Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)
*Факультет комп'ютерно-інформаційних систем і програмної інженерії*
(назва факультету )
*Кафедра кібербезпеки*
(повна назва кафедри)

# КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр
(освітній рівень)

на тему: "Дослідження моделей і методів контролю цілісності й автентичності даних у телекомунікаційних мережах та їх колізійних властивостей"

Виконав: студент (ка)

Спеціальності:
*125 «Кібербезпека»*
(шифр і назва напряму підготовки, спеціальності)

*Гродський Б.П.*
підпис (прізвище та ініціали)

Керівник *Максимчук О.О.*
*підпис (прізвище та ініціали)*

Нормоконтроль *Лобур Т.Б.*
підпис (прізвище та ініціали)

Завідувач кафедри *Загородна Н.В.*
підпис (прізвище та ініціали)

Рецензент
підпис (прізвище та ініціали)

м.Тернопіль – 2022

Міністерство освіти і науки України
**Тернопільський національний технічний університет імені Івана Пулюя**

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра кібербезпеки
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Загородна Н.В.
(підпис) (прізвище та ініціали)
«__» _____ 2022 р.

# ЗАВДАННЯ
## НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 125 Кібербезпека
(шифр і назва спеціальності)

Студенту Гродському Богдану Павловичу
(прізвище, ім'я, по батькові)

1. Тема роботи " Дослідження моделей і методів контролю цілісності й автентичності даних у телекомунікаційних мережах та їх колізійних властивостей

Керівник роботи Максимчук Олександр Олександрович, асистент викладач каф. КБ
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 23 » 03 2022 року № 4/7-178

2. Термін подання студентом завершеної роботи 17.06.2022

3. Вихідні дані до роботи

4. Зміст роботи (перелік питань, які потрібно розробити)

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

6. Консультанти розділів роботи

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|---|---|---|---|
| | | завдання видав | завдання прийняв |
| Безпека життєдіяльності, основи хорони праці | Пулька Ч.В., проф. кафедри МТ | | |

7. Дата видачі завдання 16.02.2022 р.

КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів роботи | Термін виконання етапів роботи | Примітка |
|---|---|---|---|
| 1. | Ознайомлення з завданням до кваліфікаційної роботи | 16.02 – 19.02 | *Виконано* |
| 2. | Підбір джерел пметодах контролю контролю цілісності й автентичності | 20.02 – 27.02 | *Виконано* |
| 3. | Опрацювання джерел в галузі дослідження | 28.02 – 16.03 | *Виконано* |
| 4. | Вибір програмних засобів розробки | 17.03 – 20.03 | *Виконано* |
| 5. | Реалізація моделі і методу для сформування коду автентичності і контролю цілісності | 20.03-05.04 | *Виконано* |
| 6. | Оформлення розділу «Аналіз протоколів, що здійснюють контроль автентичності та цілісності даних у телекомунікаційних мережах» | 06.03 – 17.04 | *Виконано* |
| 7. | Оформлення розділу «Реалізація моделі і методу для сформування коду автентичності і контролю цілісності, що базується на модульних перетвореннях» | 18.04 – 29.04 | *Виконано* |
| 8. | Оформлення розділу «Дослідження колізійних властивостей кодів контролю цілісності й автентичності даних і обґрунтування пропозицій з їх удосконалення» | 30.04 – 13.05 | *Виконано* |
| 9. | Виконання завдання до підрозділу «Безпека життєдіяльності, основи хорони праці» | 14.05 – 21.05 | *Виконано* |
| 10. | Оформлення кваліфікаційної роботи | 22.05 – 05.06 | *Виконано* |
| 11. | Нормоконтроль | 06.06 – 12.06 | *Виконано* |
| 12. | Перевірка на плагіат | 10.06 – 15.06 | *Виконано* |
| 13. | Попередній захист кваліфікаційної роботи | 16.06 – 19.06 | *Виконано* |
| 14. | Захист кваліфікаційної роботи | 24.06.2022 | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

Студент _____ Гродський Б.П.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Максимчук О.О.
(підпис) (прізвище та ініціали)

# АНОТАЦІЯ

Дослідження моделей і методів контролю цілісності й автентичності даних у телекомунікаційних мережах та їх колізійних властивостей // Кваліфікаційна робота ОР «Бакалавр» // Гродський Богдан Павлович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра кібербезпеки, група СБс-42 // Тернопіль, 2022 // С. , рис. – , табл. – , сл. – , додат. – .

*Ключові слова:* ЗАХИСТ ІНФОРМАЦІЇ, ПРОТОКОЛИ МЕРЕЖНОЇ БЕЗПЕКИ, ГЕШ-ФУНКЦІЯ, UMAC, MASH-1, MASH-2, ГЕШУВАННЯ, ЦІЛІСНІСТЬ, АВТЕНТИЧНІСТЬ, NESSIE.

Мета роботи полягає у в аналізі механізмів забезпечення автентичності і цілісності інформації у телекомунікаційних мережах, оцінці їх основних ймовірно-часових характеристик, програмної реалізації нових методів формування MAC-кодів на основі використання модулярних перетворень.

В результаті виконання роботи отримані наступні результати:

- проведено аналіз стану та обґрунтування шляхів удосконалювання моделей і методів контролю цілісності та автентичності даних, алгоритмів ключового й безключового гешування для формування кодів виявлення маніпуляцій (MDC) і кодів автентифікації повідомлень (MAC);
- реалізовано математичний апарат і здійснено дослідження колізійних властивостей кодів контролювання цілісностей та аутентичності даних, обґрунтовано пропозиції з їх удосконалювання;
- реалізовано модель і метод формування кодів контролю цілісності та автентичності даних на основі доведено стійкого ключового гешування;
- досліджено ефективність запропонованих моделей і методів контролю цілісності та автентичності даних, обґрунтовано практичні рекомендації з їх використання в протоколах безпеки комунікаційних мереж.

# ANNOTATION

Study of Models and Methods of Control of Data Authenticity and Completeness in Telecommunication Networks and their Collision Characteristics // Qualification thesis of educational level "Bachelor" // Hrodskyi Bohdan Pavlovych // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Cybersecurity, СБс-42 group // Ternopil, 2022 // P. , fig. - , table. - , chair. - , added. - .

*Keywords:* INFORMATION SECURITY PROTOCOLS, NETWORK SECURITY, HASH FUNCTIONS, UMAC, MASH-1, MASH-2, HASHING, INTEGRITY, AUTHENTICITY, NESSIE.

Purpose is to analyze the mechanisms to ensure the authenticity and integrity of information in telecommunication networks, assessing their likely-time major characteristics of program implementation of new methods of generating the MAC code through the use of modular transformations.

As a result of the implementation of the following results:

conducted analysis and study ways to improve models and methods for monitoring the integrity and authenticity of data, algorithms and key keyless hash codes to identify the formation of manipulation (MDC) authentication messages and codes (MAC);

implemented mathematical tools and the research of conflicting properties codes control the integrity and authenticity of data substantiated proposals for their improvement;

implemented model and method of forming the control codes of integrity and authenticity of data based on proven resistant hash key;

investigated the effectiveness of the proposed models and methods of monitoring the integrity and authenticity of data, practical recommendations on their use in security protocol telecommunication networks.

ЗМІСТ

# ВСТУП

Сучасні комунікаційні мережі характеризуються швидким зростанням кількості користувачів і споживачів інформації, розширенням спектра комунікаційних послуг, які надаються, насамперед забезпеченням доступу до різних мультимедійних сервісів та технологій, підтримки вилучених користувачів, обслуговуванню суб'єктів автоматизованої інформаційної взаємодії тощо. Ці тенденції зумовлюють різке підвищення обсягів даних, що оброблюються і передаються, і, як наслідок, жорсткість ймовірнісно-часових вимог, які пропонуються до основних компонент комунікаційних мереж. Найважливішим показником ефективності сучасних комунікаційних мереж є їх безпека, під якою розуміється здатність до забезпечення цілісності, автентичності і конфіденційності даних, що обробляються та передаються. Реалізація цих характеристик напряму впливає на рівень захищеності від сучасних загроз мережної безпеки й, зрештою, якість комунікаційних послуг, які надаються. Проведений аналіз основних загроз безпеки, а також особливостей побудови комунікаційних протоколів і протоколів захисту інформації в комунікаційних мережах свідчить про стрімкий розвиток обчислювальних можливостей та IT-технологій, які «модернізують» сучасні види загроз, розширюють і вдосконалюють технології, їх реалізацію, створюють нові сучасні технології зламу систем мережної безпеки. Виникає протиріччя між ймовірнісно-часовими вимогами до перспективних механізмів контролюванні автентичності та інформації в умовах безперервного вдосконалювання загроз інформаційної безпеки і реальним станом існуючих моделей, методів й обчислювальних алгоритмів.

Проведений аналіз показав, що одним з найбільш ефективних підходів до побудови механізмів контролю автентичності і цілісності груп даних в телекомунікаційних мережах є ключове і безключове гешування, яке використовується для того, щоб сформувати спеціалізовані коди, завдяки яким буде виявлено ряд маніпуляцій (MDC – Manipulation Detection Code) для контролю цілісності даних. Практичне застосування відповідних

механізмів дозволяє без залучення додаткових засобів забезпечувати підвищення цілісності та автентичності даних, що оброблюються та передаються в мережах телекомунікацій.

Аналіз сучасних протоколів безпеки телекомунікаційних систем і мереж свідчить про їх нездатність і потенційну уразливість до сучасних інформаційних загроз. Зокрема, у роботі Niels Ferguson, Bruce Schneier “A Cryptographic Evaluation of IPSec” під час оцінки протоколів мережної безпеки IPSec виявлена наявність уразливостей практично в усіх головних компонентах, відзначається необхідність удосконалювання механізмів безпеки, які використовуються для забезпечення високого рівня стійкості. У першу чергу це стосується механізмів, які потрібні для забезпечення контролю цілісності та автентичності інформації, оскільки загрози спотворення, втрати або дублювання інформації, так само як і нав’язування хибної інформації або неправильних режимів функціонування окремих компонентів телекомунікаційної мережі, є найнебезпечнішими й найбільш поширеними (понад 87 %) на сьогодні загрозами.

Перспективні механізми безпеки повинні забезпечувати вирішення покладених на них завдань у надзвичайних умовах різкого зростання обсягів даних, що обробляються та передаються, розширення спектру загроз інформаційної безпеки на всіх етапах життєвого циклу телекомунікаційних систем і мереж. Отже, тема дипломної роботи, яка присвячена дослідженню моделей, методів і алгоритмів, що в сукупності використовуються, щоб сформувати коди контролю автентичності та цілісності пакетів даних для протоколів безпеки телекомунікаційних мереж, є актуальною.

Об’єкт дослідження – процеси, які здійснюють контроль автентичності груп даних, а також їх цілісності в протоколах безпеки телекомунікаційних мереж.

Предметом є дослідження моделі й методи контролю цілісності та автентичності пакетів даних у протоколах безпеки телекомунікаційних мереж.

# 1 АНАЛІЗ ПРОТОКОЛІВ, ЩО ЗДІЙСНЮЮТЬ КОНТРОЛЬ АВТЕНТИЧНОСТІ ТА ЦІЛІСНОСТІ ДАНИХ У ТЕЛЕКОМУНІКАЦІЙНИХ МЕРЕЖАХ

## 1.1. Протоколи цілісності й автентичності даних в телекомунікаційних мережах

Найважливішим показником ефективності сучасних телекомунікаційних систем і мереж є їх безпека (здатність до забезпечення цілісності, автентичності й конфіденційності оброблюваних і переданих даних). Для технологій такого призначення застосовується все дозволена абривіатура – “захищений канал” (secure channel) [1,4]. У цьому контексті під цим терміном ми повинні розуміти, що захист усіх даних відбувається між двома вузлами системиою. Канал для захисту інформації будується завдяки системних засобів, які реалізуються завдяки моделям систем OSI.(рис. 1.1).


Рисунок 1.1 - Схема реалізацій послуг безпеки телекомунікаційної мережі на різних рівнях моделі OSI

Далі розглянемо протоколи мережної безпеки IPSec, а також зробимо дослідження застосовуваних механізмів, які забезпечують різного роду захист переданих даних.

Internet Protocol Security (IPSec) – повний набір стандартів, який гарантує безпеку передачі даних мережами. Його можна доповнювати новими алгоритмами, протоколами мережевої безпеки, а також її функціями.

Застосування протоколів IPSec гарантує:

- конфіденційність – дані, що передаються, будуть захищені від несанкціонований перегляд;
- автентичність – гарантують передачу даних від того відправника, що від правляє, і не від ніякого іншого;
- цілісність – гарантія передачі даних в початковому вигляді, тобто вони ні в якому разі не будуть втрачені, про дубльовані чи спотворені.

Ядро IPSec складається з трьох протоколів:

- AH (Authentification Header) – це протокол автентифікації;
- ESP (Encapsulation Security Payload) – це протокол шифрування;
- IKE (Internet Key Exchange) – це протокол для обміну ключами.

Проведений аналіз показав, що застосовувані механізми контролю цілісності й автентичності пакетів даних у протоколах безпеки IP-мереж ґрунтуються на використанні моделей і методів ключового й безключового гешування інформації [10, **Error! Reference source not found.**, **Error! Reference source not found.**]. Перші використовуються для формування кодів виявлення маніпуляцій (MDC) і призначені для побудови механізмів контролю цілісності, інші – для формування кодів автентифікації повідомлень (MAC), призначених для побудови механізмів контролю автентичності оброблюваних і переданих даних.

## 1.2. Аналіз і порівняльні дослідження відомих методів створення кодів для автентичності і контролю за цілісністю інформації на основі алгоритма UMAC

Manipulation Detection Code ( надалі MDC) і Message Authentification Code ( надалі MAC) – є ефективним механізмом для контролювання

цілісності і автентичності інформації в сучасних телекомунікаційних мережах. Це так зване гешування інформації, використовуване як для формування наборів кодів, щоб виявляти маніпуляції з інформацією та для контролю цілісності даних, так і кодів. Практичне застосування відповідних механізмів дозволяє без залучення додаткових засобів забезпечувати необхідні показники цілісності й автентичності оброблюваних і переданих даних у мережах телекомунікацій.

Доведено, що найбільш ефективним механізмом контролю цілісності й автентичності інформації в телекомунікаційних мережах є багатошарові конструкції на основі універсального гешування. Розглянуто модель і структурну схему каскадного ключового гешування для формування кодів автентифікації UMAC. Загальна класифікація функцій гешування наведена на рис. 1.2.

Швидкість обчислень, які наведені в табл. 1.1, визначалися кількістю циклів процесору $S$. Як видно з табл. 1.1 наприклад, алгоритм TWO-TRACK-MAC поступається лише по швидкодії тільки алгоритму UMAC приблизно у 3 – 7 разів, але має у 2,5 рази більший розмір MAC-коду.

Аналізуючи табл. 1.1, бачимо, що схема UMAC, завдяки поліноміальним функціям, дозволяє отримати швидкісне хешування. У цьому зв’язку схему UMAC визнано одним з переможців міжнародного криптографічного конкурсу NESSIE.

Коди автентичності повідомлень, сформованих за схемою UMAC, слід використовувати у всіх системах, де швидкість обробки даних є одним із пріоритетних показників ефективності. До таких систем відносяться, насамперед, протоколи мережної безпеки, призначені для забезпечення цілісності й автентичності переданих пакетів даних, у тому числі й в IP-мережах. Фактично, за результатами європейського конкурсу NESSIE, схему UMAC визнано одним з найбільш ефективних кандидатів на заміну застосовуваних алгоритмів формування MAC, наприклад, на заміну

алгоритмів HMAC-MD5-96, HMAC-SHA-1-96, DES-MAC, HMAC- ГОСТ Р 34.11-94, HMAC- ГОСТ Р 34.11-2001 у протоколі AH IPSec.

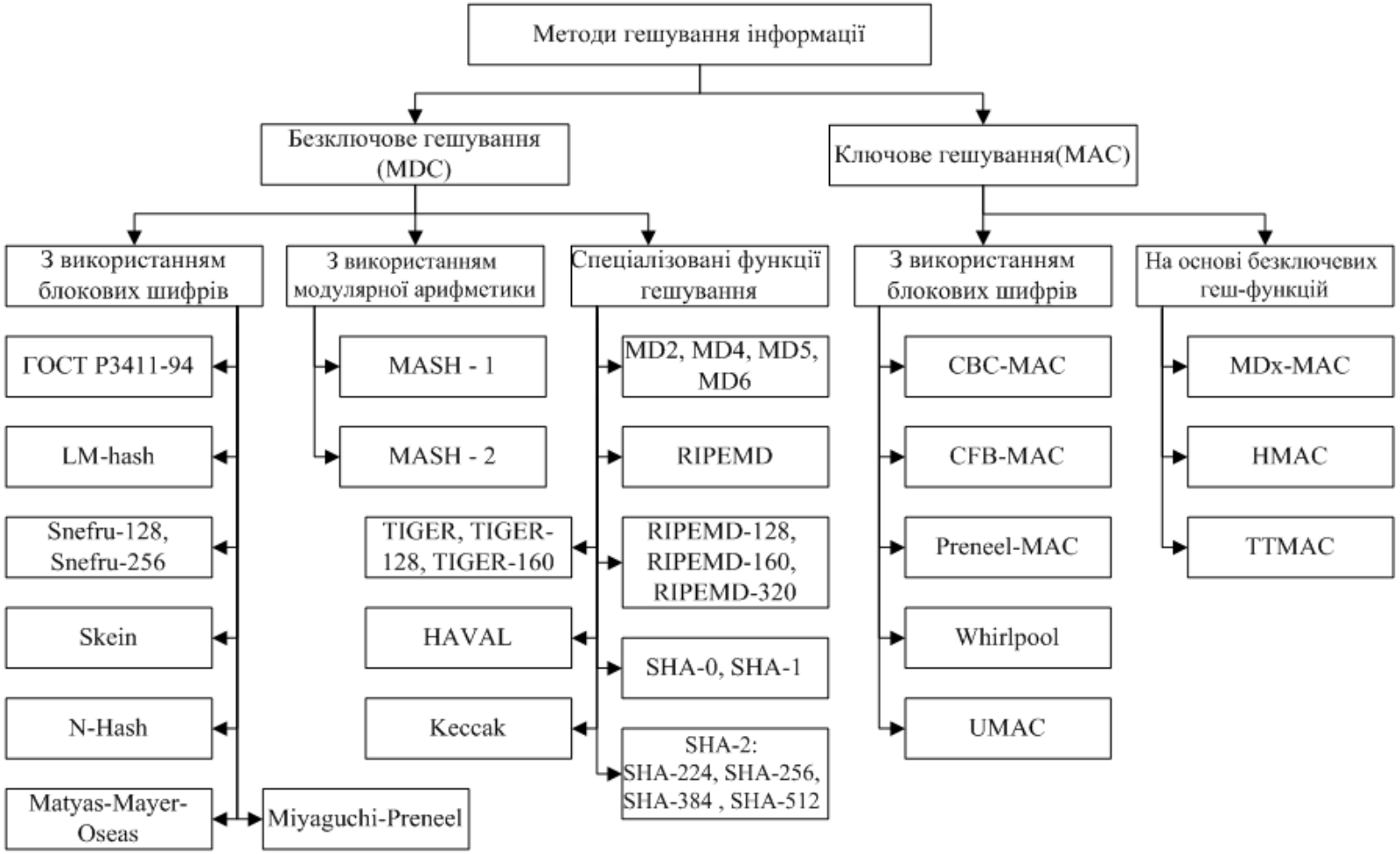



Рисунок 1.2 - Класифікація функцій гешування

У табл. 1.1. наведені основні результати з оцінки швидкодії найбільш відомих і безпечних алгоритмів.

