– MAC-коди, побудовані на основі БСШ (CBC-MAC);

– MAC-коди, побудовані на основі безключових геш-функцій (HMAC);

– MAC-коди, побудовані на основі сімейства універсальних гешфункцій (UMAC).

4. Коди автентифікації повідомлень на основі геш-функцій

Серед багатьох методів гешування та способів реалізації кодів автентифікації повідомлень, особливе місце займає HMAC (Hash-based Message Authentication Code, HMAC) – вкладена конструкція, яка обчислює MAC для основної функції геша h, повідомлення X і секретний ключ K. Альтернативою для HMAC є конструкції MDx-MAC, які можуть бути засновані на MD5, SHA, RIPEMD або аналогічних геш-функціях [13].

Код автентифікації повідомлень HMAC широко використовується у сучасних веб-сервісах і протоколах обміну даними. Його застосування забезпечує високий рівень захисту від підробки повідомлень, втручання та інших загроз. HMAC визначено як обов'язковий (mandatory to implement) механізм у межах IPsec, а також він застосовується в ключових протоколах TLS (Transport Layer Security, TLS), який замінив SSL, і SET (Secure Electronic Transaction, SET). На відміну від SSL, де для створення MAC використовувався застарілий MD5, TLS реалізує HMAC із сучасними геш-функціями, такими як SHA-256, що значно підвищує рівень криптографічного захисту.

HMAC став одним із ключових інструментів побудови безпечних веб-сервісів. Його використання забезпечує баланс між високим рівнем захисту, продуктивністю та простотою реалізації, що робить його оптимальним вибором для побудови захищених систем обміну повідомленнями.

Конструкція HMAC має такий загальний вигляд [9]:

$$MAC(text) = HMAC(K, text) = H\left(\left(K_0 \oplus opad\right) \| H\left(\left(K_0 \oplus ipad\right) \| text\right)\right),$$

де B – розмір блоку (у байтах) вхідних даних для затвердженої геш-функції, L – розмір блоку (у байтах) вихідних даних затвердженої геш-функції, H – затверджена геш-функція, K – секретний ключ, спільний між

відправником та цільовим одержувачем, *text* – дані, для яких обчислюється *HMAC*, K_0 – ключ K після попередньої обробки для формування B-байтового ключа, *opad* – зовнішній блок байту 0x5c повторений B разів, *ipad* – внутрішній блок байту 0x36 повторений B разів, || – операція конкатенації, $\oplus$ – операція виключне АБО.

5. Порівняльний аналіз швидкодії HMAC алгоритмів

Швидкодія алгоритмів HMAC значною мірою залежить від обраної геш-функції, її реалізації та обчислювального середовища.

У таблиці 1 наведені значення швидкодії різних реалізацій HMAC для декількох типів процесорів. Швидкість обчислень визначається кількістю циклів процесора, затрачених на обробку одного байту повідомлення [13].

Геш-функції сімейства SHA-1 та SHA-2 (SHA-256, SHA-384, SHA-512) демонструють гарний баланс між криптографічною стійкістю та швидкодією [14, 15]. Хоча HMAC-MD5 і HMAC-MD4 демонструють вищу швидкодію, вони не рекомендуються до використання через вразливості. Зокрема, MD5 був зламаний на практиці, що дозволило підробляти цифрові підписи та сертифікати.

Таблиця 1

Порівняння швидкодії HMAC алгоритмів, Мбіт/с

Алгоритм	Довжина MAC-коду (біт)	Довжина ключа (біт)	Тип ПЕОМ				
			Pentium2	PIII/Linux	Pentium4	Xeon	AMD
HMAC-Whirlpool	512	512	86	72	98	103	100
HMAC-MD4	128	512	4.7	4.7	6.4	6.4	4.7
HMAC-MD5	128	512	7.2	7.3	9.4	9.4	7.4
HMAC-RIPEMD	160	512	23	18	27	26	21
HMAC-SHA-0	160	512	16	15	23	23	13
HMAC-SHA-1	160	512	16	15	25	24	12
HMAC-SHA-2	256 384 512	512	40 84 84	39 84 84	40 124 124	39 132 132	33 72 72
HMAC-Tiger	192	512	24	21	28	26	20

Таким чином, HMAC є універсальним та надійним механізмом автентифікації повідомлень. Серед усіх розглянутих варіантів HMAC-SHA-256 є найбільш рекомендованим для впровадження у веб-сервіси QRNG та інших відкритих середовищах передачі даних, де актуальні загрози MITM-атак, підробки або втрати цілісності інформації.