Таблиця 1.1 - Швидкодія MAC-алгоритмів (кількість циклів $S$)

| Алгоритм | Довжина Мас-Коду $l_h$ (біт) | Довжина ключа $l_K$ (біт) | Тип ЕОМ | | | | |
|---|---|---|---|---|---|---|---|
| | | | Pentium 2 | PIII/Linux | Pentium4 | Xeon | AMD |
| TTMAC | 160 | 160 | 21 | 21 | 40 | 37 | 21 |
| UMAC-16 | 64 | 128 | 6.1 | 6.0 | 6.2 | 6.1 | 6.2 |
| UMAC-32 | 64 | 128 | 2.5 | 2.9 | 6.7 | 6.6 | 1.9 |
| 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-RIPE-MD | 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<br>384<br>512 | 512 | 40<br>84<br>84 | 39<br>84<br>84 | 40<br>124<br>124 | 39<br>132<br>132 | 33<br>72<br>72 |
| HMAC-TIGER | 192 | 512 | 24 | 21 | 28 | 26 | 20 |
| CBC MAC-RIJNDAEL (EMAC) | 128 | 128 | 24 | 26 | 26 | 27 | 31 |
| CBC MAC-DES | 64 | 56 | 62 | 61 | 72 | 69 | 54 |
| CBC MAC-SHACAL | 512 | 160 | 31 | 31 | 67 | 74 | 29 |

Застосовуючи універсальне гешування разом із блоковим симетричним шифруванням не гарантує збереження властивості універсальності результуючого MAC-коду. На сьогодні ці питання є мало дослідженими й потребують більш детального вивчення.

Невиконання умов даного хешування і зниження колізійних властивостей схеми UMAC може створити передумову для зниження рівня цілісності й автентичності інформації.

## 1.3. Модель формування кодів достовірності повідомлень при застовуванні UMAC.

Беручи до уваги склад алгоритму UMAC, формуємо код достовірності інформації (Tag). Використовуємо для обчислення таку функцію:

$$Tag = UMAC(K, M, Nonce, Taglen) = Y \oplus Pad,$$

де $K$ - це зашифрований ключ. Його довжина$Keylen$, яка рівна стандартній довжині зашифрованого ключа. У випадку використання алгоритму шифрування AES, $Keylen$ належить множині припустимих значень $\{16,\ 24,\ 32\}$ байт);

$M$ – інформаційне повідомлення, що підлягає автентифікації, представлене таким масивом-рядком $2^{67}$ біт ($2^{64}$ байт);

$Nonce$ – унікальне восьмибайтне число, різне для кожного вхідного повідомлення $M$;

$Taglen$ –це число з множини $\{4,\ 8,\ 12,\ 16\}$, яке формує довжину коду $Tag$ в байтах;

$Hash(K, M, Taglen)$ - це функція стандартного хешування повідомлення $M$, що використовує зашифрований ключ $K$;

$PDF(K,\ Nonce,\ Taglen)$ - дана команда формує псевдовипадкову підкладку $(Pad)$, враховуючи значення $Nonce$ та $K$;

“$\oplus$” – підсумок результатів гешування повідомлень $Y = Hash(K, M, Taglen)$ й сформованої підкладки $Pad = PDF(K, Nonce, Taglen)$, тобто

$$Tag = Hash(K, M, Taglen) \oplus PDF(K, Nonce, Taglen).$$

Довжина геш-коду$Y$, підкладки $Pad$ й коду $Tag$ належать множині припустимих значень $\{32, 64, 96, 128\}$ біт. Такі значення $Taglen$ відповідають випадку формуванню кодів достовірності повідомлень UMAC – 32, UMAC – 64, UMAC – 96 або UMAC – 128 відповідно.

Розглянемо схему формування геш-кодів $Y = Hash(K, M, Taglen)$ і підкладки $Pad = PDF(K, Nonce)$.

### 1.4. Схема формування геш-кодів

Обчислення значення функції $Hash(K, M, Taglen)$ ключового універсального гешування інформаційного повідомлення $M$і $K$ виконується в три етапи (застосовується три рівні (шару) ключового гешування): $Hash_{L1}$, $Hash_{L2}$ і$Hash_{L3}$ відповідно.

Другий рівень гешування $Hash_{L2}$ виконується тільки в тому випадку, якщо довжина повідомлення, яке гешується, $M$ перевищує 1 024 байт.

Довжина геш-коду $Y$ кратна 32 бітам, його значення $Y = Hash(K, M, Taglen)$ для будь-якої довжини $Taglen$ формується за допомогою об’єднання (конкатенації) декількох (від однієї до чотирьох) послідовностей $Y_{L3_i}$:

$$Y = Hash(K, M, Taglen) = Y_{L3_1} \| Y_{L3_2} \| \dots \| Y_{L3_{It}}, It = Taglen/4,$$

де $Y_{L3_i}$ – результат багаторівневого гешування повідомлення $M$ на $i$ -ій ітерації з використанням відповідних ключів, $i = 1,2,\dots,It$.

Структурна схема ітеративного формування геш-кодів $Y$, псевдовипадкової підкладки $Pad$ й коду достовірності повідомлень $Tag$ (рис.1.3).

Розглянемо формування геш-коду $Y_{L3_i}$ на $i$ -й ітерації. Для цього позначимо результат багаторівневого гешування на довільній $i$ -й ітерації в такий спосіб:

$$Y_{L3_i} = Y_{L3} = Hash_{L3}\left(K_{L3_1}, K_{L3_2}, Hash_{L2}\left(K_{L2}, Hash_{L1}(K_{L1}, M)\right)\right),$$

де $Hash_{L1}(K_{L1}, M)$, $Hash_{L2}(K_{L2}, Y_{L1})$ і $Hash_{L3}\left(K_{L3_1}, K_{L3_2}, Y_{L2}\right)$ – функції ключового гешування першого, другого й третього рівнів, які залежать від номера ітерації секретними ключами $K_{L1}$, $K_{L2}$, $K_{L3_1}$, $K_{L3_2}$ відповідно.

Функція ключового гешування першого рівня (Level 1) призначена для формування геш-коду $Y_{L1} = Hash_{L1}(K_{L1}, M)$ за введеним інформаційним повідомленням $M$, що підлягає автентифікації й поданому рядку від 1біт до $2^{67}$ біт ($2^{64}$ байт).

Функція ключового гешування другого рівня (Level 2) здійснює формування геш-коду $Y_{L2} = Hash_{L2}(K_{L2}, Y_{L1})$. Якщо довжина повідомлення $M$ дорівнює або менше 1 024 байт, цей рівень гешування не виконується і обчислюється відразу геш-код третього рівня:

$$Y_{L3_i} = Y_{L3} = Hash_{L3}\left(K_{L3_1}, K_{L3_2}, Hash_{L1}(K_{L1}, M)\right).$$

Функція ключового гешування третього рівня (Level 3) призначена для формування геш-коду $Y_{L3} = Hash_{L3}\left(K_{L3_1}, K_{L3_2}, Y_{L2}\right)$.

Обчислення відповідних геш-кодів $Y_{L1}$, $Y_{L2}$ і $Y_{L3}$ на кожному рівні реалізується під керуванням власного ключа: перший рівень управляється ключем$K_{L1}$, другий – $K_{L2}$, третій – двома ключами $K_{L3_1}$ й $K_{L3_2}$ відповідно.

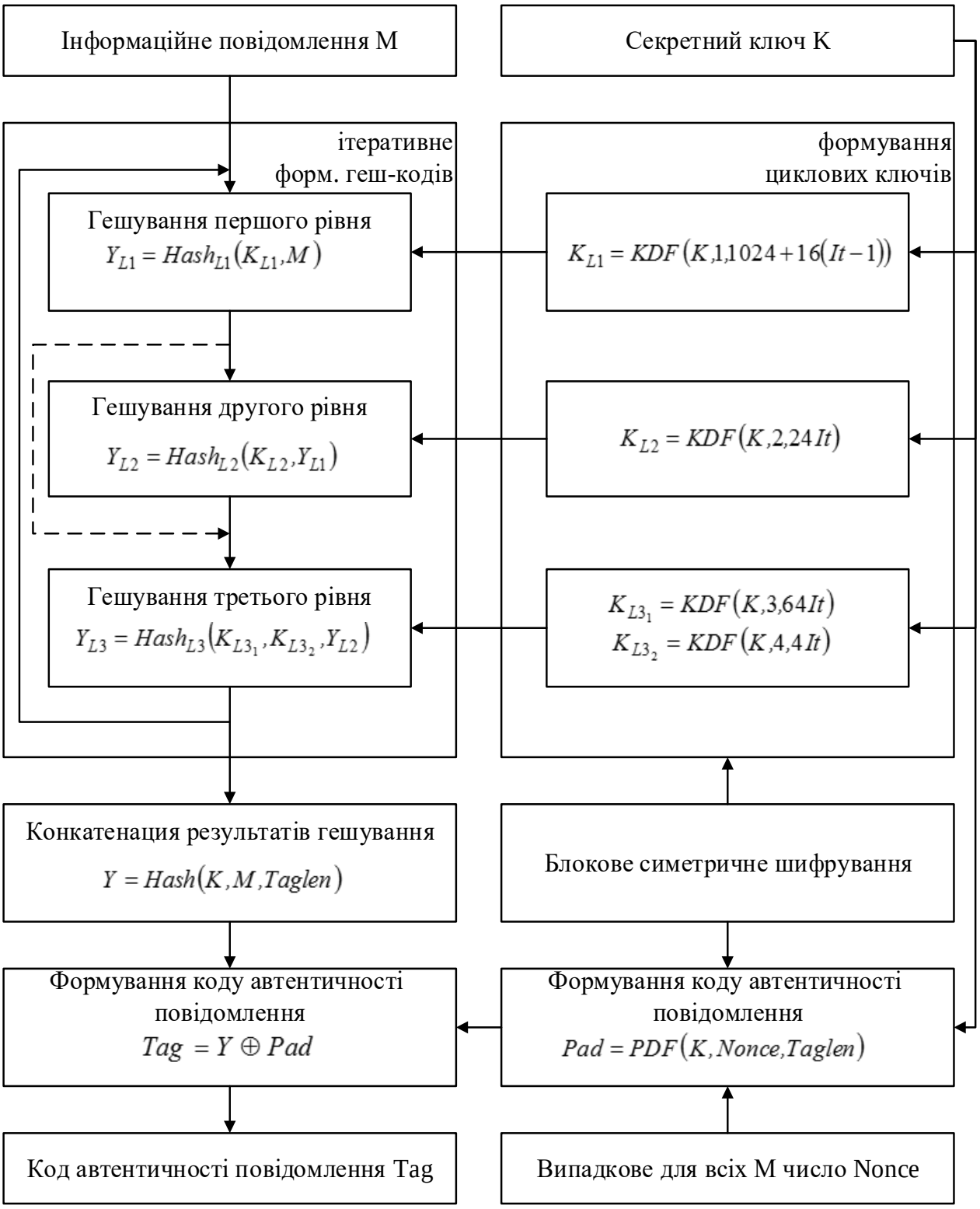



Рисунок 1.3 - Структурна схема ітеративного формування геш-кодів$Y$, псевдовипадкової підкладки $Pad$ й коду контролю цілісності й автентичності $Tag$

Ключові послідовності$K_{L1}$,$K_{L2}$,$K_{L3_1}$, $K_{L3_2}$ формуються за введеним секретним ключем $K$ довжини $Keylen$ байт із використанням спеціальної

функції $KDF(K, Index, Numbyte)$ (Key-Derivation Function – KDF), де $Index$ і $Numbyte$ – цілі додатні числа $\leq 2^{64}$:

$K_{L1} = KDF(K, Index, Numbyte)$,$Index = 1$, $Numbyte = 1024 + 16(It - 1)$;
$K_{L2} = KDF(K, Index, Numbyte)$,$Index = 2$, $Numbyte = 24It$;
$K_{L3_1} = KDF(K, Index, Numbyte)$,$Index = 3$, $Numbyte = 64It$;
$K_{L3_2} = KDF(K, Index, Numbyte)$,$Index = 4$, $Numbyte = 4It$.

Таким чином, для кожної $i$ -ї ітерації ($i = 1,2,\ldots, It$) використовується 4 ключі:$K_{L1}$,$K_{L2}$,$K_{L3_1}$, $K_{L3_2}$ по 1 024, 24, 64 і 4 байт відповідно (див. рис. 2.1).

Перший рівень гешування виконує розбиття масиву-рядка $M$ -розмірності до $2^{64}$ байт на блоки $M_i$ по 1 024 байт із наступним перетворенням кожного блоку функцією $NH(K_{L1}, M_i)$. Отримані результати $Hash_{L1_i} = NH(K_{L1}, M_i)$ поєднуються у рядок $Y_{L1} = Hash_{L1}(K_{L1}, M)$, коротший за інформаційну послідовність у 128 разів. Цей рядок і є результатом гешування першого рівня:

$$Y_{L1} = Hash_{L1}(K_{L1}, M) = NH(K_{L1}, M_0)\|NH(K_{L1}, M_1)\|\ldots\|NH(K_{L1}, M_{n-1}),$$

де $n = \left[\frac{Length(M)}{1024}\right]$;
$[x]$ – ціла частина числа$x$;
$Length(M)$ – байтова довжина інформаційного повідомлення $M$.

Значення функції $Hash_{L1_i} = NH(K_{L1}, M_i)$ обчислюється за таким правилом. Інформаційний блок $M_i$ розбивається на чотирьохбайтові підблоки так, що

$$M_i = M_{i_1}\|M_{i_2}\|\ldots\|M_{i_t},$$

де $t = \left[\frac{Length(M_i)}{4}\right]$. Тому $t = \left[\frac{1024}{4}\right] = 256$.

Аналогічним чином ключова послідовність $K_{L1}$ подається у вигляді послідовностей чотирьохбайтових підблоків:

$$K_{L1} = K_{L1_1} \| K_{L1_2} \| \ldots \| K_{L1_t}.$$

Після чого (приймаючи як початковий стан $Hash_{L1_i} = 0$) для всіх $j = 1,9,17\ldots, t-7$ виконуються такі операції:

$$Hash_{L1_i} = Hash_{L1_i} +_{64} ((M_{i_{j+0}} +_{32} K_{L1_{j+0}}) \times_{64} (M_{i_{j+4}} +_{32} K_{L1_{j+4}})),$$
$$Hash_{L1_i} = Hash_{L1_i} +_{64} ((M_{i_{j+1}} +_{32} K_{L1_{j+1}}) \times_{64} (M_{i_{j+5}} +_{32} K_{L1_{j+5}})),$$
$$Hash_{L1_i} = Hash_{L1_i} +_{64} ((M_{i_{j+2}} +_{32} K_{L1_{j+2}}) \times_{64} (M_{i_{j+6}} +_{32} K_{L1_{j+6}})), Hash_{L1_i} =$$
$$Hash_{L1_i} +_{64} ((M_{i_{j+3}} +_{32} K_{L1_{j+3}}) \times_{64} (M_{i_{j+7}} +_{32} K_{L1_{j+7}})),$$

де $+_{64}$, $+_{32}$ – операції підсумовування за модулями $2^{64}$ і $2^{32}$ відповідно;

$\times_{64}$ – операція множення за модулем $2^{64}$. У результаті обчислень формуються восьмибайтні значення $Hash_{L1_i}$, спрощену схему такого формування представлено на рис.1.4 [24, 29, 38];

Встановлено, що розглянута функція ключового гешування $NH$ належить до класу універсальних гешуючих функцій.

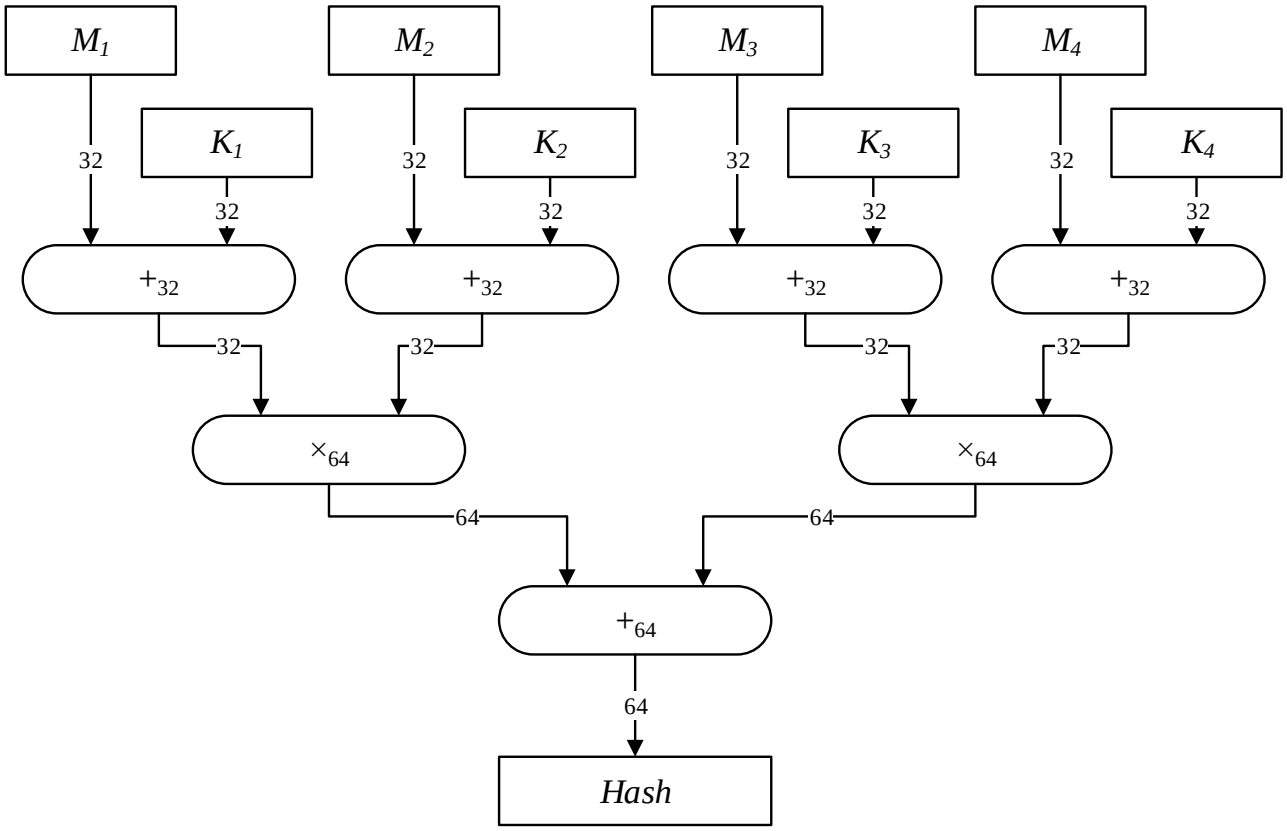



Рисунок 1.4 - Спрощена схема формування геш-образу $Hash_{L1_i} = NH(K_{L1}, M_i)$

Як вхідні дані дана функція використовує: $Wordbits \in [64,\ 128]$; $Maxwordrange$ - число, яке несе позитивне значення і менше $2^{Wordbits}$; $k$ - число, яке залежить від ключа $K_{L2}$,а це будь-яке число з даного діапазону$[0,\ldots,\ prime(Wordbits) - 1]$; $prime(x)$ - найбільше ціле число, яке є менше $2^x$;$M_P = Y_{L1} = Hash_{L1}(K_{L1}, M)$ - гешоване значення.

Згідно алгоритму UMAC, як $prime(x)$ використовуються такі константи: $prime(36) = 2^{36} - 5,\quad prime(64) = 2^{64} - 59,\quad prime(128) = 2^{128} - 159.$ Байтову довжину $M_P$ позначимо $Bytelength(M_P)$. Залежно від довжини $M_P$ використовуються наступні функції хешування 2-го рівня:

- $M_P$ (довжина даних) не більше за $2^{17}$байт, тоді гешування $Poly$ рахується з $Wordbits = 64$; $Maxwordrange = 2^{64} - 2^{32}$;$k = k64$ - ряд, який утворюється за допомогою 1-ших вісьмома байтами ключа $K_{L2}$ та спеціальною восьмибайтною маскою;

- $M_P$ - більше за $2^{17}$байт, але менше за $2^{64}$, тоді дані обробляються за допомагою функції поліноміального хешування $Poly(64, 2^{64} - 2^{32}, k64, M_P)$, а а байти даних, що залишилися обробляються $Poly$ функцією з $Wordbits = 128$ параметрами $Maxwordrange = 2^{128} - 2^{96}$; $k = k^{128}$; – ряд, що формується з останніх 16 байтів $K_{L2}$.

- Дані $M_P$діляться на два блоки $Wordbytes = Wordbits/8$байтів:

$$M_P = M_{P_1} \| M_{P_2} \| \dots \| M_{P_n}$$

де $n = Bytelength(M_P)/Wordbytes$,

Результат хешування – це дані поліноміальної ф-ції:

$$Y_{L2} = \left(M_{P_n} + kM_{P_{n-1}} + \dots + k^{n-1}M_{P_1} + k^n\right) mod(p),$$

Знаходження геш-значення $Y_{L2} = Poly_n$ являється ціле число, яке входить в діапазон $[0, \dots, prime(Wordbits) - 1]$.

У роботах [8, 31, 38] доведено, що розглянута функція поліноміального ключового гешування $Poly(Wordbits, Maxwordrange, k, M_P)$ належить до класу універсальних функцій гешуванняю

$Hash_{L3}\left(K_{L3_1}, K_{L3_2}, Y_{L2}\right)$ - це 3-й рівень хешування. Він відбувається шляхом перетворення даних на вхідні данідвного коду $Y$ – це фіксована довжина 32біта.

Дані, які гешуються, $Y_{L2}$ й ключова послідовність $K_{L3_1}$одинаково діляться на 8 блоків, і кожний цей блок – це повне число $Y_{L2_i}$ й $K_{L3_{1_i}}$, $i = 1,2, \dots, 8$.

Геш-значення $Y_{L3}$ обчислюється в такий спосіб:

$$Y_{L3} = \left(\left(\left(\sum_{i=1}^{m} Y_{L2_i} K_{L3_{1_i}}\right) mod(prime(36))\right) mod(2^{32})\right) xor(K_{L3_2}),$$

де $(x)xor(y)$ – операція “виключення АБО” зі значеннями $x$ й $y$.

У роботах [8, 30, 38] показано, що розглянута функція ключового гешування $Y_{L3} = Hash_{L3}\left(K_{L3_1}, K_{L3_2}, Y_{L2}\right)$ належить до класу універсальних функцій гешування.

1.5 Схема формування псевдовипадкової підкладки (PDF: Pad-Derivation Function).

Функція $PDF(K, Nonce, Taglen)$ призначена для формування псевдовипадкової підкладки $Pad$, використовуваної на заключному етапі формування коду достовірності повідомлення.

У якості вхідних даних використовується секретний ключ $K$ довжини $Keylen$ байт і випадкове восьмибайтне число$Nonce$, а також ціле число$Taglen$. Процес створення псевдовипадкової підкадки заключається у створенні самого підключа:

$$K' = KDF(K, Index, Numbyte), Index = 0, Numbyte = Keylen$$

Тепер формуємо шифрування та послідовність ключових бітів $Nonce$ на сформованому підключі $K'$, тобто:

$$Pad = PDF(K, Nonce, Taglen) = Enchiper(KDF(K, 0, Keylen), Nonce).$$

Для дослідження колізійних властивостей кодів контролю цілісності й автентичності в цій роботі використовується математичний апарат, в основі якого лежить принцип масштабованості, що полягає в заміні об’єкта досліджень його зменшеною моделлю. Із цією метою у дипломній роботі пропонується зменшена модель схеми UMAC (mini-UMAC), її програмна реалізація. З використанням математичного апарата в наступному розділі буде доведено, що результуючі коди автентичності повідомлень $Tag = UMAC(K, M, Nonce, Taglen) = Y \oplus Pad$ формуються не рівноймовірно.

## 1.6 Висновки за розділом 1

Проведений аналіз і дослідження показали, що сучасні телекомунікаційні системи й мережі, використовуючи останні досягнення в розвитку електронних комунікацій і IT-технологій, постійно розширюють спектр послуг,включаючи з обслуговування суб'єктів автоматизованої інформаційної взаємодії, забезпечення доступу до різних мультимедійних сервісів і технологій, підтримки віддалених користувачів тощо.

Особливо нагальними ці питання є у телекомунікаційних мережах спеціального призначення, де відмова в обслуговуванні або вихід конкретних параметрів якості за встановлені межі може привести до катастрофічних наслідків у фінансовому секторі, промисловості, енергетичному комплексі та ін. Як приклад такої системи можна навести створювану в Україні Національну систему міжнародних електронних платежів (НСМЕП), що являє собою велику територіально розподілену мультисервісну телекомунікаційну систему спеціального призначення, яка використовує високопродуктивні обчислювальні комплекси й складні механізми комплексного захисту інформаційних технологій.

В дослідженнях видно, що переважна більшість порушень безпеки (понад 87 %) стосуються порушення цілісності та конфіденційності інформаційних даних. Лідируючу позицію з реалізації загроз мережної безпеки займають відхилення, що приводять як до витоку закритої інформації, так і до нав'язування хибної інформації або неправильної роботи компонентів телекомунікаційної системи.

Найбільшу небезпеку інформаційним ресурсам у сучасних телекомунікаційних системах і мережах становлять загрози цілісності та автентичності інформації. Це визначене тим, що порушення достовірності оброблюваних і переданих даних, неправомірне спотворення або фальсифікація, знищення чи поширення частинки інформації, наносять істотного матеріального і морального збитку різним суб'єктам державі,

юридичним і фізичним особам, що беруть участь у процесах інформаційної взаємодії.

Виникає протиріччя між зростаючими ймовірнісно-часовими вимогами до перспективних механізмів контролю цілісності й автентичності інформації в умовах безперервного вдосконалювання загроз інформаційної безпеки й реальним станом існуючих моделей, методів і обчислювальних алгоритмів, застосовуваних у протоколах мережної безпеки,

Проведений аналіз показав, що одним з найбільш ефективних підходів до побудови механізмів контролю цілісності й автентичності інформації є ключове й безключове гешування даних. Практичне використання відповідних механізмів безпеки дозволяє без залучення додаткових засобів забезпечувати необхідні показники оброблюваних і переданих даних.

# 2 РЕАЛІЗАЦІЯ МОДЕЛІ І МЕТОДУ ДЛЯ СФОРМУВАННЯ КОДУ АВТЕНТИЧНОСТІ І КОНТРОЛЮ ЦІЛІСНОСТІ, ЩО БАЗУЄТЬСЯ НА МОДУЛЬНИХ ПЕРЕТВОРЕННЯХ

Проведені дослідження показали, що застосування багатошарових схем ключового гешування дозволяє будувати ефективні механізми контролю цілісності й автентичності інформації в телекомунікаційних системах і мережах. Однак відомі багатошарові конструкції (на прикладі алгоритму UMAC) поряд з високими показниками швидкодії і криптографічної стійкості за рахунок застосування криптографічного шару втрачають властивості універсального гешування, що приводить до погіршення властивостей, що впливають на формування повідомлень. Тому перспективним напрямком досліджень у цьому змісті є розробка й теоретичне обґрунтування нових схем ключового гешування, що дають можливість забезпечити як високі колізійні властивості (зі збереженням властивостей універсального гешування), так і високі показники безпеки.

У цьому розділі досліджені властивості модулярних перетворень і побудованих на їх основі методів безключового гешування інформації (MASH-1 і MASH-2), а також методів ключового гешування, побудованих на основі алгоритмів MASH-1 і MASH-2 при зміні вектора ініціалізації як секретних ключових даних. Розглянуто різні види циклових функцій у схемі ітеративного гешування, побудовані з використанням модулярних перетворень, стійкість яких еквівалентна розв’язанням однієї з відомих теоретико-складних задач. Встановлено, що застосування циклових функцій на модулярних перетвореннях дозволить будувати універсальні класи, що є функціями хешування, що і дозволять забезпечити високі показники безпеки та застосовувати моделі доведеної стійкості, і – при виконанні визначених обмежень на параметри модулярних перетворень забезпечити високі колізійні властивості. На основі отриманих результатів проведених досліджень універсального ключового

гешування з доведеною стійкістю, а також реалізовані метод і модель для формування кодів цілісності таних, їх контролю та автентичності.

## 2.1. Дослідження властивостей модулярних перетворень і методів гешування інформації на їх основі

Модулярні перетворення широко використовуються при побудові криптографічних алгоритмів перетворення інформації, в тому числі при побудові асиметричних засобів захисту інформації й протоколів розповсюдження ключових даних [**Error! Reference source not found.**, 24], для формування псевдовипадкових послідовностей [**Error! Reference source not found.**], методів гешування та інших механізмів захисту інформації [15, **Error! Reference source not found.**, 31].

Проведемо дослідження геш-функцій, що використовують модулярні перетворення, проаналізуємо особливості їх побудови й можливості застосування при побудові багатошарових схем взаємозамінного хешування кодів контролю цілісності та автентичності даних.

Аналіз літературних джерел [**Error! Reference source not found.**, 20, 15, 25, 25, **Error! Reference source not found.**, 29] показує, що модулярні перетворення застосовуються на сьогодні при побудові безключових геш-функцій. Так, у четвертій частині міжнародного стандарту ISO/IEC 10118-4 визначено дві безключові функції гешування MASH-1 і MASH-2, які використовують модулярну арифметику, а саме модульне піднесення у степінь для побудови геш-коду [24, **Error! Reference source not found.**, 29].

Найбільш ефективні схеми ітеративного гешування з використанням модулярної арифметики базуються на модулярному піднесенні у квадрат, тобто на застосуванні циклової функції виду:

$$f(x_i, H_{i-1}) = (x_i \oplus H_{i-1})^2 \, mod \, N, \qquad (2.1)$$

або

$$f(x_i, H_{i-1}) = ((x_i)^2 \oplus H_{i-1})^2 \, mod \, N. \quad (2.2)$$

У визначених міжнародним стандартом ISO/IEC 10118-4 геш-функціях MASH-1 і MASH-2 використані такі циклові функції:

$$f(x_i, H_{i-1}) = \left(\left((x_i \oplus H_{i-1}) \vee A\right)^2 mod \, N\right) \perp n \oplus H_{i-1} \quad (2.3)$$

і

$$f(x_i, H_{i-1}) = \left(\left((x_i \oplus H_{i-1}) \vee A\right)^{2^8+1} mod \, N\right) \perp n \oplus H_{i-1}, \quad (2.4)$$

де $\vee$ – операція побітного логічного АБО;

$\oplus$ – підсумовування (XOR);

$\perp n$ – збереження молодших $n$-розрядів $m$-розрядного результату.

Алгоритм обчислення значення геш-функції MASH-1 має такий вид [8, 15, **Error! Reference source not found.**]. Вхід. Двійковий рядок x довжиною $0 \leq b \leq 2^{n/2}$. Вихід. $n$-розрядний геш-код від рядка $x$, довжиною приблизно дорівнює довжині модуля $N$.

Попередня обробка – це доповнення рядка $x$ нульовими бітами і одержання двійкового рядка.

Щоб розширити кожний блок ( $x_i$) в блок $y_i$, необхідно втавити комбінацію 1111. Але останній блок формується винятковою комбінацією 1010.

Закінчення. Як геш-код береться n-розрядний блок $H_{t+1}$.

Два алгоритми MASH-1 та MASH-2 відрізняються один від одного показником степенем у функції (2.4). У табл. 2.1. наведені результати порівняльного аналізу показників ефективності безключових функцій гешування, та геш-функцій на модулярній арифметиці MASH-1 і MASH-2.

Наведено оцінку моделей безпеки, введених у проекті NESSIE [24, 29, 38].

Таблиця 2.1 - Результати порівняльного аналізу деяких безключових функцій гешування

| Геш-функція | Довжина геш-коду | Застосовувані перетворення | Швидкість обробки даних | Модель безпеки (за NESSIE) |
|---|---|---|---|---|
| SHA-2 | 256, 384, 512 | Логічні й арифметичні | $10^8..10^9$ біт/з | Практична таємність (Practical Security) |
| Whirlpool | 512 | У скінченних полях Галуа | $10^7..10^8$ біт/з | Практична таємність (Practical Security) |
| ДЕРЖСТАНДАРТ 34311-95 | 256 | Блокове симетричне шифрування | $10^7..10^8$ біт/з | Практична таємність (Practical Security) |
| RIPEMD-160 | 160 | Логічні й арифметичні | $10^8..10^9$ біт/з | Практична таємність (Practical Security) |
| MASH-1 | Визначається розмірністю модуля перетворень. | Модулярне піднесення у квадрат | $10^5..10^6$ біт/з | Безпека доказової стійкості. Якщо відповідають обмеженням на RSA-подібні системи. («Provable» Security) |
| MASH-2 | Визначається розмірністю модуля перетворень. | Модулярне піднесення у степінь $2^8+1 = 257$ | $10^4..10^5$ біт/з | Безпека доказової стійкості. Якщо відповідають обмеженням на RSA-подібні системи. («Provable» Security) |

Проведений аналіз показав, що недоліками алгоритмів MASH-1 і MASH-2 низькі швидкості формування геш-кодів. Фактично вона визначається швидкістю RSA-подібного шифрування, яке на 2 – 3 порядки нижче швидкості шифрування сучасними блоково-симетричними шифрами. Необхідно відзначити, що алгоритми гешування MASH-1 і MASH-2 не повною мірою відповідають обмеженням на параметри модульного піднесення у степінь, які встановлені для RSA-систем (а відповідно, й забезпечуваної моделі безпеки доведеної стійкості). Дійсно, за специфікацією криптографічної RSA-системи, що

забезпечує безпеку доведеної стійкості (за моделлю безпеки NESSIE), значення модульної експоненти *e* повинне бути обране з умови, що

$$gcd(e,\varphi(N)) = 1, \tag{2.5}$$

де *gcd(x,y)* – найбільший загальний дільник чисел *x* і *y*.

Значення експоненти *e* не повинне містити загальних дільників із числом (значенням функції Ейлера) $\varphi(N)$:

$$\varphi(N) = (p-1)(q-1),\ N = pq. \tag{2.6}$$

За специфікацією алгоритмів MASH-1 і MASH-2 ця умова може не виконуватися. Так, наприклад, в алгоритмі MASH-1 показник степені $e = 2$, що при непарних значеннях простих чисел *p* і *q* завжди порушує умову (2.5).

В алгоритмі MASH-2 показник степені $e = 2^8+1 = 257$, що також не завжди веде до виконання умови (2.5): значення функції Ейлера може, наприклад, ділитися на показник степені $e = 2^8+1 = 257$.

Таким чином, модель безпеки доведеної стійкості (за класифікацією моделей безпеки NESSIE) може бути застосована до алгоритмів MASH-1 і MASH-2 тільки умовно. Повної відповідності завдання знаходження прообразу або секретного ключа у схемі гешування не спостерігається.

Розглянемо циклові функції MASH-1 і MASH-2 стосовно побудови ключових універсальних функцій гешування. Оберемо варіант гешування, коли початковий стан (вектор ініціалізації) задається деяким ключовим правилом, тобто $H_0 = Key$. У цьому випадку маємо деякий клас геш-функцій, що залежать від параметра *Key*.

Для експериментальної перевірки властивостей універсального гешування була розроблена програмна реалізація алгоритмів гешування MASH-1 і MASH-2 при зміні значень вектора ініціалізації секретним ключем. Лістинг коду програмної реалізації наведений у дод. А.