6. Основні компоненти захищеного веб-сервісу QRNG

На етапі практичної реалізації веб-сервісу QRNG для захищеного обміну повідомленнями основна увага приділялася забезпеченню автентифікації користувачів, перевірці цілісності та достовірності повідомлень, а також захисту даних від несанкціонованого доступу. Впровадження механізму HMAC на основі SHA-256 дозволяє гарантувати як цілісність переданих даних, так і їхню автентичність.

Архітектура веб-сервісу QRNG (рис. 3) побудована на взаємодії двох основних компонентів – клієнта та сервера – через захищене WebSocket-з'єднання. Кожен компонент має чітко визначені обов'язки:

– клієнтська частина відповідає за введення та валідацію даних, формування структур запитів, створення HMAC-кодів для перевірки цілісності повідомлень, підтримку сесії за допомогою cookie, візуалізацію повідомлень у чаті та обробку відповідей від сервера;

– серверна частина здійснює обробку вхідних підключень, автентифікацію користувачів, перевірку формату та достовірності повідомлень, передавання повідомлень між клієнтами, обробку винятків, а також контроль активності та статусу підключених користувачів.

Ключові механізми захисту та логіки функціонування системи включають:

– встановлення захищеного TLS-з'єднання – весь трафік між клієнтом і сервером передається через HTTPS та WSS (для обміну повідомленнями), що гарантує конфіденційність і захист від перехоплення;

– автентифікацію користувача, яка здійснюється шляхом порівняння HMAC, згенерованого на основі введеного пароля, із записом у базі даних. Варто зазначити, що база містить лише HMAC-геші паролів, що забезпечує додатковий рівень безпеки: навіть у разі компрометації бази даних зловмисник не зможе відновити оригінальний пароль;

– перевірку достовірності приватних повідомлень – кожне повідомлення супроводжується HMAC-кодом, який перевіряється на стороні отримувача;

– валідацію JSON-структур – всі повідомлення аналізуються на предмет коректності формату та наявності обов'язкових полів;

– обробку винятків – система реагує на типові помилки, включаючи спроби підробки MAC, відсутність критичних полів, некоректну структуру запитів, або спроби несанкціонованого доступу і генерує відповідні відповіді;

– синхронізацію онлайн-статусу користувачів – через службові повідомлення status: online/offline підтримується актуальний список підключених клієнтів;

– ідентифікацію користувачів у межах сесії – за допомогою cookie клієнт автоматично ідентифікується при кожному запиті, забезпечуючи стійку авторизацію протягом сеансу.

Таким чином, веб-сервіс QRNG забезпечує надійний та безпечний обмін повідомленнями за рахунок валідації користувачів, перевірки автентичності кожного повідомлення та контролю за цілісністю даних.

7. Блок-схеми логіки системи

В роботі для чіткого представлення алгоритмів взаємодії розроблені блок-схеми, що демонструють логіку роботи основних механізмів системи. Також надано опис процедури автентифікації та обміну приватними повідомленнями [16].

7.1 Автентифікація користувача.

Процес починається із введення користувачем логіна та пароля, після чого ці дані передаються на сервер. Сервер перевіряє, чи існує користувач у базі даних. У разі позитивної відповіді, генерується код HMAC-SHA-256 для введеного пароля, який порівнюється з тим, що зберігається у базі. Якщо значення збігаються, користувач отримує повідомлення про успішну автентифікацію та доступ до чату.

У разі помилки на будь-якому з етапів (некоректні вхідні дані, невірний формат JSON, відсутність користувача або помилкове HMAC-значення), клієнт отримує повідомлення про помилку з відповідним поясненням. Це дозволяє локалізувати причину відмови та забезпечити захищене середовище ще до входу користувача в систему.

Логіка автентифікації користувача наведена на рисунку 4.

Блок-схема була розроблена згідно з [17], для візуалізації використовувався онлайн конструктор блок-схем [18].

7.2 Обмін приватними повідомленнями.

Кожна сесія користувача базується на збереженні унікального ID, який передається у cookie. При вході користувача цей ID асоціюється з WebSocket-з'єднанням. Сервер підтримує список активних користувачів, що оновлюється при вході/виході через спеціальні службові повідомлення типу status: online або status: offline. Ці повідомлення надсилаються автоматично при зміні стану користувача.

Логіка автентифікації приватних повідомлень наведена на рисунку 5.

Передача повідомлень відбувається за принципом «клієнт – сервер – клієнт». Спочатку перевіряється, чи коректно введені дані (текст повідомлення, ID отримувача). Потім повідомлення разом із згенерованим HMAC надсилається на сервер. Сервер QRNG перевіряє онлайн-статус отримувача, і при позитивному результаті передає повідомлення одержувачу.

Отримувач виконує обчислення HMAC для перевірки автентичності. У разі збігу повідомлення виводиться у чаті. У протилежному випадку – ігнорується як потенційно змінене або підроблене. У разі недоступності одного з учасників передача повідомлення не здійснюється, і користувач інформується про невдалу доставку.

У поданих блок-схемах чітко розмежовані функції клієнта і сервера. Клієнт відповідає за формування вхідних даних та ініціацію запитів, сервер – за перевірку, обробку та генерацію відповідей. HMAC виступає як центральний елемент перевірки цілісності та достовірності, реалізуючи захист від підміни або зміни повідомлення.

Валідація структури JSON, перевірка наявності користувачів, синхронізація стану клієнтів (онлайн/офлайн) – усе це є невід'ємними частинами загальної архітектури безпечного обміну. Таким чином, система виявляє будь-які спроби підміни або фальсифікації повідомлень ще до їх відображення, що забезпечує довіру між користувачами.

8. Тестування веб-сервісу QRNG для захищеного обміну повідомленнями

Після реалізації функціональних компонентів системи було проведене тестування з метою перевірки стабільності роботи веб-сервісу QRNG, відповідності заявленим вимогам безпеки, правильності логіки автентифікації, обробки повідомлень та забезпечення захищеного з'єднання. Частково результати тестування наведені у цій статті на рисунках 6-13.

8.1 Тестування автентифікації користувачів.

Етап тестування автентифікації користувачів передбачав перевірку реакції системи на різні сценарії взаємодії з формою входу, коректність валідації введених даних та збереження сесії після входу до системи.

Попередньо було створено три тестові облікові записи (user#10, user#11, user#12).

1. Порожні поля.

У разі спроби автентифікації без введення логіну та/або пароля система генерує повідомлення про необхідність заповнення відповідних полів. При цьому запит до сервера не виконується, що свідчить про коректну реалізацію клієнтської валідації.

2. Неправильні облікові дані.

При введенні неправильного логіну або пароля система реагує повідомленням про помилку автентифікації. Сервер перевіряє існування користувача, обчислює HMAC від пароля, і при його невідповідності очікуваному значенню повертає повідомлення про відмову у доступі.

Рис. 4. Блок-схема автентифікації користувача

3. Успішна автентифікація.

При правильних облікових даних сервер успішно виконує перевірку HMAC, після чого користувача перенаправляє до чату. У браузері автоматично зберігається cookie з user_id, який ідентифікує користувача в поточній сесії.

8.2 Тестування захищеного з’єднання.

Підтвердження використання протоколів HTTPS та WSS при з’єднанні між клієнтом і сервером здійснювалося за допомогою кількох інструментів:

1. Перевірка через браузер.
2. Перевірка WebSocket-з’єднання.
3. Перевірка з використанням OpenSSL.
4. Перевірка захищеного з’єднання.

За допомогою утиліти Wireshark був проведений аналіз мережевого трафіку, щоб впевнитися у тому, що дані не передаються у відкритому вигляді.