Для виконання експериментальних досліджень обрані такі параметри:$p = 17$,$q = 19$, $N = 323$. Дослідження полягали в перевірці умов універсального гешування при повному перебиранні всіх значень векторів ініціалізації ($Key = 0, \ldots, 2^m - 1, m = 8$) за вибіркою значень інформаційних блоків. Отримані результати зведено в табл. 2.2.

Таблиця 2.2 - Результати зроблених досліджень на основі колізійних властивостей ключового гешування, побудованих на основі алгоритмів MASH-1 і MASH-2 при зміні значень вектора ініціалізації секретним ключем

| | На основі алгоритму MASH-1 | На основі алгоритму MASH-2 |
|---|---|---|
| $\widetilde{m}(n_1)$ | 41,42 | 0 |
| $\widetilde{D}(n_1)$ | 42,74 | 0 |
| $P_\partial = P(\lvert\widetilde{m}(n_1) - m(n_1)\rvert < 5)$ | 0,98 | ≈ 1 |
| $\widetilde{m}(n_2)$ | 3,99 | 1 |
| $\widetilde{D}(n_2)$ | 0,01 | 0 |
| $P_\partial = P(\lvert\widetilde{m}(n_2) - m(n_2)\rvert < 0{,}025)$ | 0,99 | ≈ 1 |
| $\widetilde{m}(n_3)$ | 0,26 | 0,31 |
| $\widetilde{D}(n_3)$ | 0,21 | 0,22 |
| $P_\partial = P(\lvert\widetilde{m}(n_3) - m(n_3)\rvert < 0{,}1)$ | 0,97 | 0,97 |

Дослідження проводилися над вибіркою, обсягу $N$ = 100, для формування кожного елемента вибірки розраховувався максимум з множини з $M = 100$ кортежів елементів. Таким чином, загальний обсяг формованих наборів склав $NM = 10^4$.

Для проведених $N = 100$ експериментів оцінювалися математичні сподівання $m(n_1)$, $m(n_2)$ і$m(n_3)$, дисперсії $D(n_1)$, $D(n_2)$ і $D(n_3)$, а також для фіксованої точності $\varepsilon$ розраховувалися за формулою $P(\lvert\widetilde{m}(n_i) - m(n_i)\rvert < \varepsilon), i = 1, 2, 3$.

Проаналізуємо отримані результати статистичних досліджень, зіставимо їх з теоретичним оцінками: числом $\mathrm{P}_{\text{кол}} \cdot |H| = 1$ (1-ший критерій), $|H|/|B| = 1$ (2-гий критерій) та число $P_{кол} \cdot |H| = 1$ (3-тій критерій).

Як видно з наведених у табл. 2.2 даних, реалізація схеми ключового гешування, з допомогою алгоритму MASH-1 при зміні значень вектора ініціалізації секретним ключем не дозволяє забезпечити високі колізійні властивості. Кількість колізій, що виникають, суттєво вища верхньої теоретичної границі, як за першим, так і за другим критеріями, отже, така конструкція не є схемою універсального й, тим більше, строго універсального гешування. Цей результат отриманий з високою довірчою ймовірністю$P_{\text{д}} = P(|\widetilde{m}(n_i) - m(n_i)| < \varepsilon) > 0{,}9,\ i = 1,\ 2,\ 3$ для високої точності. Так, для 1-го критерію довірчий інтервал склав $41{,}42 \pm 5$ (довірча ймовірність 0,98), для другого критерію – $3{,}99 \pm 0{,}025$ (довірча ймовірність 0,99), для третього – $0{,}26 \pm 0{,}1$ (довірча ймовірність 0,97). Розглянута схема ключового гешування на основі алгоритму MASH-1 при зміні значень вектора ініціалізації секретним ключем задовольняє тільки третьому критерію ($\widetilde{m}(n_3) = 0{,}26$).

Використання ключового гешування на основі алгоритму MASH-2 при зміні значень вектора ініціалізації секретним ключем навпаки забезпечує високі колізійні характеристики універсального гешування. За всіма трьома критеріями отримані оцінки лежать нижче верхньої теоретичної границі $\widetilde{m}(n_i) < 1,\ \ i = 1,\ 2,\ 3$. Це положення підтверджено, практично, з 100 відсотковою ймовірністю. Так для першого й другого критеріїв значення дисперсії $D(n_1)$ і $D(n_2)$, що характеризують розсіювання значень кількості правил гешування (правил формування МАС), при яких виконуються рівності, щодо їх математичних сподівань $m(n_1)$ і $m(n_2)$ відповідно, дорівнюють нулю, що означає ідентичність отриманих результатів у всіх проведених дослідах і, практично вірогідно маємо $m(n_1) = 0,\ m(n_2) = 0$. Отримана оцінка за третім критерієм також лежить нижче верхньої теоретичної оцінки ($\widetilde{m}(n_3) = 0{,}31$) і це значення підтверджене з високою довірчою ймовірністю $P_{\text{д}} = P(|\widetilde{m}(n_3) -$

$m(n_3)| < 0{,}1) = 0{,}97$ для фіксованої точності (довірчий інтервал рівний $0{,}31 \pm 0{,}1$).

Пояснення такої поведінки модулярних перетворень у схемах MASH-1 і MASH-2 лежить в обраних параметрах модульної експоненти. Так, для алгоритму MASH-1 циклова функція завжди порушує умову (2.5). В алгоритмі MASH-2 показник ступеня встановлений рівним $e = 2^8+1 = 257$, що для обраних параметрів $p = 17,\ \ q = 19,\ \ N = 323$ задовольняє обмеженню (2.5): $gcd(\,e, \phi(N)) = gcd(\,257,\ 288) = 1$. Отже, ключове гешування, побудоване на основі модулярних перетворень, у деяких випадках дозволяє забезпечити властивості універсального й строго універсального гешування. Для виконання цих властивостей необхідне виконання умови (2.5), що й демонструє при обраних параметрах схема на основі алгоритму MASH-2.

Аналіз даних, наведених у табл. 2.2, дозволяє стверджувати про адекватність отриманих експериментальних результатів. Для фіксованої точності $\varepsilon$ отримані високі значення довірчої ймовірності, що говорить про обґрунтованість і вірогідність отриманих результатів, відповідність їх статистичним властивостям усієї генеральної сукупності даних.

Таким чином, виконані дослідження показали, що застосування перетворень із використанням модулярної арифметики дозволяє будувати універсальні й строго універсальні класи функцій гешування, які, з одного боку, дозволяють забезпечити високі колізійні властивості, а з іншого – при виконанні визначених обмежень на значення модулярної експоненти гарантують високі показники безпеки й застосовність моделі доведеної стійкості.

Основними недоліками подібних конструкцій є:

– дуже висока складність перетворень, зумовлена використанням як циклової функції модулярного піднесення у степінь. Фактично складність застосовуваних перетворень вище складності блокового симетричного шифрування на 2 – 3 порядки, що й зумовлює відповідне підвищення часу формування кодів автентифікації повідомлень (див. табл. 2.1);

– формування кодів автентифікації повідомлень із використанням ключового гешування, побудованого на основі алгоритму MASH-1 зі змінними векторами ініціалізації, не дозволяє будувати універсальні й строго універсальні класи геш-функцій (див. табл. 2.2). Це зумовлено використанням у якості показника степені циклової функції значення e = 2, що при непарних значеннях простих чисел p і q завжди порушує умову (2.5);

– формування кодів автентифікації повідомлень із використанням ключового гешування, побудованого на основі алгоритму MASH-2 зі змінюваними векторами ініціалізації, у деяких випадках (при виконанні умови (2.5)) дозволяє будувати універсальні й строго універсальні класи геш-функцій (див. табл. 2.2). Однак не для всіх значень початкових параметрів (простих чисел $p$ і $q$) ця умова здійсненна;

– модель безпеки доведеної стійкості (за класифікацією моделей безпеки NESSIE) до розглянутих методів гешування може бути застосована тільки при виконанні визначених обмежень на значення експоненти й модуля перетворення, тобто при виконанні обмежень для RSA-подібних систем на параметри модульного піднесення у степінь. Повної відповідності завдання знаходження прообразу або секретного ключа в схемі гешування і теоретико-складної задачі дискретного логарифмування (або завдання RSA) для розглянутих методів гешування не спостерігається.

Таким чином, як показують проведені дослідження, застосування модулярних перетворень потенційно може розв’язати завдання побудови стійких універсальних функцій гешування з доведеною стійкістю, однак для цього необхідно усунути виявлені протиріччя. Перспективним напрямком подальших досліджень є розробка методу ключового універсального гешування з доведеною стійкістю на основі модулярних перетворень із урахуванням виявлених закономірностей.

2.2. Реалізація ітеративного ключового гешування з доведеною стійкістю з використанням модулярних перетворень

В основі реалізованого методу універсального гешування лежить ітеративна схема формування геш-коду із цикловою функцією, побудованою з використанням модулярних перетворень.

Використаний алгоритм дозволяє суттєво прискорити процедуру обчислення циклових функцій, що лежать в основі методу універсального гешування.

У табл. 2.3 подано залежності складності реалізації операції піднесення у степінь через ланцюжок множень. Там же наведено порядок модуля перетворення, потрібного для забезпечення мінімально необхідного рівня безпеки.

Дані в другому рядку табл. 2.3 наведені з умови еквівалентності (за обчислювальною складністю) операції піднесення у квадрат і операції множення.

Аналіз даних табл. 2.3 показує, що реалізація використаного методу універсального гешування через традиційний алгоритм піднесення у степінь обчислювально недосяжна. Кількість множень, яку потрібно виконати для обчислення одного значення циклової функції навіть при мінімальному рівні безпеки, перевищує можливості найсучасніших обчислювальних систем.

Таблиця 2.3 - Оцінки обчислювальної складності реалізації операції піднесення у степінь різними методами

| Метод піднесення у степінь | Порядок модуля перетворень / рівноцінна довжина ключа симетричного криптоалгоритму | | |
|---|---|---|---|
| | 1024 / 80 | 3072 / 128 | 15360 / 256 |
| Ланцюх добутків | 10308 | 10924 | 104623 |
| Швидкий алгоритм піднесення в ступінь | 2046 | 6142 | 30718 |

Результати проведених досліджень наведені в табл. 2.4 та 2.5, практичної реалізації алгоритмів залежно від кількості логічних операцій і структури схем хешування.

Таблиця 2.4 - Кількість логічних операцій при практичній реалізації алгоритмів гешування

| Алгоритм | Логічна операція | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | AND | OR | XOR | ROTR | SHR | + | ROLs | NOT | MOD | MUL |
| MD-5 | - | - | - | - | - | 960 | 256 | - | - | - |
| RIPEMD-128 | - | - | - | - | - | 396 | 128 | - | - | - |
| RIMEMD-160 | - | - | - | - | - | 650 | 320 | - | - | - |
| SHA-1 | 400 | 240 | 320 | - | - | 320 | 160 | - | - | - |
| SHA-256 | 320 | - | 448 | 384 | - | 448 | - | 64 | - | - |
| SHA-384 | 400 | - | 560 | 480 | - | 560 | - | 80 | - | - |
| SHA-512 | 400 | - | 560 | 480 | - | 560 | - | 80 | - | - |
| MASH1 | - | 6 | 4 | - | 6 | 1 | 1 | - | 6 | 5 |
| MASH2 | - | 6 | 4 | - | 6 | 1 | 1 | - | 6 | 260 |
| GMAC | 2 | 5 | 56 | - | 21 | 171 | 22 | 2 | 30 | 197 |

Таблиця 2.5 - Кількість кроків і циклів в схемах гешування

| Алгоритм | Загальна кількість кроків | Кількість циклів r | Кількість кроків у циклі s | Константи які відрізняються для кожного |
|---|---|---|---|---|
| MD-5 | 64 | 4 | 16 | Крок |
| RIPEMD-128 | 64 | 4 | 16 | Цикл |
| RIMEMD-160 | 80 | 5 | 16 | Цикл |
| SHA-1 | 80 | 4 | 20 | Цикл |
| SHA-2 - 256 | 64 | 1 | 64 | Крок |
| SHA-2 - 384 | 80 | 1 | 80 | Крок |
| SHA-2 - 512 | 80 | 1 | 80 | Крок |

Використані алгоритми формування геш-кодів реалізовані в програмному вигляді, лістинг програмного коду наведений у дод. А.

## 2.3. Обґрунтування що до використання моделі каскадного формування MAC із використанням модулярних перетворень

Проведені дослідження показали, що використання модулярних перетворень дозволяє реалізувати доказове стійке гешування інформації, що задовольняє колізійним властивостям універсальних геш-функцій. Доведений рівень безпеки обґрунтовується відомістю завдання знаходження прообразу й/або завдання відновлення секретних ключових даних до розв'язанням однієї з відомих теоретико-складних задач.

У той же час як показали проведені дослідження, універсальне гешування з використанням модулярних перетворень має істотний недолік – високу обчислювальну складність формування геш-кодів. Перспективним напрямком у цьому розумінні є розробка багатошарових схем універсального гешування, кінцевому етапі формування геш-коду. Це, як показано нижче, з одного боку, забезпечує високі колізійні властивості результуючої схеми формування кодів автентичності повідомлень та контролю їх цілісності іншого– забезпечує високі показники швидкодії й доведений рівень безпеки використовуваних перетворень.

Пояснимо останню тезу такими міркуваннями. Нехай перші шари універсального гешування (як і в методі-прототипі UMAC) реалізуються з використанням високошвидкісних (але криптографічно слабких) схем Картера–Вегмана, поліноміальних конструкцій тощо (див. рис. 2.1). Припустимо, що складність таких перетворень дорівнює складності схеми UMAC, тобто порядку 6 циклів на один байт інформаційних даних. Насправді ця оцінка сильно завищена, оскільки найбільш витратним у схемі UMAC є останній криптографічних шар, з використанням алгоритму шифрування AES. Отже, оцінка в 6 циклів на один байт оброблюваних даних є оцінкою в найгіршому разі, тобто верхньою оцінкою. Припустимо також, що на останньому, криптографічному етапі замість алгоритму AES використовується розглянута

схема доведеної стійкості універсального гешування на основі модулярних перетворень (див. модель на рис. 2.1).

Для оцінки складності завершального етапу формування MAC застосуємо дані табл. 2.6.

Тоді результуюча складність як кількість циклів процесора на один оброблюваний байт даних є усередненою оцінкою за всіма шарами перетворення в використаній каскадній конструкції обчислення кодів контролю цілісності й автентичності даних. Оскільки основна частина оброблюваних даних надходить тільки на перші шари перетворення (див. модель на рис. 2.1), а останній, криптографічний шар з модулярними перетвореннями застосовується лише один раз для обробки результату гешування попередніми шарами схеми, оцінка складності для великих обсягів оброблюваних даних буде прагнути до оцінки складності схеми UMAC. Для підтвердження наведених міркувань у табл. 2.6 подана приблизна оцінка складності формування кодів контролю цілісності й автентичності даних використаною схемою з використанням модулярних перетворень (див. рис. 2.1).

Дані, зазначені в табл. 2.6, отримані розрахунковим шляхом за допомогою усереднення верхньої оцінки складності універсального гешування на перших шарах перетворень (6 циклів на один байт) і оцінки складності модулярних перетворень (з використанням циклових функцій).

Прочерками в табл. 2.6 позначені місця, у яких гешування на модулярних перетвореннях (криптографічний шар) не може бути виконане.


Рисунок 2.1- Використана схема кодів з використанням модулярних перетворень

Таблиця 2.6 - Оцінка складності формування MAC схемою в кількості S-циклів 32-розрядного процесора на один байт оброблюваних даних

| Рівень стійкості (еквівалентна довжина ключа, біт) | Довжина вхідних даних, байт | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 128 | 256 | 512 | 1024 | 2048 | 4096 | 8192 | 16384 | 32768 | 65536 |
| 80 | 518 | 262 | 134 | 70 | 38 | 22 | 14 | 10 | 8 | 7 |
| 128 | – | – | 1158 | 582 | 294 | 150 | 78 | 42 | 24 | 15 |
| 256 | – | – | – | – | 7206 | 3606 | 1806 | 906 | 456 | 231 |

Так, наприклад, модулярне перетворення для рівня стійкості з еквівалентною довжиною ключа симетричного шифру в 128 біт необхідно реалізувати для довжини модуля не менше 3 072 біт, при вхідних даних у 256 байт (2 048 біт) формування геш-коду неможливо.

Аналіз даних табл. 2.6 підтверджує наведені вище міркування про зниження питомої складності перетворення (кількості циклів процесора на один байт вхідних даних) при збільшенні довжини оброблюваних інформаційних даних. Практично це означає, що із зростанням довжини блоків даних використана схема формування кодів контролю цілісності й автентичності щодо обчислювальної складності еквівалентна застосовуваним на сьогоднішній день у протоколах мережної безпеки (у тому числі в протоколах IPSec) алгоритмам MD-5 і SHA-1, а також алгоритмам SHA-2, CBC MAC-Rijndael і ін.

Для перевірки стійкості алгоритмів-претендентів гешування використовуємо набір таких тестів за певною методикою дослідження статистичних властивостей хеш-функцій.

Для проведення тестування були взяті наступні параметри: довжина тестованої послідовності n = 106біт;  тестовані послідовності m = 100; рівень значимості $\alpha$ = 0,01. Отже, об’єм тестованої вибірки становить: N = 106× 100 = 108 біт. Результати тестування алгоритмів хешування зведені в табл. 2.7

Таблиця 2.7 - Результати тестування алгоритмів гешування

| Генератор | Кількість тестів, у яких тестування пройшло більше 99% послідовностей | Кількість тестів, у яких тестування пройшло більше 96% послідовностей |
|---|---|---|
| 1 | 2 | 3 |
| BBS | 134 (71%) | 189 (100%) |
| FIPS 197 | 126 (67%) | 189 (100%) |
| Blake | 130 (69%) | 189 (100%) |
| CubeHash | 137 (73%) | 189 (100%) |
| ECHO | 139 (74%) | 189 (100%) |
| Grostl | 140 (75%) | 189 (100%) |
| Keccak | 134 (71%) | 187 (98,94%) |
| MASH-1 | 101 (53%) | 47 (24%) |
| MASH-2 | 126 (67%) | 189 (100%) |
| MASH(EC) | 141 (74%) | 189 (100%) |
| UMAC 32 | 167 (88%) | 189 (100%) |
| HMAC-SHA-256 | 134 (71%) | 187 (98%) |
| EMAC | 138 (73%) | 189 (100%) |
| RIPEMD-160 | 129 (68%) | 189 (100%) |
| UMAC+MASH-2 | 173 (91%) | 189 (100%) |

Отримані результати підтверджують теоретичні дослідження стійкості використаного каскадного методу хешування UMAC з використанням в останньому шарі в якості псевдоподложкі алгоритми модулярних перетворень MASH-1 і MASH-2.