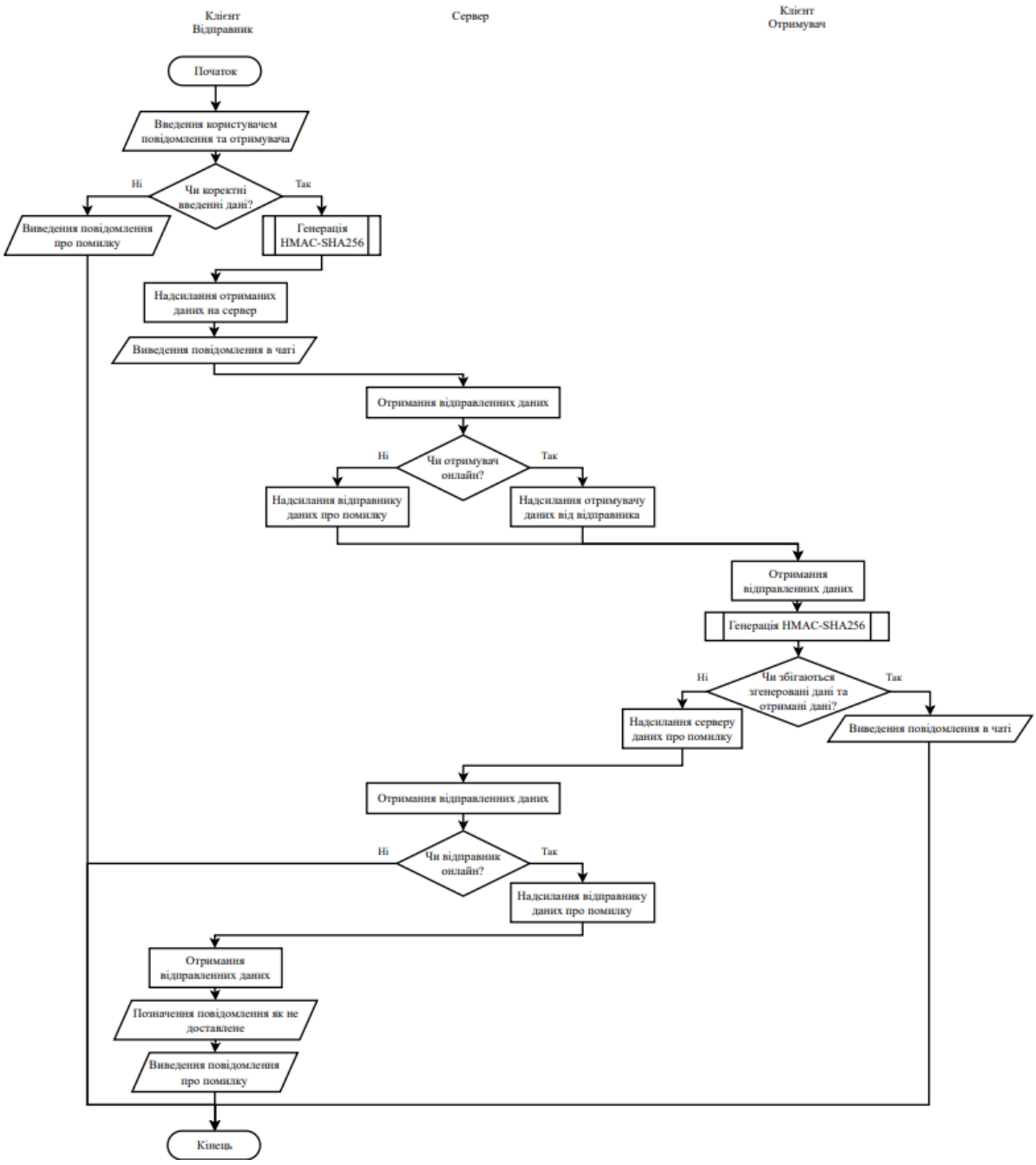

Рис. 5. Блок-схема обміну приватними повідомленнями

Після запуску сервісу та відкриття веб-сторінки з логіном/чатом у браузері, у Wireshark фіксується трафік на порту 9001. Всі пакети позначено як TLS, а вміст поля як Encrypted Application Data, що свідчить про те, що дані передаються у зашифрованому вигляді. Результат тестування наведено на рисунку 6.

8.3 Тестування онлайн-статусу та сесій.

Метою тестування є перевірка коректності підтримки статусу користувача в режимі реального часу, а також реакції системи на зміну стану.

1. Вхід нового користувача.

Після успішної автентифікації користувача, інші учасники чату автоматично бачать його в списку онлайн.

2. Вихід користувача.

У разі виходу з акаунту або закриття вкладки браузера, WebSocket-з’єднання завершується, і сервер надсилає всім підключеним клієнтам повідомлення status: offline. У результаті, користувач зникає зі списку активних.

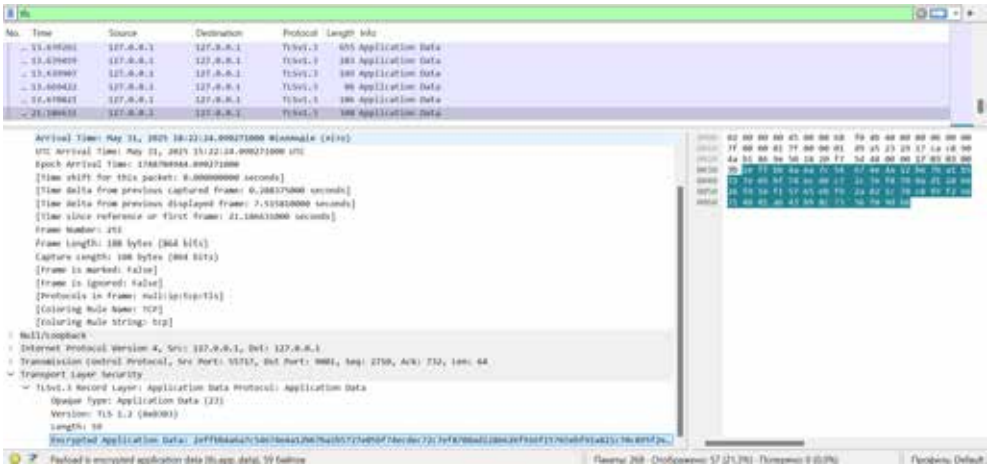

Рис. 6. Перехоплений трафік у Wireshark

8.4 Тестування обміну повідомленнями.

Метою цього етапу є перевірка коректності обробки публічних і приватних повідомлень, включаючи нетипові ситуації та перевірку захисту цілісності через HMAC.

1. Порожнє повідомлення.

Якщо користувач намагається надіслати порожнє повідомлення, виводиться повідомлення про помилку.

2. Успішне публічне повідомлення.

При надсиланні публічного повідомлення, воно з’являється у всіх підключених користувачів, вказується ім’я відправника.

Результати тестування наведено на рисунках 7-8.

Рис. 7. Публічне повідомлення (інтерфейс відправника)

Рис. 8. Публічне повідомлення (інтерфейс одного з отримувачів)

3. Приватне повідомлення.

Спроба надіслати повідомлення самому собі – система перевіряє, чи збігається ID відправника з ID отримувача. У разі збігу виводиться повідомлення про помилку, відправка не відбувається (рис. 9).

Рис. 9. Виведення помилки при спробі відправити повідомлення самому собі

Порожній ID отримувача – повідомлення не обробляється, виводиться клієнтська помилка (рис. 10).

Рис. 10. Виведення помилки при спробі відправити повідомлення при відсутності ID отримувача

4. Приватне повідомлення офлайн користувачу.

Сервер перевіряє, чи є отримувач серед активних підключень. Якщо користувач офлайн – повідомлення не доставляється, відправник інформується про недоступність доставки.

5. Імітація підробки HMAC у повідомленні.