Отримані результати теоретичних досліджень дозволяють обґрунтувати практичні рекомендації з використання моделей та методів каскадного оформлення кодів для контролю цілісності і автентичності даних для підвищення безпеки телекомунікаційних систем та мереж.

Як показано в розділі 1 для забезпечення цілісності й автентичності даних у телекомунікаційних мережах використовуються коди виявлення маніпуляцій (MDC), коди автентифікації повідомлень (MAC). У той же час використана схема забезпечує рівень безпеки доведеної стійкості й колізійні

властивості на рівні строго універсального гешування. Оскільки специфікаціями протоколів AH і ESP IPSec передбачене використання нових, більш ефективних алгоритмів формування ICV, для захисту пакетів даних у телекомунікаційних мережах пропонується використання моделей і методів каскадного формування кодів автентичності повідомлень, їх цінності та контролю їх захисту.

## 2.4 Зменшення моделі UMAC (mini-UMAC)

Для дослідження каскадних схем використовуємо колізійні властивості. Дані схеми формують автентичність і контроль цілісності даних. В основу даного дослідження лягло два підходи:

1. Здійснення оцінки розподілу колізійних кодів (образів).
2. Використання зменшених моделей для збереження структури початкового алгоритму.

Для першого підходу модель UMAC використовує різноманітні слої перетворення, і навіть блоковий симетричний шифр. Ця модель складається з шарів, структури яких масштабовані. На кожному з цих шарів необхідно здійснити дослідження колізійних характеристик, особливо тих, що створені завдяки шифру $Pad$ (псевдовипадкові підкладки). Це дослідження проводиться шляхом аналізу їхнього впливу код автентифікації нашої зменшеної моделі UMAC.

Для другого підходу потрібно здійснити дослідження головних ефективних показників. Цей підхід сьогодні широко використовується для дослідження криптографічних властивостей блокових симетричних шифрів. Так, наприклад, у роботах [8, 11] розроблені зменшені моделі криптоалгоритмів AES, Camelia, ADE, Лабіринт, Калина, Мухомор і ін., використання яких дозволило експериментально досліджувати диференціальні й лінійні властивості відповідних шифрів, оцінити їх стійкості атак диференціального й лінійного криптоаналізу. Крім того, на

основі аналізу зменшених моделей у працях [8, 11] реалізований підхід до оцінки ефективності блокових симетричних шифрів в розрізі обчислювальних витрат, необхідних для досягнення шифром асимптотичних характеристик випадкової підстановки.

У цій роботі пропонується подальший розвиток даного напрямку, що полягає у застосуванні зменшених моделей окремих шарів перетворень для оцінки колізійних властивостей формованих контролю цілісності й автентичності даних.

Отже, ми побачили, що схеми, які лягли в основу формування UMAC, поділяються на такі шари:

- $Y = Hash(K, M, Taglen)$ - це 3-х рівневе багатофункціональне гешування, завдяки якому ми отримуємо готовий геш-код;

- $Pad = PDF(K, Nonce, Taglen)$ - спеціальне перетворення, що формує псевдовипадкову підкладку, на основі блокового симетричного шифру;

- $Tag = UMAC(K, M, Nonce, Taglen) = Y \oplus Pad$ - остання перетворення, результатом якого сформований код автентифікації повідомлення.

Нижче будемо розглядати шари окремо, враховувати їхнє масштабування, проведемо моделювання використовуваних перетворень, використовуючи принцип масштабованості, що полягає в заміні об’єкта досліджень його зменшеною моделлю [8].

### 2.4.1. Зменшена модель трирівневого універсального гешування.

Зменшену версію 3-х рівневого хешування будуємо методом зменшення розмірів блоків необхідних даних , наприклад, у 8разів. Але це не змінить структуру перетворень. $Y_{min\,i}$ - величина коду моделі першого шару. Вона кратна 4бітам, і формується завдяки чотирьом послідовностям $Y_{min\,i_{L3_i}}$:

$$Y_{min\,i} = Y_{min\,i_{L3_1}} \| Y_{min\,i_{L3_2}} \| Y_{min\,i_{L3_3}} \| Y_{min\,i_{L3_4}},$$

Процес формування одного блоку $Y_{min\,i_{L3_i}}$ (другий рівень гешування у зменшеній моделі виконувати не будемо) відбувається так:

$$Y_{min\,i_{L3_i}} = Y_{min\,i_{L3}} =$$

$$Hash_{min\,i_{L3}}\left(K_{min\,i_{L3_1}}, K_{min\,i_{L3_2}}, Hash_{min\,i_{L1}}\left(K_{min\,i_{L1}}, M_{min\,i}\right)\right),$$

де $K_{min\,i_{L1}}$,$K_{min\,i_{L3_1}}$, $K_{min\,i_{L3_2}}$ – ключові послідовності міні-UMAC;

$Hash_{min\,i_{L1}}$ і $Hash_{min\,i_{L3}}$ – зменшені версії гешування першого й третього рівнів відповідно.

На першому рівні масив-рядок $M_{min\,i}$ – розмірності 32 біта перетворюється функцією $NH(K_{L1}, M_i)$. Ця дія показує результат гешування першого рівня: $Y_{min\,i_{L1}} = NH_{min\,i}\left(K_{min\,i_{L1}}, M_{min\,i}\right)$. Дані наступної функції$NH_{min\,i}\left(K_{min\,i_{L1}}, M_{min\,i}\right)$ знаходимо по такому принципу: $M_{min\,i}$ (блок інформації) розділяємо на 8 рівних під блоки, кожний з яких має по 4біта:

$$M_{min\,i} = M_{min\,i_1} \| M_{min\,i_2} \| \dots \| M_{min\,i_8}.$$

Тому і послідовність ключа $K_{L1}$представляється як ряд послідовності восьми під блоків:

$$K_{min\,i_{L1}} = K_{min\,i_{L1_1}} \| K_{min\,i_{L1_2}} \| \dots \| K_{min\,i_{L1_8}}.$$

Після чого (визначаючи початковий стан $Hash_{L1} = 0$) обчислюються:

$$Hash_{min\,i_{L1}} = Hash_{min\,i_{L1}} +_8 ((M_{min\,i_0} +_4 K_{min\,i_{L1_0}}) \times_8 (M_{min\,i_4} +_4 K_{min\,i_{L1_4}}));$$

$$Hash_{min\,i_{L1}} = Hash_{min\,i_{L1}} +_8 ((M_{min\,i_1} +_4 K_{min\,i_{L1_1}}) \times_8 (M_{min\,i_5} +_4 K_{min\,i_{L1_5}}));$$

$$Hash_{min\,i_{L1}} = Hash_{min\,i_{L1}} +_8 ((M_{min\,i_2} +_4 K_{min\,i_{L1_2}}) \times_8 (M_{min\,i_6} +_4 K_{min\,i_{L1_6}}));$$

$$Hash_{min\,i_{L1}} = Hash_{min\,i_{L1}} +_8 ((M_{min\,i_3} +_4 K_{min\,i_{L1_3}}) \times_8 (M_{min\,i_7} +_4 K_{min\,i_{L1_7}}));$$

де$+_8$, $+_4$ – операції підсумовування за модулями.

Ось і отримали завдяки цим обчисленням 8-мибітне значення $Y_{min\,i_{L1}} = Hash_{min\,i_{L1}}$.

На останньому рівні хешування вхідні дані $Y_{min\,i_{L1}}$ перетворяться у код $Y_{min\,i_{L3}}$ довжиною 4біта. Основні послідовності це є $K_{min\,i_{L3_1}}$та $K_{min\,i_{L3_2}}$, і їхні довжини будуть відповідно 16 і 4біта.

Загешовані значення $Hash_{min\,i_{L1}}$та основна послідовність $K_{min\,i_{L3_1}}$ одинаково діляться на 4 блоки. Кожний з цих представляє ціле число $Y_{min\,i_{L2_i}}$ й $K_{min\,i_{L3_{1_i}}}$, $i = 1,2,\ldots,4$. Геш-значення $Y_{min\,i_{L3}}$ обчислюється в такий спосіб:

$$Y_{min\,i_{L3}} = \left(\left(\left(\sum_{i=1}^{4} Y_{min\,i_{L2_i}} K_{min\,i_{L3_{1_i}}}\right) mod(17)\right) mod(2^4)\right) xor(K_{min\,i_{L3_2}}),$$

де $(x)xor(y)$ – операція “виключення або” для $x$ й $y$.

2.4.2. Зменшена модель блокового симетричного шифру AES.

Міні-версія блокового симетричного шифру AES для формування псевдовипадкової підкладки докладно розглянута в роботах [8, 12, 13]. Найбільш простою у реалізації є міні-версія шифру AES (Baby-Rijndael), яка запропонована К. Бергманом [8]. При розгляді зменшеної моделі шифру, ми також побачимо як формується псевдовипадкові підкладки.

Якщо 16-бітний відкритий текстовий блок, який буде позначений шістнацятковими значеннями $h_0$, $h_1$, $h_2$, $h_3$, то $h_0$ буде першим взідним потоком з 4 біт. Але це значення буде розглядатися як біт вищого рангу і представлений представлений $h_0 = 8$, $h_1 = c$, $h_2 = 7$, $h_3 = 1$.

Ключ, довжина якого також 16біт, буде позначений $k_0$, $k_1$, $k_2$, $k_3$ , а це також шістнадцяткові числа.

Початковий блок загружається як набір $h_0$, $h_1$, $h_2$, $h_3$, а такому вигляді

$$\begin{bmatrix} h_0 & h_2 \\ h_1 & h_3 \end{bmatrix}.$$

Дя прикладу, якщо вхідним блоком буде такий набір: 1000 1100 0111 0001, то загружені дані набудуть такого вигляду:

$$\begin{bmatrix} 8 & 7 \\ c & 1 \end{bmatrix}, \text{ де матриця } 8\times2 \text{ буде } \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

Baby-Rijndael включає дещо ідентичні за структурою. Спочатку вхідний блок загружається в початковий стан, як описано вище, а потім відбувається шифрування. В процесі ще обчислюються раундові ключі. Загальна структура шифрування така:

$$E(a) = r_4 \circ r_3 \circ r_2 \circ r_1 \circ (a \oplus k_0),$$

де a – позначає стан;

$k_0$, $k_1$, $k_2$, $k_3$ – раундові ключі й $r_i(a) = (t \cdot \tilde{\sigma}(S(a))) \oplus k_i$ за винятком $r_4$, тому що немає множення на t. Після завершення шифрування 16-бітний блок вивантажується у такій самій послідовності як і він загружався.

Наступним кроком буде розгляд окремих компонентів шифрування:

Sub-байт – це S – здійснюється заміна таблиці, що використовується для кожної 16-кової цифри стану:

$$\begin{bmatrix} h_0 & h_2 \\ h_1 & h_3 \end{bmatrix} \xrightarrow{S} \begin{bmatrix} S(h_0) & S(h_2) \\ S(h_1) & S(h_3) \end{bmatrix},$$

де функція s задається табл. 2.10.

Таблиця 2.10 - Таблиця заміни реалізовані S-блоком Baby-Rijndael

| *x* | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *S(x)* | a | 4 | 3 | b | 8 | e | 2 | c | 5 | 7 | 6 | f | 0 | 1 | 9 | d |

Shiftrows: Операція $\hat{\sigma}$ впливає на зміну входу у 2-гому рядку:

$$\begin{bmatrix} h_0 & h_2 \\ h_1 & h_3 \end{bmatrix} \xrightarrow{\tilde{\sigma}} \begin{bmatrix} h_0 & h_2 \\ h_3 & h_1 \end{bmatrix}.$$

Mixcolumns: Матриця t є наступною матрицею 8x8 бітів:

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

Для цього перетворення стан розглядається як 8×2 бітова матриця. Стан множиться зліва на *t*, використовуючи матричне множення за модулем 2: $a = t \times a$.

Keyschedule: на початку шифру й наприкінці кожного раунду стан побітно підсумовується за модулем 2 з раундовим ключем. Стовпці раундових ключів визначені рекурсивно в такий спосіб:

$$w_0 = \begin{pmatrix} k_0 \\ k_1 \end{pmatrix}, w_1 = \begin{pmatrix} k_2 \\ k_3 \end{pmatrix};$$

$$w_{2i} = w_{2i-2} \oplus S(reverse(w_{2i-2})) \oplus r_i;$$

$w_{2i+1} = w_{2i-1} \oplus w_{2i}$ для всіх $i = 1,\ 2,\ 3,\ 4,$

де $r_i = \binom{2^{i-1}}{0}$, а функція reverse замінює 2 входи.

2.4.2. Зменшена модель кінцевого перетворення.

Міні-версія кінцевої зміни зменшеної моделі UMAC заключається в формуванні звітності за другим модулем $Y_{min\,i}$ та $Pad_{min\,i}$:

$$Tag_{min\,i} = Y_{min\,i} \oplus Pad_{min\,i}.$$

Можемо зробити висновок, що коли масштабувати перетворення на конкретних,окремих шарах схеми створення автентичності даних та контролювання її цілісності, ми можемо отримати міні-версію UMAC, а також систематично здійснювати дослідження колізійних функцій кодів.

Процес масштабування даних під час створення зменшеної моделі UMAC показаний на рис.2.2.

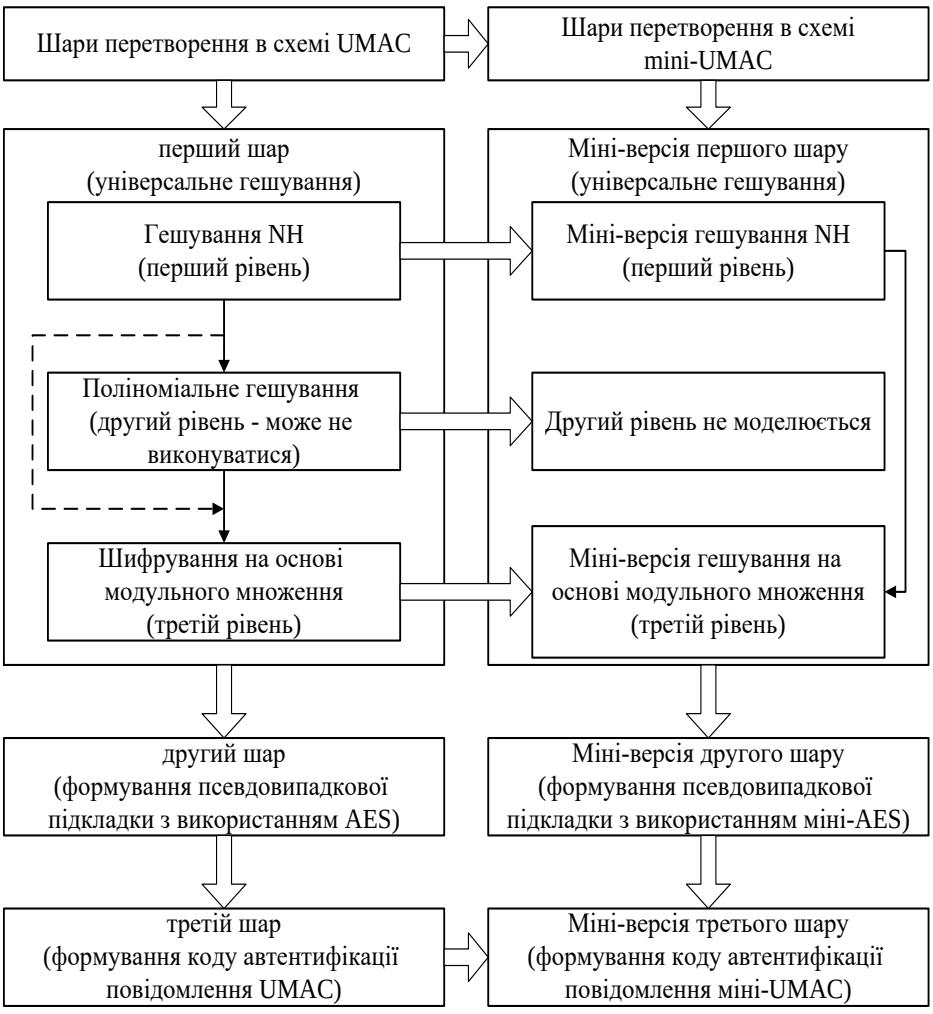



Рисунок 2.2 - Схема масштабування UMAC

Таким чином, для дослідження колізійних властивостей каскадних схем формування кодів контролю цілісності та автентичності даних у дипломній роботі пропонується використовувати принцип масштабованості, що полягає в заміні об’єкта досліджень його зменшеною моделлю. Використана зменшена модель (mini-UMAC) багатошарового (каскадного) оформлення коду автентичності даних і контролю цілісності, її програмна реалізація є конкретною реалізацією розглянутого підходу.

## 2.5. Розробка програмної реалізації міні-версії каскадного формування кодів контролю цілісності та автентичності даних (mini-umac)

Для реалізації розглянутого підходу до дослідження колізійних властивостей коду автентичності даних і контролю цілісності розроблена програмна реалізація міні-версії багатошарової каскадної конструкції (mini-UMAC). Реалізація виконана мовою високого рівня програмування C# з використанням компілятора Visual Studio 12. Її основна форма введення-виведення наведена на рис. 2.3.

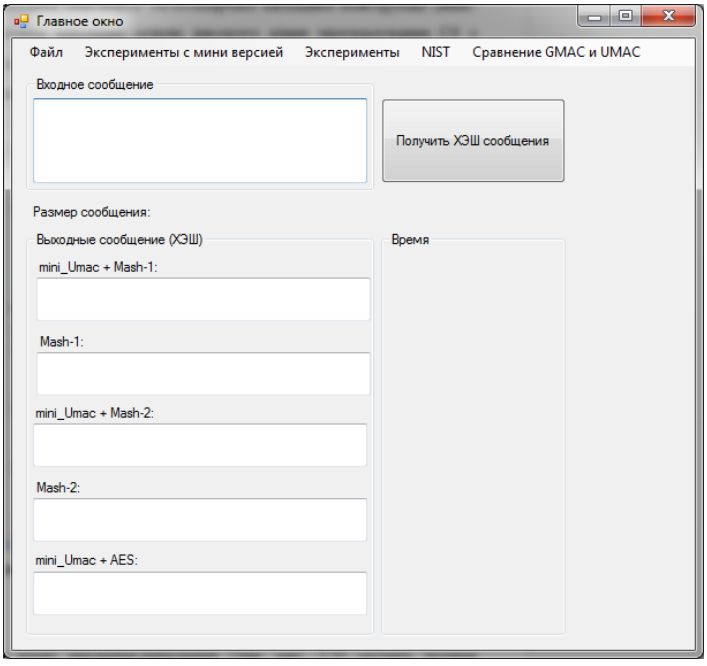



Рисунок 2.3 - Основна форма введення-виведення програмної реалізації mini-UMAC

Основне вікно введення-виведення (див. рис. 2.3) містить чотири основних поля. У верхній частині форми знаходиться поле «Вхідне повідомлення» для введення даних, що підлягають обробці.