Для імітації підробки HMAC у вхідному повідомленні була штучно модифікована логіка його обчислення. Через розбіжності між обчисленим та переданим HMAC-кодом, отримувач відкидає отримане приватне повідомлення, а відправник інформується про невідповідність. Результати тестування наведені на рисунках 11-13.

```
case "incoming_private_msg":
    const hmacKey = "key_" + payload.user_from;
    //const expectedHmac = hmacSha256(payload.text, hmacKey);
    const expectedHmac = "wrongHMAC";
    if (expectedHmac === payload.hmac) {
        createPrivateChat(payload.user_from, `User #${payload.user_from}`);
        displayMessage(payload.user_name, payload.text, "private", payload.user_from);
    }
```

Рис. 11. Імітація підробки HMAC в коді

Рис. 12. Повідомлення про неможливість доставки через помилковий HMAC-код (інтерфейс відправника)

Рис. 13. Повідомлення не відображено через помилковий HMAC-код (інтерфейс отримувача)

6. Успішне приватне повідомлення.

У разі, якщо обидва користувачі онлайн і всі параметри повідомлення коректні, система успішно передає повідомлення від одного користувача до іншого. Отримувач виконує перевірку HMAC, яка проходить успішно, після чого повідомлення з’являється у вікні чату.

В результаті проведеного тестування підтверджено коректність реалізації ключових функціональних складових веб-сервісу QRNG, зокрема механізмів автентифікації, шифрування трафіку, обробки повідомлень і підтримки статусу користувачів. Усі перевірені сценарії – як коректні, так і з внесеними навмисно помилками – дали очікувані результати, що свідчить про стабільність роботи системи та відповідність реалізованого сервісу вимогам інформаційної безпеки. Це включає дотримання базових принципів захисту інформації: конфіденційності, цілісності та автентичності даних. Отримані результати також підтверджують практичну придатність веб-сервісу QRNG та його здатність ефективно протидіяти типовим загрозам у мережевому середовищі.

Висновки

Таким чином, для забезпечення захищеного обміну повідомленнями у веб-сервісі QRNG доцільно використовувати комбінацію криптографічних алгоритмів, захищених протоколів передавання даних, кодів автентифікації та організаційних заходів, спрямованих на контроль доступу і своєчасне оновлення компонентів системи. Такий комплексний підхід дозволяє зменшити ризики перехоплення, модифікації або компрометації повідомлень, що циркулюють у системі.

Практичне значення роботи полягає в можливості використання отриманих результатів для підвищення рівня захисту веб-сервісу QRNG, зокрема через інтеграцію HMAC-SHA-256 у механізми перевірки автентичності повідомлень.

Під час дослідження також були виявлені недоліки окремих підходів до автентифікації, пов'язані з вразливістю до атак на паролі, складністю управління ключами та залежністю від апаратної платформи. Подальші дослідження доцільно спрямувати на вивчення методів динамічного управління ключами, оцінювання продуктивності HMAC у реальних веб-сервісах QRNG з великою кількістю клієнтів.