У це поле користувач вносить повідомлення у вигляді набору ASCII-символів, після чого відповідні цифрові дані при натисканні кнопки «Одержати геш-повідомлення» обробляються розглянутою вище схемою формування кодів контролю цілісності та автентичності mini-UMAC.

Результати багатошарового каскадного гешування у вигляді коду автентичності даних і контролю цілісності виводиться в поля, розміщені в нижній частині основної форми введення-виведення (див. рис. 2.3). Це і є шукані коди, сформований за схемою mini-UMAC. Нижче поля вводу повідомлення і в правій частині вікна виводяться короткі параметри процесу обчислення коду контролю цілісності й автентичності даних, а саме: «Розмір вхідного повідомлення», у кількості ASCII символів, а також витрачений час на обробку введених даних, виміряний в мілісекундах. Приклад роботи цієї форми наведений на рис. 2.4.

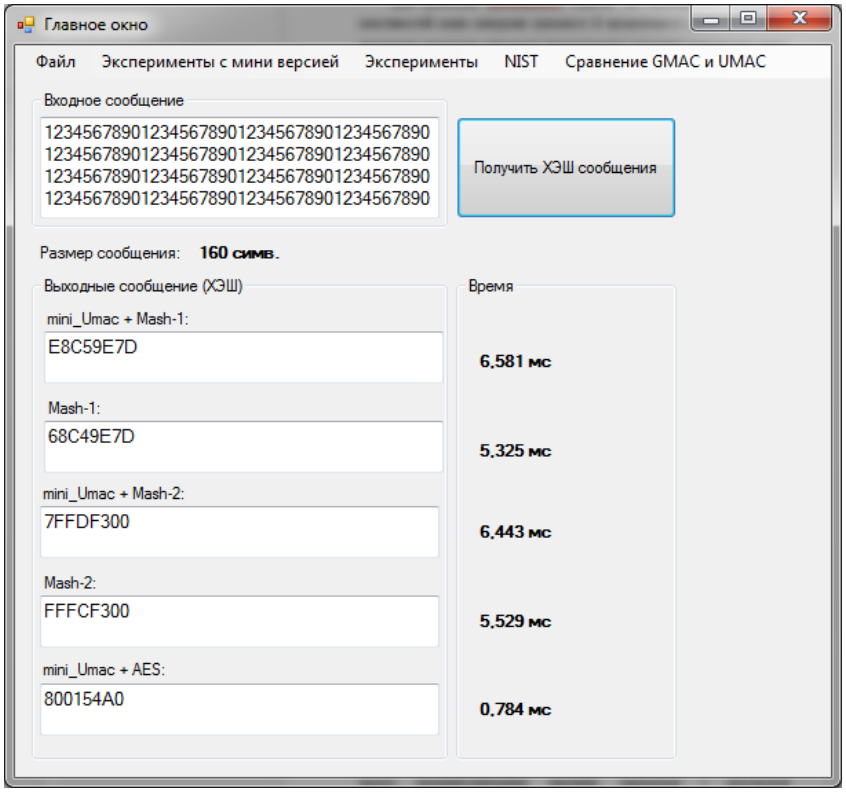
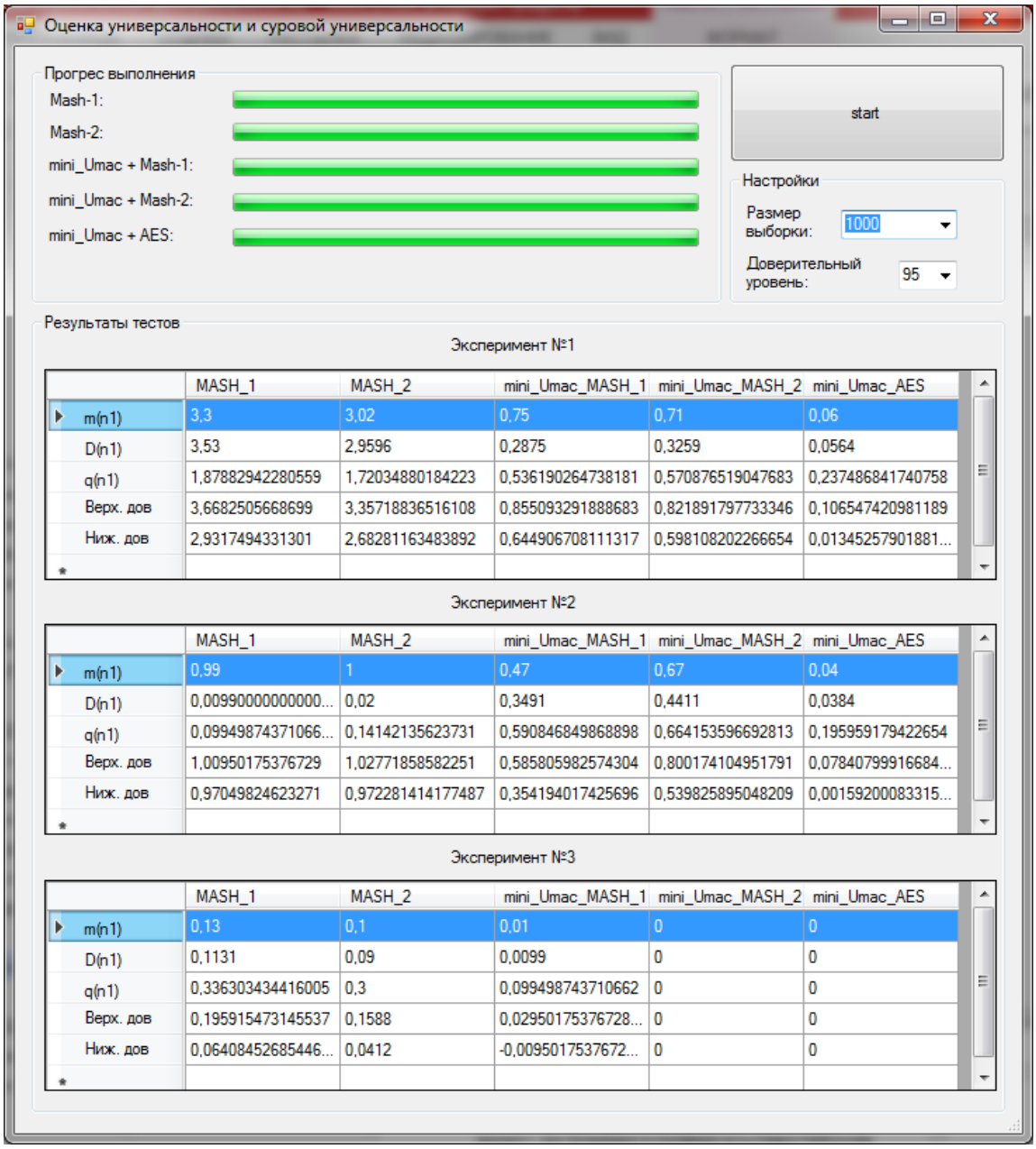



Рисунок - 2.4. Приклад роботи головної форми

Для дослідження колізійних властивостей формованих кодів контролю цілісності й автентичності в основній формі введення-виведення передбачено пункт меню «Експерименти». При його натисканні відкривається відповідна форма введення-виведення вихідних параметрів і результатів експериментальних досліджень (рис. 2.5).




| | MASH_1 | MASH_2 | mini_Umac_MASH_1 | mini_Umac_MASH_2 | mini_Umac_AES |
|---|---|---|---|---|---|
| m(n1) | 3,3 | 3,02 | 0,75 | 0,71 | 0,06 |
| D(n1) | 3,53 | 2,9596 | 0,2875 | 0,3259 | 0,0564 |
| q(n1) | 1,87882942280559 | 1,72034880184223 | 0,536190264738181 | 0,570876519047683 | 0,237486841740758 |
| Верх. дов | 3,6682505668699 | 3,35718836516108 | 0,855093291888683 | 0,821891797733346 | 0,106547420981189 |
| Ниж. дов | 2,9317494331301 | 2,68281163483892 | 0,644906708111317 | 0,598108202266654 | 0,01345257901881... |

Эксперимент №2

| | MASH_1 | MASH_2 | mini_Umac_MASH_1 | mini_Umac_MASH_2 | mini_Umac_AES |
|---|---|---|---|---|---|
| m(n1) | 0,99 | 1 | 0,47 | 0,67 | 0,04 |
| D(n1) | 0,00990000000000... | 0,02 | 0,3491 | 0,4411 | 0,0384 |
| q(n1) | 0,09949874371066... | 0,14142135623731 | 0,590846849868898 | 0,664153596692813 | 0,195959179422654 |
| Верх. дов | 1,00950175376729 | 1,02771858582251 | 0,585805982574304 | 0,800174104951791 | 0,07840799916684... |
| Ниж. дов | 0,97049824623271 | 0,972281414177487 | 0,354194017425696 | 0,539825895048209 | 0,00159200083315... |

Эксперимент №3

| | MASH_1 | MASH_2 | mini_Umac_MASH_1 | mini_Umac_MASH_2 | mini_Umac_AES |
|---|---|---|---|---|---|
| m(n1) | 0,13 | 0,1 | 0,01 | 0 | 0 |
| D(n1) | 0,1131 | 0,09 | 0,0099 | 0 | 0 |
| q(n1) | 0,336303434416005 | 0,3 | 0,099498743710662 | 0 | 0 |
| Верх. дов | 0,195915473145537 | 0,1588 | 0,02950175376728... | 0 | 0 |
| Ниж. дов | 0,06408452685446... | 0,0412 | -0,0095017537672... | 0 | 0 |

Рисунок 2.5 - Форма введення-виведення для експериментальних досліджень колізійних властивостей програмної реалізації mini-UMAC

Форма введення-виведення вихідних параметрів і результатів експериментальних досліджень (див. рис. 2.3) містить такі органи керування. У правій верхній частині форми знаходяться кнопка при натисканні якої приводяться в дію алгоритми дослідження колізійних властивостей геш функцій. Поле вибіру, яке знаходиться нижче кнопки запуску, дозволяє вибрати розмір досліджуємої вибірки.

Розроблена програмна реалізація дозволяє експериментально досліджувати три окремі властивості, умовно позначені першим, другим і третім пунктами меню.

## 2.6 Висновки за розділом 2

Для аналізу колізійних властивостей MAC-кодів використано підхід на основі зменшених моделей (міні-версій) алгоритмів формування UMAC, що дозволяє зберегти їх алгебраїчну структуру.

Розроблено програмний пакет який реалізує математичний апарат щодо методики статистичних досліджень аналізу колізійних властивостей, що дозволяють визначити розподіл кодів і отримувати оцінки колізійних властивостей з необхідною точністю.

Реалізовані модель і метод каскадного формування MAC-кодів з використанням на останньому етапі криптографічно сильної функції взаємозамінного хешування, яке базується нв основі модульного перетворення. Поставлені задачі вирішуються суворо універсального гешування, малу обчислювальну складність та високі показники безпеки.

# 3 ДОСЛІДЖЕННЯ ВЛАСТИВОСТЕЙ КОЛІЗІЇ КОДІВ КОНТРОЛЮ ЦІЛІСНОСТІ Й АВТЕНТИЧНОСТІ ДАНИХ І ОБҐРУНТУВАННЯ ПРОПОЗИЦІЙ З ЇХ УДОСКОНАЛЕННЯ

Проведений аналіз показав, що найбільшою обчислювальною ефективністю володіє відібраний при здійсненні європейського конкурсу NESSIE алгоритм UMAC (Message Authentification Code using Universal Hashing) [11], для формування кодів контролю цілісності й автентичності в якому використовуються сімейства універсальних функцій хешування [10**Error! Reference source not found.**]. Кількість колізій (зіткнень) формованих геш-образів для кожного введеного ключа універсального гешування не перевищує деякої заздалегідь заданої величини, а криптостійкість UMAC забезпечується на рівні обраного криптоалгоритму (за специфікацією рекомендований алгоритм шифрування AES). Однак вплив використовуваного криптоалгоритму на колізійні властивості кодів достовірності повідомлень UMAC на сьогодні мало досліджені, забезпечення властивостей універсального гешування в такій багатошаровій конструкції не обґрунтоване.

На основі отриманих результатів експериментальних досліджень розробляються пропозиції з удосконалення методів каскадного формування MAC і забезпечення високих колізійних властивостей.

## 3.1. Обґрунтування що до використання математичного апарату статистичного дослідження колізійних властивостей кодів контролю цілісності й автентичності даних

Ідея універсального гешування полягає у визначенні такого набору елементів кінцевої множини $H$ геш-функцій $h: A \to B, |A| = a,\ |B| = b$, щоб випадковий вибір функції $h \in H$ забезпечував би малу ймовірність колізії, тобто для будь-яких різних входів $x_1$ і $x_2$ ймовірність того, що $h(x_1) = h(x_2)$

(імовірність колізії, зіткнення), не може перевищувати деякої заздалегідь заданої величини $\varepsilon$:

$$P_{кол} = P(h(x_1) = h(x_2)) \leq \varepsilon,$$

причому ймовірність колізії може бути розрахована як

$$P_{кол} = \frac{\delta_H(x_1,x_2)}{|H|},$$

де $\delta_H(x_1, x_2)$ – кількість таких геш-функцій у $H$, при яких значення$x_1, x_2 \in A,\ x_1 \neq x_2$ викликають колізію, тобто $h(x_1) = h(x_2)$.

Наведемо два визначення універсального гешування [19, 21, 24].

Нехай $0 < \varepsilon < 1$. $H$ є $\varepsilon$ – універсальним геш-класом (скорочено $\varepsilon - U(H, A, B)$), якщо для двох різних елементів $x_1, x_2 \in A$ існує не більше $H \cdot \varepsilon$ функцій $f \in H$ таких, що$h(x_1) = h(x_2)$, якщо $\delta_H(x_1, x_2) \leq \varepsilon|H|$ для всіх$x_1, x_2 \in A,\ x_1 \neq x_2$.

Нехай $0 < \varepsilon < 1$. $H$ є $\varepsilon$ – строго універсальним геш-класом (скорочено $\varepsilon - SU(H, A, B)$), якщо виконуються такі умови:

- для кожного $x_1 \in A$ й для кожного$y_1 \in B$,

$$|\{h \in H: h(x_1) = y_1\}| = |H|/|B|;$$

- для кожного$x_1, x_2 \in A,\ x_1 \neq x_2$ і для кожного$y_1, y_2 \in B$,

$$|\{h \in H: h(x_1) = y_1, h(x_2) = y_2\}| \leq \varepsilon|H|.$$

Визначення універсального класу геш-функцій еквівалентне визначенню такого алгоритму формування MAC, при якому кількість різних правил формування коду контролю цілісності й автентичності даних (кількість ключів), при яких існує колізія для двох довільних вхідних послідовностей, обмежена. Кількість таких ключів не може перевищувати значення$P_{кол} \cdot |H|$, де $P_{кол}$ – імовірність колізії, $|H|$ – кількість усіх правил (ключів).

Визначення строго універсального класу геш-функцій еквівалентне визначенню такого алгоритму формування кодів контролю цілісності та автентичності даних, при якому будуть виконуватися такі правила:

Кількість правил формування MAC (кількість ключів), при яких для довільної вхідної послідовності значення коду контролю цілісності й автентичності даних не змінюється, обмежене. Кількість таких ключів не може перевищувати значення$|H|/|B|$, де $|H|$ – кількість усіх ключів; $|B|$ – кількість можливих станів MAC.

Кількість правил формування коду контролю цілісності та автентичності даних (кількість ключів), при яких для двох довільних вхідних послідовностей відповідні їм значення MAC не змінюються, обмежене. Кількість таких ключів не може перевищувати значення$P_{\text{кол}}|H|$, де $\text{P}_{\text{кол}}$ – імовірність колізії; $|H|$ – кількість усіх ключів. Імовірність колізії кодів контролю цілісності та автентичності даних у схемі зі строго універсальним гешуванням визначається як $P_{\text{кол}} \leq \varepsilon$.

В основі використаної методики статистичного дослідження колізійних властивостей формованих елементів $h(x)$ лежить емпірична оцінка максимумів кількісті ключів (правил гешування), при яких:

Для довільних$x_1, x_2 \in A,\ x_1 \neq x_2$ виконується рівність:

$$h(x_1) = h(x_2). \tag{3.1}$$

Для довільних $x_1 \in A$ і $y_1 \in B$ виконується рівність:

$$h(x_1) = y_1. \tag{3.2}$$

Для довільних$x_1, x_2 \in A,\ x_1 \neq x_2$ і $y_1, y_2 \in B$ виконуються рівності:

$$h(x_1) = y_1, h(x_2) = y_2. \tag{3.3}$$

Оцінка за першим критерієм відповідає перевірці виконання умови для універсального класу геш-функцій, оцінка з другого та третього критеріїв – умовам для строго універсального класу геш-функцій.

Введемо такі позначення:

$$n_1(x_1, x_2) = |\{h \in H: h(x_1) = h(x_2)\}|, x_1, x_2 \in A,\ x_1 \neq x_2;$$

$$n_2(x_1, y_1) = |\{h \in H: h(x_1) = y_1\}|, x_1 \in A,\ y_1 \in B;$$

$$n_3(x_1, x_2, y_1, y_2) = |\{h \in H: h(x_1) = y_1, h(x_2) = y_2\}|, x_1, x_2 \in A, x_1 \neq x_2,$$
$$y_1, y_2 \in B.$$

Перший показник $n_1(x_1, x_2)$ характеризує кількість правил гешування (правил формування MAC), при яких для заданих $x_1, x_2 \in A,\ \ x_1 \neq x_2$ виконується рівність (3.1), тобто кількість ключів, при яких існує колізія для двох вхідних послідовностей $x_1$ і $x_2$.

Другий показник $n_2(x_1, y_1)$ характеризує кількість правил гешування (за якими формуються MAC) таких, що для даних$x_1 \in A$, $y_1 \in B$ виконується рівність (3.2), тобто виконується наступне правило: кількість ключів, при яких для вхідної послідовності $x_1$ MAC -коду $y_1$ залишається не змінною.

Третій показник $n_3(x_1, x_2, y_1, y_2)$ характеризує кількість правил формування MAC таких, що для даних $x_1, x_2 \in A, x_1 \neq x_2,\ \ y_1, y_2 \in B$ виконується рівність (3.3). Це означає, що кількість ключів, за умови, що для двох вхідних послідовностей $x_1$ і $x_2$ відповідні їм значення MAC -кодів рівні $y_1$ і $y_2$ залишається не змінною. Оскільки кількість ключів, при яких можуть виконуватися рівності (3.1), (3.2) і (3.3), не повинні бути більшими за $P_{кол} \cdot |H|$, $|H|/|B|$ і $P_{кол} \cdot |H|$ нас буде цікавити максимальна кількість таких ключів для кожного з сформованих набору елементів. Розглянемо статистичні характеристики максимумів цих величин, а потім проведемо порівняння отриманих результатів із кількістю $P_{кол} \cdot |H|$ (для першого критерію), із кількістю $|H|/|B|$ (для другого критерію) і кількістю $P_{кол} \cdot |H|$ (для третього критерію).

Таким чином, в якості статистичних показників оцінок колізійних властивостей, за якими будемо проводити експериментальні дослідження, будемо використовувати:

- математичні сподівання $m(n_1)$, $m(n_2)$ і $m(n_3)$ максимумів кількості правил гешування (правил формування МАС), при яких виконуються рівності (3.1), (3.2) і (3.3) відповідно;
- дисперсії$D(n_1)$, $D(n_2)$ і$D(n_3)$, що характеризують розсіювання значень відносно математичних сподівань (кількості правил МАС-кодів), при яких виконуються рівності (3.1), (3.2) і (3.3), щодо їх математичних сподівань $m(n_1)$, $m(n_2)$ і$m(n_3)$ відповідно.

Оцінку колізійних властивостей за наведеними критеріями будемо робити в середньостатистичному розумінні. Тобто при постановці експерименту слід використовувати обмежений набір елементів $x_1, x_2 \in A$, що $x_1 \neq x_2$ й відповідають геш-образам (МАС)$y_1, y_2 \in B$, розцінюючи відповідні результати як статистичну вибірку. Незсуненою та ефективною оцінкою для математичного сподівання m випадкової величини X є середнє арифметичне елементів її вибірки $X_i$ (або статистичне середнє) [7]:

$$\widetilde{m} = \frac{1}{N}\sum_{i=1}^{N} X_i, \tag{3.4}$$

де $N$ – кількість реалізацій випадкової величини X. Оцінка дисперсії випадкової величини X визначається за формулою:

$$\widetilde{D} = \frac{1}{N-1}\sum_{i=1}^{N}(X_i - \widetilde{m})^2. \tag{3.5}$$

Через центральну граничну теорему теорії ймовірностей при великих значеннях кількості реалізацій N середнє арифметичне буде мати розподіл, близький до нормального [7] з математичним сподіванням

$$m[\tilde{m}] \approx \tilde{m}. \quad (3.6)$$

і середнім квадратичним відхиленням

$$\sigma[\tilde{m}] \approx \frac{\sigma}{\sqrt{N}}, \quad (3.7)$$

де $\sigma$ – середнє квадратичне відхилення оцінюваного параметра.

Довірча ймовірність - ймовірність того, що оцінка $\tilde{m}$ відхилиться від свого математичного сподівання менше ніж на $\varepsilon$, визначається виразом [7]

$$P(|\tilde{m} - m| < \varepsilon) \approx 2\Phi\left(\frac{\varepsilon}{\sigma[\tilde{m}]}\right), \quad (3.8)$$

де $\Phi(x)$ – функція Лапласа:

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_0^x e^{-\frac{t^2}{2}} dt. \quad (3.9)$$

З цього випливає, що після дослідження колізійних властивостей, будемо використовувати метод математичної статистики або статистичну повірку гіпотези. Генеруючи випадкові величини *X,* формуємо вибірку в такий спосіб:

– для середньостатистичної оцінки математичного сподівання $m(n_1)$ й дисперсії $D(n_1)$, де $n_1(x_1, x_2)$ - випадкова величина, при якому виконується рівність $h(x_1) = h(x_2)$, отже, вибірку *N:* $X_1$, $X_2$, ..., $X_N$ сформуємо відбором *N* множин, у кожній з яких міститься *M* пар елементів $x_1, x_2 \in A$, $x_1 \neq x_2$ і оцінюється $n_1(x_1, x_2)$, тобто загальний обсяг формованих пар елементів$x_1, x_2 \in A$, $x_1 \neq x_2$ складе *NM*;

– для середньостатистичної оцінки $m(n_2)$ й $D(n_2)$ як випадкова величина виступає максимум $n_2(x_1, y_1)$, при якому виконується рівність$y_1 =$

$h(x_1)$, отже, вибірку *N:* $X_1$, $X_2$, ..., $X_N$ сформуємо відбором *N* множин, у кожній з яких міститься $M$ пар елементів $x_1 \in A,\ y_1 \in B$ і оцінюється $n_2(x_1, y_1)$. Загальний обсяг формованих пар елементів $x_1 \in A$, $y_1 \in B$ складе *NM*;

– для середньостатистичної оцінки $m(n_3)$ й $D(n_3)$ як випадкова величина виступає максимум $n_3(x_1, x_2, y_1, y_2)$, при якому виконуються рівності $y_1 = h(x_1)$ й$y_2 = h(x_2)$, отже, вибірку *N:* $X_1$, $X_2$, ..., $X_N$ сформуємо відбором $N$ множин, у кожній з яких міститься *M* четвірок елементів $x_1, x_2 \in A, x_1 \neq x_2,\ \ y_1, y_2 \in B$ і оцінюється $n_3(x_1, x_2, y_1, y_2)$, загальний обсяг формованих четвірок складе *NM*.

Під час проведення експерименту з колізійними властивостями хешування оцінюємо значення $\widetilde{m}(n_i)$, яке є середнів арифметичним. Зафіксуємо точність $\varepsilon$ та обчислимо значення функції, яка носить назву Лапласа:

$$P(|\widetilde{m}(n_i) - m(n_i)| < \varepsilon) \approx 2\Phi\left(\frac{\varepsilon}{\sigma[\widetilde{m}(n_i)]}\right);$$

$$\sigma[\widetilde{m}(n_i)] \approx \frac{\sqrt{\widetilde{D}(n_i)}}{\sqrt{N}},\ i = 1,2,3.$$

При постановці оберненого значення:

$$\widetilde{m}(n_i) - t_\rho \cdot \sigma[\widetilde{m}(n_i)] < m(n_i) < \widetilde{m}(n_i) + t_\rho \cdot \sigma[\widetilde{m}(n_i)], i = 1,2,3, \qquad (3.10)$$

де $t_\rho$ – корінь рівняння $2\Phi(t_\rho) = P_\partial$.

Таким чином дана методика, використовуючи зменшені моделі окремих шарів перетворень MAC, на основі оцінки розподілу зіткнень формованих образів дозволяє експериментально досліджувати колізійні властивості кодів автентичності іконтролю цілісності даних

## 3.2. Результати експериментальних досліджень колізійних властивостей кодів автентичності і контролю цілісності даних

З використанням розробленої зменшеної моделі UMAC (mini-UMAC) і методики статистичного дослідження колізійних властивостей кодів

контролю цілісності та автентичності даних проведемо експериментальну оцінку розподілу кількісті зіткнень (колізій) формованих образів.

Оскільки в розглянутій вище схемі mini-UMAC на першому шарі (при формуванні геш-коду $Y_{min\,i}$) використовуються сімейства універсальних функцій гешування, докладно вивчені в роботах, статистичні дослідження проведемо тільки: на другому шарі формування псевдовипадкової підкладки $Pad_{min\,i}$ і на кінцевому етапі формування кодів автентифікації (після виконання підсумовування $Tag_{min\,i} = Y_{min\,i} \oplus Pad_{min\,i}$). Саме на цих етапах, на погляд й порушуються властивості універсальності формованих кодів автентифікації.

При здійснені статистичних досліджень колізійних властивостей формованих значень $Pad_{min\,i}$ і $Tag_{min\,i}$ для кожного експерименту оцінювалися математичні сподівання $m(n_1)$, $m(n_2)$ і $m(n_3)$, дисперсії $D(n_1)$, $D(n_2)$ і $D(n_3)$, а також для фіксованої точності $\varepsilon$ розраховувалися відповідні довірчі ймовірності $P(|\widetilde{m}(n_i) - m(n_i)| < \varepsilon)$, $i = 1,\ 2,\ 3$.

Дослідження проводилися над вибіркою обсягу N = 100 для формування кожного елемента вибірки розраховувався максимум за множиною з M = 1000 кортежів елементів.

Таким чином, загальний обсяг формованих наборів склав NM = $10^5$. Отримані результати експериментальних досліджень зведено в табл. 3.1

При дослідженні колізійних властивостей кодів автентифікації, сформованих з використанням міні-версії шифру AES, кількість ключів, для яких виконується рівність $h(x_1) = h(x_2)$, при всіх випробуваннях дорівнювало нулю, тобто $n_1(x_1, x_2) = 0$ у всіх $N = 100$ дослідах. Цей результат пояснюється такою властивістю. Шифр AES (як і його міні-версія), реалізує бієктивне відображення множини відкритих текстів у множину шифрограм, тобто для фіксованого ключа формовані шифртексти, що відповідають різним відкритим текстам, будуть різні.

Таблиця 3.1 - Результати експериментальних досліджень колізійних властивостей коду автентичності іконтролю цілісності даних, які сформовані з використанням mini-AES і mini-UMAC

| | mini-AES, $Pad_{min\,i}$ | mini-UMAC, $Tag_{min\,i}$ |
|---|---|---|
| $\widetilde{m}(n_1)$ | – | 4,23 |
| $\widetilde{D}(n_1)$ | – | 0,18 |
| $P_\partial = P(\lvert\widetilde{m}(n_1) - m(n_1)\rvert < 0{,}1)$ | – | 0,98 |
| $\widetilde{m}(n_2)$ | 6,68 | 4,78 |
| $\widetilde{D}(n_2)$ | 0,42 | 0,42 |
| $P_\partial = P(\lvert\widetilde{m}(n_2) - m(n_2)\rvert < 0{,}15)$ | 0,98 | 0,98 |
| $\widetilde{m}(n_3)$ | 0,19 | 5,31 |
| $\widetilde{D}(n_3)$ | 0,15 | 0,24 |
| $P_\partial = P(\lvert\widetilde{m}(n_3) - m(n_3)\rvert < 0{,}1)$ | 0,99 | 0,96 |

Проведені експериментальні дослідження з першого введеного критерію саме й полягали в підрахунку кількості ключів, при яких спостерігається зіткнення (колізія, збіг) двох шифртекстів, що відповідають двом різним відкритим текстам, що є неможливим за визначенням бієктивного шифру. У зв'язку із цим статистичні дані за першим критерієм для міні-версії шифру AES у табл. 3.1 не наведені як неінформативні.

Проаналізуємо отримані результати статистичних досліджень колізійних властивостей кодів контролю цілісності й автентичності, зрівняємо отримані результати середньостатистичних оцінок математичних сподівань $m(n_1)$, $m(n_2)$ і $m(n_3)$ кількості правил гешування (правил формування MAC), при яких виконуються рівності (3.1), (3.2) і (3.3) відповідно, з теоретичним оцінками: числом $\mathrm{P}_{\text{кол}} \cdot |H|$ (для першого критерію), із числом $|H|/|B|$ (для другого критерію) і числом $P_{кол} \cdot |H|$ (для третього критерію).

Розглянемо перший критерій, за яким оцінюється кількість правил гешування (правил формування MAC), при яких існує колізія (збіг MAC) для двох довільних вхідних послідовностей. Відповідно до теоретичних оцінок ця величина обмежена зверху числом $P_{кол} \cdot |H|$. Конкретизуємо цю (теоретичну)

оцінку для кодів автентифікації, сформованих з використанням mini-AES і mini-UMAC.

Потужність ключової множини для mini-AES і mini-UMAC становить$|H| = 2^{16}$, потужність множини формованих кодів автентифікації – $|B| = 2^{16}$. Якщо використовувати верхню оцінку ймовірності колізій як обернену величину потужності формованих кодів автентифікації $\mathrm{P}_{\text{кол}} = 2^{-16}$, одержимо $n_1(x_1, x_2) \leq P_{кол} \cdot |H| = 1$.

Для міні-версії шифру AES ця умова виконується (обґрунтовується біективністю шифрувального перетворення), однак колізійні властивості mini-UMAC суттєво поступаються цій верхній теоретичній оцінці.

Фактично, кількість колізій перевищує теоретичну границю більш ніж у чотири рази й це положення підтверджене з високою довірчою ймовірністю:

$$P_{\partial} = P(|\tilde{m}(n_1) - m(n_1)| < 0{,}1) > 0{,}98.$$

Розглянемо другий критерій, за яким оцінюється кількість правил гешування (правил формування MAC), при яких для довільної вхідної послідовності значення коду не міняється. Відповідно до теоретичних оцінок ця величина для кодів автентифікації, сформованих з використанням mini-AES і mini-UMAC, обмежена зверху числом:

$$n_2(x_1, y_1) \leq |H|/|B| = 1.$$

Отримані експериментальні результати свідчать, що колізійні властивості кодів автентифікації, сформованих з використанням mini-AES і mini-UMAC, не задовольняють другому критерію, у кілька разів перевищує теоретичну оцінку для універсального гешування. Для підтвердження вірогідності отриманого результату в табл. 3.1 за другим критерієм наведене значення довірчої ймовірності (для величини довірчого інтервалу $\varepsilon = 0{,}15$):

$$P_{\partial} = P(|\tilde{m}(n_2) - m(n_2)| < 0{,}15) = 0{,}98.$$

Відповідно до *третього критерію* оцінюється кількість правил гешування (правил формування MAC), але MAC не змінюються. Теоретична оцінка цієї величини для універсального гешування обмежена зверху

числом$P_{\text{кол}}|H|$, що при використанні верхньої оцінки ймовірності колізій $P_{кол} = 2^{-16}$ дає:

$$n_3(x_1, x_2, y_1, y_2) \leq P_{кол} \cdot |H| = 1.$$

Значення, наведені у табл. 3.1, свідчать про те, що колізійні властивості кодів автентифікації, сформованих з використанням mini-AES, задовольняють третьому критерію. У той час кількість ключів mini-UMAC, більш ніж у п’ять разів перевищує верхню теоретичну оцінку.

Аналіз даних табл. 3.1 свідчить про адекватність отриманих результатів і відповідність їх статистичним властивостям усієї генеральної сукупності даних. Для фіксованої точності $\varepsilon$ отримані високі значення довірчої ймовірності, що говорить про обґрунтованість і вірогідність отриманих експериментальних результатів.

Таким чином, з отриманих результатів статистичних досліджень колімаційних властивостей коду автентичності і контролю цілісності даних, сформованих з використанням mini-AES і mini-UMAC, можна зробити такі важливі в прикладному відношенні висновки:

- криптографічний шар для отримання коду автентичності іконтролю цілісності повідомлень (mini-AES) задовольняє властивостям універсального гешування, імовірність колізії формованих геш-образів не перевищує наперед заданої величини (перший критерій). Це пояснюється, насамперед, тим, що шифрування випадкового (унікального) для всіх інформаційних повідомлень значення $Nonce$ приводить до формування множини випадкових (унікальних) для всіх інформаційних повідомлень псевдовипадкових підкладок $Pad$. Отже, формування псевдовипадкових підкладок $Pad$ здійснюється в результаті бієктивного відображення множини випадкових (унікальних) для всіх інформаційних повідомлень значень $Nonce$, у результаті чого колізії (зіткнення) підкладок $Pad$ відсутні за визначеним. У той же час, даний шар перетворень не задовольняє властивостям строго універсального гешування (не виконується другий критерій) (див. табл. 3.1). Крім того, забезпечення властивостей універсального гешування на цьому

шарі припускає формування й передачу випадкового для кожного повідомлення значення $Nonce$, що вимагає додаткових часових і програмно-апаратних витрат;

- результат формування кодів контролю цілісності й автентичності за схемою mini-UMAC не задовольняє властивостям як універсального гешування, так і властивостями строго універсального гешування. Це пояснюється тим, що схема із простим підсумовуванням за модулем два (XOR) двох результатів універсального гешування не завжди зберігає особливості хешування.

## 3.3 Висновки за розділом 3

Для дослідження колізійних властивостей коду автентичності і контролю цілісності даних використано ймовірнісний математичний апарат, який дозволяє в середньостатистичному розумінні одержувати оцінки кількості колізій із заданими довірчим інтервалом і необхідною точністю. Зокрема, уведені статистичні показники, що характеризують колізійні властивості коду автентичності іконтролю цілісності даних, а саме:

– математичне сподівання й дисперсія максимумів чи кількісті ключів, при яких існує колізія для двох довільних вхідних послідовностей;

– математичне сподівання й дисперсія максимумів кількості ключів, для таких вхідних послідовностей, що значення геш-коду не змінюється;

– математичне сподівання й дисперсія максимумів кількості ключів, при яких для двох заданих вхідних послідовностей відповідні їм значення геш-кодів не змінюються.

Уведені статистичні показники оцінки колізійних властивостей характеризують виконання умов універсального і строго універсального гешування, тобто визначають рівномірність розподілу кодів контролю цілісності й автентичності за всією множиною ключових даних.

При постановці експериментів використовувався обмежений набір вхідних елементів і відповідних їм геш-образів (MAC). Отримані результати слід розглядати як вибірку з генеральної сукупності. Таким чином, математичний апарат, використовуючи зменшені моделі окремих шарів перетворень, на основі оцінки розподілу зіткнень формованих образів дозволяє експериментально досліджувати колізійні властивості кодів контролю цілісності й автентичності даних.

У результаті проведених експериментальних досліджень колізійних властивостей каскадного формування кодів автентичності і контролю цілісності даних за схемою mini-UMAC встановлено:

- Перший і другий шари наших кодів задовольняють властивостям універсального гешування, ймовірність колізії формованих геш-образів (MAC) не перевищує наперед заданої величини. У той же час використовувані функції відображення не є строго універсальними геш-функціями, оскільки не виконуються обмеження кількості правил хешування.
- Третій шар кодів повідомлень не задовольняє властивостям універсального гешування, колізійні властивості схеми автентифікації повідомлень знижуються і не відповідають поставленим вимогам.