Список використаної літератури

1. Пірог О.В. Безпека вебдодатків : навч. посібн. Житомир : Житомирська політехніка, 2025. – 290 с.
2. Моргуль Д.М., Нарєжній О.П., Гріненко Т.О. Класифікація атак та вимоги кібербезпеки до веб-ресурсу QRNG. ISSN 0485-8972. *Radiotekhnika No. 220*. С. 50-59. DOI:10.30837/rt.2025.1.220.04.
3. Васильченко Д.І., Лавровський І.М. Огляд типових уразливостей web-сайтів організацій у 2019-2020 році. *Сучасний захист інформації.* 2021. № 1(45). С. 41-46.
4. Кравчук Н.В., Коробейнікова Т.І. Огляд проблематики захищеного доступу до вебсерверів. *Вісник Львівського державного університету безпеки життєдіяльності.* 2024. № 30. С. 78-89.
5. Коробейнікова Т.І., Матвійчук А.А., Непийвода М.В. Огляд рішень для покращення процесів автентифікації та авторизації на вебресурсах. *SWorldJournal.* 2024. № 25-01, С. 82-88.
6. RFC 6455:2011. The WebSocket Protocol / I. Fette, A. Melnikov. – [Чинний від 2011-12]. Вид. офіц. Internet Engineering Task Force (IETF), 2011. 71 с. (Request for Comments; 6455).
7. ISO/IEC 27002:2022. Information security, cybersecurity and privacy protection – Information security controls. 3rd ed. [Чинний від 2022-02]. Вид. офіц. Geneva: ISO/IEC, 2022. 152 с.
8. НД ТЗІ 2.5-010-03. Вимоги до захисту інформації WEB-сторінки від несанкціонованого доступу. [Чинний від 2003-04-15]. Вид. офіц. Київ: ДСТСЗІ СБ України, 2003. 25 с.
9. FIPS PUB 198-1. The Keyed-Hash Message Authentication Code (HMAC). [Чинний від 2008-07]. Вид. офіц. Gaithersburg, MD: NIST, 2008. 7 с.
10. NIST SP 800-224 ipd. Keyed-Hash Message Authentication Code (HMAC): Specification of HMAC and Recommendations for Message Authentication / M. S. Turan, L. T. A. N. Brandão. [Чинний від 2024-06]. Вид. офіц. Gaithersburg, MD: NIST, 2024. 24 с. Initial Public Draft.
11. Моргуль Д.М., Нарєжній О.П., Гріненко Т.О. Модель порушника та модель загроз для веб-сервісу QRNG. ISSN 0485-8972. *Radiotekhnika No. 221.* С. 31-38. DOI:10.30837/rt.2025.2.221.04.
12. Євсеєв С.П., Йохов О.Ю, Король О.Г. Гешування даних в інформаційних системах: монографія. Харків: ХНЕУ, 2013. 312 с.
13. Остапов С.Е., Євсеєв С.П., Король О.Г. Технології захисту інформації : навчальний посібник. Харків: ХНЕУ, 2013. 476 с.
14. Малярова Д. М. Порівняння швидкодії HMAC алгоритмів. Проблеми та перспективи забезпечення цивільного захисту : матеріали міжнародної науково-практичної конференції молодих учених. Харків, 2024. С. 256.
15. Малярова Д. М. Порівняльний аналіз алгоритмів гешування SHA-1 та сімейства SHA-2. *Глобалізація наукових знань : міжнародна співпраця та інтеграція галузей науки* : матеріали VI міжнародної студентської наукової конференції. Вінниця, 2024. С. 270-271.
16. Малярова Д.М., Гріненко Т.О., Нарєжній О.П. Використання HMAC-SHA-256 як методу захисту у вебсервісах. *Молоді вчені 2025 – від теорії до практики* : матеріали XV всеукраїнської науково-практичної конференції. Дніпро : Журфонд, 2025. С. 313-316.
17. ДСТУ ISO 5807:2016. Символи та угоди щодо документації стосовно даних, програм та системних блок-схем, схем мережевих програм та схем системних ресурсів (ISO5807:1985, IDT). [Чинний від 10.10.2016]. К.: ДП "УкрНДНЦ", 2016. 30 с. (Обробляння інформації).

18. Блок-схема онлайн – виробник блок-схеми. Free Online Apps for Most Popular File Formats. URL: https://products.aspose.app/diagram/uk/flowchart (дата звернення: 28.06.2025).

References

1. Piroh, O. V. (2025). Bezpeka vebdodativ: navchalnyi posibnyk [Web application security]. Zhytomyr: Zhytomyrska politekhnika. [in Ukrainian].

2. Morhul, D. M., Nariezhnii, O. P., & Hrinenko, T. O. (2025). Klasyfikatsiia atak ta vymohy kiberbezpeky do veb-resursu QRNG [Classification of attacks and cybersecurity requirements for the QRNG web resource]. *Radiotekhnika*, (220), 50-59. https://doi.org/10.30837/rt.2025.1.220.04. [in Ukrainian].

3. Vasylchenko, D. I., & Lavrovskyi, I. M. (2021). Ohliad typovykh urazlyvostei web-saitiv orhanizatsii u 2019–2020 rotsi [Overview of typical vulnerabilities of organization web-sites in 2019–2020]. *Suchasnyi zakhyst informatsii*, 1(45), 41-46. [in Ukrainian].

4. Kravchuk, N. V., & Korobeinykova, T. I. (2024). Ohliad problematyky zakhyshchenoho dostupu do vebserveriv [Overview of secure access issues to web servers]. *Visnyk Lvivskoho derzhavnoho universytetu bezpeky zhyttiediialnosti*, (30), 78–89. [in Ukrainian].

5. Korobeinykova, T. I., Matviichuk, A. A., & Nepyivoda, M. V. (2024). Ohliad rishen dlia pokrashchennia protsesiv avtentyfikatsii ta avtoryzatsii na veb-resursakh [Overview of solutions for improving authentication and authorization processes on web resources]. *SWorldJournal*, (25-01), 82-88. [in Ukrainian].

6. Fette, I., & Melnikov, A. (2011). *RFC 6455: The WebSocket Protocol*. Internet Engineering Task Force (IETF). Retrieved from https://www.rfc-editor.org/rfc/rfc6455.

7. International Organization for Standardization & International Electrotechnical Commission. (2022). *ISO/IEC 27002:2022 – Information security, cybersecurity and privacy protection – Information security controls* (3rd ed.). Geneva: ISO/IEC.

8. Derzhspetszviazok of Ukraine. (2003). *ND TZI 2.5-010-03: Vymohy do zakhystu informatsii WEB-storinky vid nesanktsionovanoho dostupu* [Requirements for protecting web pages from unauthorized access]. Kyiv: DSTSZI SB Ukrainy. [in Ukrainian].

9. National Institute of Standards and Technology. (2008). *FIPS PUB 198-1: The Keyed-Hash Message Authentication Code (HMAC)*. Gaithersburg, MD: NIST.

10. National Institute of Standards and Technology. (2024). *NIST SP 800-224 (Initial Public Draft): Keyed-Hash Message Authentication Code (HMAC): Specification and Recommendations for Message Authentication*. Gaithersburg, MD: NIST.

11. Morhul, D. M., Nariezhnii, O. P., & Hrinenko, T. O. (2025). Model porushnyka ta model zahroz dlia veb-servisu QRNG [Attacker model and threat model for QRNG web service]. *Radiotekhnika*, (221), 31–38. https://doi.org/10.30837/rt.2025.2.221.04. [in Ukrainian].

12. Yevseiev, S. P., Yokhov, O. Yu., & Korol, O. H. (2013). *Heshuvannia danykh v informatsiinykh systemakh: monohrafiia* [Data hashing in information systems: Monograph]. Kharkiv: KhNEU. [in Ukrainian].

13. Ostapov, S. E., Yevseiev, S. P., & Korol, O. H. (2013). *Tekhnolohii zakhystu informatsii: navchalnyi posibnyk* [Information protection technologies: Textbook]. Kharkiv: KhNEU. [in Ukrainian].

14. Maliarova, D. M. (2024). Porivniannia shvydkodii HMAC alhorytmiv [Comparison of HMAC algorithm performance]. In *Problemy ta perspektyvy zabezpechennia tsyvilnoho zakhystu* (p. 256). Kharkiv. [in Ukrainian].

15. Maliarova, D. M. (2024). Porivnialnyi analiz alhorytmiv heshuvannia SHA-1 ta simeistva SHA-2 [Comparative analysis of SHA-1 and SHA-2 hashing algorithms]. In *Hlobalizatsiia naukovykh znan: Mizhnarodna spivpratsia ta intehratsiia haluzei nauky* (pp. 270–271). Vinnytsia. [in Ukrainian].

16. Maliarova, D. M., Hrinenko, T. O., & Nariezhnii, O. P. (2025). Vykorystannia HMAC-SHA-256 yak metodu zakhystu u veb-servisakh [Use of HMAC-SHA-256 as a protection method in web services]. In *Molodi vcheni 2025 – vid teorii do praktyky* (pp. 313–316). Dnipro: Zhurfond. [in Ukrainian].

17. Ukrainian Scientific Research and Training Center for Standardization. (2016). *DSTU ISO 5807:2016 – Symbols and conventions for documentation relating to data, program and system flowcharts (ISO 5807:1985, IDT)*. Kyiv: DP "UkrNDNTs".

18. Aspose. (2025). *Flowchart online generator*. Free Online Apps for Most Popular File Formats. Retrieved from https://products.aspose.app/diagram/uk/flowchart (Accessed: June 28, 2025).

Дата першого надходження рукопису до видання: 22.11.2025
Дата прийнятого до друку рукопису після рецензування: 18.12.2025
Дата публікації: 31.12.2025