Таким чином, проведені дослідження дозволили встановити, що каскадна схема, яка використовується для строгого формування коду автентичності та цілісності даних, їх контролю. Ці дії здійснюються завдяки використанню універсальних функцій гешування та алгоритму блокового симетричного шифрування (схема UMAC) не задовольняє вимогам універсального гешування, її колізійні властивості знижені в результаті застосування криптографічного перетворення на останньому етапі формування автентифікаторів.

Перспективним напрямком досліджень є розробка й теоретичне обґрунтування нових схем ключового гешування, що використовують багатошарову конструкцію, що і дозволяють високі показники безпеки.

# 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

## 4.1 Ергономічні аспекти безпеки життєдіяльності

Важливою проблемою ергономіки є сумісність людини з машинами, механізмами. Тому основне завдання ергономіки – вивчення зв'язків між елементами системи ЛМС, розробка методів сумісності основного її компонента – людини з іншими середовищами та машинами, технікою.

Завдання ергономіки:

– розробка основ проектування діяльності людини-оператора з врахуванням специфіки експлуатації технічних систем та факторів навколишнього середовища;

– вивчення закономірностей взаємодії людини з технічними системами та навколишнім середовищем;

– формування принципів побудови системи ЛМС та алгоритмів дії у них людини-оператора;

– розробка перспективних форм праці людини і пов'язаних з нею технічних систем, факторів навколишнього середовища;

– розробка методів дослідження, проектування та експлуатації системи ЛМС, які забезпечують безпеку людини, ефективність праці.

Основним об'єктом ергономіки є система ЛМС. Проблемами взаємодії людини та машини займається також інженерна психологія, яка вивчає закономірності процесів інформаційної взаємодії людини у системі ЛМС.

У системі ЛМС завжди є 3 елементи: предмет праці, засоби праці та суб'єкт праці. Найменшою цільною одиницею, де наявні вказані елементи, є місце праці.

Місце праці – це зона, де відбувається трудова діяльність людини. Місце праці обладнане засобами відображення інформації, органами керування та допоміжним обладнанням.

Організацією місця праці називається проведення системи заходів щодо його обладнання засобами та предметами праці і їх розташуванням у визначеному порядку з метою досягнення:

– оптимізації умов трудової діяльності;

– безпеки праці;

– максимальної ефективності;

– комфортності роботи людини.

До робочого місця ставляться такі вимоги:

– достатній робочий простір, який дає змогу працюючій людині здійснювати необхідні рухи та переміщення;

– достатні фізичні, зорові та слухові зв'язки між людиною та обладнанням, а також між людьми під час виконання спільного трудового завдання;

– необхідний рівень освітлення;

– наявність необхідних засобів захисту;

– оптимальне розташування робочих місць, а також безпечні та достатні проходи для працюючих людей.

При організації робочого місця враховують основні антропометричні дані людини. Найважливішою характеристикою робочого місця є зона досягнення моторного поля.

Моторне поле – це простір робочого місця, в якому розміщені органи керування та інші технічні засоби, в якому людина здійснює рухові дії для виконання робочого завдання.

Ергономіка виробила конкретні вимоги до антропометричних показників обладнання.

Характеристика пульта:

– загальна висота: "сидячи" – 1650мм, "стоячи" – не більше ніж 1300 мм;

– висота розміщення органів керування для положення "сидячи" 530 - 1040 мм, стоячи - 1000 - 1500 мм.

Характеристики крісла:

– форма сидіння-квадратна;
– форма спинки - прямокутна вгнута;
– розмір сидіння - 400х400 мм, спинки - 300х120 мм;
– кут нахилу сидіння назад - 50 - - 60°;
– кут нахилу спинки - 50 - - 100°;

Розміри вільного місця для ніг:

– висота -не менше 600 мм;
– ширина - не менше 500 мм;
– глибина - не менше 400 мм.

Досягнення органів керування по горизонталі – півколо радіусом 600 мм. Встановлені також відстань між органами керування, їх розміри, зусилля переміщення, величина переміщення, напрямок переміщення.

Для операторів, які працюють з екранами дисплеїв та інших індикаторів, можуть бути рекомендовані такі режими праці та відпочинку.

Тривалість безперервної праці не повинна перевищувати 4-6 год. В іншому випадку працездатність через втому зору раптово знижується. Наприклад, оператор, який стежить за екраном індикатора, найуважніше працює протягом перших 30 хв чергування. А далі, внаслідок втоми зорового аналізатора, кількість помилок зростає май же в два рази та залишається незмінною до кінця другої години. Потім спостерігається нове зростання кількості помилок через загальну втому оператора. Тому для підтримки високої ефективності праці може бути рекомендований 30-хвилинний період чергування з наступною 30-хвилинною перервою.

Отже основним завданням ергономіки - забезпечення ефективної взаємодії людини і техніки, щоб перейти від техніки безпеки до безпечної техніки, яку ми використовуємо як у виробничій, так і побутовій сферах. Це один з основних напрямків ергономіки.

## 4.2 Психологічні чинники небезпеки

Виділяють комплекс чинників, що збільшують індивідуальну схильність людини до небезпеки. Це особливості темпераменту, функціональні зміни в організмі, дефекти органів відчуття, незадоволення даним видом діяльності.

Несприятливий характер діяльності (значні фізичні та розумові зусилля, незручна робоча поза, високий темп праці, нервово-емоційні перевантаження, перенапруга слухових та зорових аналізаторів, несумісність робочого місця, засобів праці, антропометричних даних людини) призводять до підвищеної фізичної та нервової втоми, яка послаблює психіку, знижує швидкість та точність орієнтації, притупляє пильність та увагу, порушує сприйняття.

Афектні стани (афект — вибух емоцій) можуть виникнути внаслідок виробничих невдач, під впливом образи. У стані афекту у людини розвивається емоційне звуження обсягу свідомості. Можуть спостерігатися різкі рухи, агресивні та руйнівні дії.

Вживання легких стимуляторів допомагає у боротьбі з сонливістю і може сприяти підвищенню працездатності на короткий період. Вживання ж активних стимуляторів на відповідальних роботах здатне викликати негативний ефект — погіршується самопочуття, зменшується швидкість реакції. Використання транквілізаторів, які діють заспокійливо та запобігають розвитку неврозів, може знижувати психічну активність, уповільнювати реакцію, викликати апатію та сонливість.

Чинники, що тимчасово підвищують індивідуальну імовірність наразитись на небезпеку.

Недосвідченість ж є одним із найважливіших факторів при безпеці роботи. Практичний досвід є безумовно важливим чинником, що підвищує безпеку праці. Він, до того ж, впливає на загальну поведінку працівника на

робочому місці, що проявляється у високому темпі, ритмі, інтенсивності роботи.

Необережність – це чинник, який підвищує імовірність наразити на небезпеку в певний момент часу не лише самого працівника, а й цілий виробничий колектив.

Втома з точки зору безпеки життєдіяльності є досить значним чинником. Як правило розрізняють фізіологічну та психічну втому.

Психічна втома виявляється такими явищами:

– зниженням сприйняття подразників, в результаті чого окремі подразники людина взагалі не сприймає, а інші сприймає лише з певним запізненням;

– зниження здатності концентрувати увагу;

– сповільненням мислення, яке, окрім того, певною мірою втрачає критичність, гнучкість, широту;

Таким чином, психічні стани, що виникають внаслідок раптових емоційних впливів, характеру діяльності, психічної втоми підвищують індивідуальну імовірність наразитись на небезпеку: з одного боку людина стає тимчасово необережною через відповідний психічний стан, а з іншого – втрачає пильність і впевненість в рухах.

## 4.3 Висновок за розділом 4

В четвертому розділі кваліфікаційної роботи висвітлено питання БЖД та ОП із застосуванням для практичної роботи.

В першому пункті описано значення ергономічних проблем в трудовій діяльності людини. Вжито заходи пов’язані із зручністю та збереженням концентрації уваги користувача.

Другий пункт досліджує дію психологічних чинників на організм людини та методи зменшення небезпек. При виконанні кваліфікаційної роботи було оцінено дію негативних психологічних чинників.

# ВИСНОВКИ

Проведений аналіз систем безпеки сучасних телекомунікаційних мереж показав, що понад 87% порушень безпеки стосуються конфіденційності, цілісності й автентичності Показано, що найбільших збитків учасникам інформаційного обміну завдають порушення цілісності й автентичності інформації. Доведено, що найефективніші методи боротьби з такими загрозами, ключове та безключове гешування, володіють суттєвими недоліками, аналіз яких показав шляхи вдосконалення моделей і методів контролю цілісності й автентичності для формування MDC і MAC кодів.

Розроблена програма реалізує математичний апарат щодо методики статистичних досліджень аналізу колізійних властивостей, що дозволяють визначити розподіл кодів, що формуються на всій множині ключових даних і отримувати оцінки колізійних властивостей з необхідною точністю.

Проведено експериментальні дослідження колізійних властивостей каскадного формування кодів контролю цілісності й автентичності даних з використанням математичного апарату. Встановлено:

Перший та другий шари формування кодів контролю цілісності й автентичності даних задовольняють вимоги універсального гешування, ймовірність колізій MAC не перевищує $\varepsilon \geq 0{,}98$. Використані функції відображення не є строго універсальними геш-функціями, оскільки не виконуються обмеження кількості правил гешування, і це не змінюється при контролі цілісності та автентичності даних.

Встановлено, що третій шар формування кодів контролю цілісності та автентичності даних повідомлень не задовольняє властивостям універсального гешування, колізійні властивості схеми автентифікації повідомлень знижуються і не відповідають поставленим вимогам.

Реалізовано модель і метод каскадного створення методів та кодів автентичності та контролю цілісності даних з використанням на останньому етапі криптографічно сильної функції хешування, яке формується завдяки медулярним перетворенням.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ


1. Євсеєв С.П., Йохов О.Ю., Король О.Г. Гешування даних в інформаційних системах. Монографія Харків: Вид. ХНЕУ, 2013. – 312 с.

2. Yevseiev, Serhii and Havrylova, Alla, “Improved UMAC algorithm with crypto-code mceliece’s scheme”, Modern Problems Of Computer Science And IT-Education: collective monograph, Pages 79 – 92, Premier Publishing s.r.o., Vienna, 2020c.

3. Євсеєв С.П. Білова М.О., Жученко О.С., Іванченко С.І., Шматко О.В. Технологія Ethernet. Лабораторний практикум з курсу “Комп’ютерні мережі” студентів спеціальностей 121, 122, 126. Львів: “Новий Світ- 2000”, 2020. – 196.

4. Hryshchuk, R., Yevseiev, S. and Shmatko, A., “Construction methodology of information security system of banking information in automated banking systems: monograph”, Pages 134–156, Premier Publishing s. r. o., Vienna, 2018.

5. Korol, Olha, Havrylova, Alla and Yevseiev Serhii, “Practical UMAC algorithms based on crypto code designs”, Przetwarzanie, transmisja I bezpieczenstwo informacji, Tom 2, Pages 221-232, Bielsko-Biala: Wydawnictwo naukowe Akademii Techniczno-Humanistycznej w Bielsku-Bialej, 2019. .

6. ГОСТ 34.310-95. міждержавний стандарт. Інформаційна технологія. криптографічний захист інформації. Процедура вироблення та перевірки електронного цифрового підпису на базі асиметричного криптографічного алгоритму. [Електронний ресурс] – К. : Держстандарт України. – Режим доступу : http://www.itsway.kiev.ua.

7. DSTU 7564–2014. Informacijni tehnologiї. Kriptografichnij zahist informaciї. Funkcija reshuvannja [Tekst]. – K. : Derzhstandart Ukraïni, 2014. – 39 s.

8. Долгов В. І. Дослідження диференціальних властивостей міні-шифрів BABY-ADe та BABY-Aes [Електронний ресурс] / В. І. Долгов, А. А. Кузнєцов, Р. В. Сергієнко та ін. – Режим доступу: http://archive.nbuv.gov.ua/portal/natural/Prre/2009_3/2.pdf.

9. ДСТУ 2481-94 Системи оброблення інформації. Інтелектуальні інформаційні технології. Терміни та визначення. – Х. : ДСТУ, 1994. – 33 с.

10. Євсєєв С. П., Король О.Г., Жукарев В.Ю. Технології комп’ютерних мереж. Мультимедійне інтерактивне електронне видання комбінованого використання. – Х.: ХНЕУ ім. С. Кузнеця, 2015. – 207 Мб. ISBN 978-966-565-2

11. Євсєєв С. П. Механізми забезпечення автентичності банківських даних у внутрішньоплатіжних системах комерційного банку. / С. П. Євсєєв, В. Є. Чевардін, С. А. Радковський. - Х.: ХНЕУ, 2008. - Віп. 6. - С. 40-44.

12. Дослідження диференціальних властивостей блочно-симетричних шифрів. / Л. С. Сорока, А. А. Кузнєцов, І. В. Московченко та ін. // Системи обробки інформації. - Х.: ХУПС, 2010. - Віп. 6(87). – С. 286–294.

13. Компьютерные сети. / Э. Таненбаум. – 4-е изд. – СПб. : Питер, 2011. – 992 с.

14. Компьютерные сети. / Э. Таненбаум, Д. Уэзеролл. – 5-е изд. – СПб. : Питер, 2012. – 960 с.

15. Конахович Г. Ф. Мережі передачі пакетних даних / Г. Ф. Конахович, В. М. Чупрін. - К.: МК-Прес, 2006. - 272 с.

16. Король О. Г. Аналіз механізмів забезпечення безпеки банківської інформації у внутрішньоплатіжних системах комерційного банку / О. Г. Король, О. О. Кузнєцов, О. М. Ткачов // Безпека та захист інформації в інформаційних та телекомунікаційних системах: матеріали І міжнародної науково-практичної конференції 28 – 29 травня 2008 р.; Управління розвитком: збірник наукових статей – Х. : Вид. ХНЄУ, 2008. - № 6 - С. 28-35.

17. Король О. Г. Аналіз загроз та механізмів захисту у внутрішньоплатіжних системах комерційного банку / О. Г. Король, С. П. Євсєєв, Н. С. Суханова // Науковий збірник Сучасна спеціальна техніка. – Київ: ДНДІ МВС України. - 2011. - Вип.1 - С. 77 - 91.

18. Король О. Г. Використання колізійних властивостей кодів автентифікації повідомлень UMAC / О. Г. Король // Системи обробки інформації. Проблеми та перспективи розвитку ІТ-індустрії. - 2010. - № 7(88). - С. 221.

19. Король О. Г. Дослідження колізійних властивостей кодів автентифікації повідомлень UMAC / О. Г. Король, А. А. Кузнєцов, С. П. Євсєєв // Прикладна радіоелектроніка. - Х.: Вид-во ХНУР, 2012. - Т. 11, № 2. - С. 171-183.

20. Король О. Г. Дослідження методів забезпечення автентичності та цілісності даних на основі односторонніх хеш-функцій / О. Г. Король, С. П. Євсєєв // Захист інформації: науково-технічний журнал. Спецвипуск (40). - 2008. - С. 50-55.

21. Король О. Г. Дослідження властивостей ключового хешування UMAC / О. Г. Король, А. А. Кузнєцов, С. А. Ісаєв // Інформаційні технології у навігації та управлінні: стан та перспективи розвитку : матеріали Першої

міжнародної науково-технічної конференції; Державне підприємство «Центральний науково-дослідний інститут навігації управління», м. Київ, 5 – 6 липня 2010р. - К. 2010. - С. 24.

22. Король О. Г. Механізми забезпечення автентичності та цілісності даних у платіжних системах / О. Г. Король // Проблеми інформатики та моделювання: матеріали Восьмої міжнародної науково-технічної конференції. – Х.: НТУ «ХПІ», 2008. – С. 39.

23. Король О. Г. Механізми забезпечення цілісності та автентичності даних у внутрішньоплатіжних системах комерційних банків / О. Г. Король, С. П. Євсєєв // Інформаційна безпека : матеріали Науково-практичної конференції, м. Київ, 26 – 27 березня 2009 нар. – К.: ДУІКТ, 2009. – С. 42–46.Король О. Г. Разработка модели и метода каскадного формирования МАС с использованием модулярных преобразований // Захист інформації : науково-технічний журнал. – 2013. – Т. 15, № 3. – С. 186.

24. Serhii Yevseiev, Volodymir Ponomarenko, Oleksandr Laptiev, Oleksandr Milov, Olha Korol, Stanislav Milevskyi, Serhii Pohasii, Andrii Tkachov, Olexander Shmatko, Yevgen Melenti, Oleksandr Sievierinov, Sergey Ostapov, Alla Gavrilova, Oleksii Tsyhanenko, Sergey Herasimov, Elena Nyemkova, Bogdan Tomashevsky, Ivan Hrod, Ivan Opirskyy, Volodymyr Zvieriev, Oleksandr Prokopenko, Vitalii Savchenko, Oleg Barabash, Valentyn Sobchuk, German Shuklin, Vladyslav Khvostenko, Oleksandr Tymochko, Maksim Pavlenko, Andrii Trystan, and Serhii Florov, SYNERGY OF BUILDING CYBERSECURITY SYSTEMS. Kharkiv, Ukraine: PC TECHNOLOGY CENTER, 2021, p. 175.

25. Р. В. Грищук, В. В. Охрімчук, “Напрямки підвищення захищеності комп’ютерних систем та мереж від кібератак”, II Міжнар. наук.-практ. конф. “Актуальні питання забезпечення кібербезпеки та захисту інформації” (Закарпатська область, Міжгірський район, село Верхнє Студене, 24-27 лют. 2016 р.). – К. : Видавництво Європейського університету, 2016 с. 60 – 61.

26. Р.В. Грищук, “Атаки на інформацію в інформаційно-комунікаційних системах”, Сучасна спеціальна техніка, №1(24), с.61 – 66. 2011.

27. Чопей Д.С. Дослідження кодів автентифікації повідомлень з використанням універсальних функцій, що хешують / Чопей Д.С. // Збірник наукових праць студентів спеціальностей «Інформаційні управляючі системи та технології», «Комп'ютерний еколого-економічний моніторинг» та МБА «Бізнес-інформатика». - Х.: ХНЕУ ім. Семена Кузнеця, 2015. - С. 55

28. Євсеєв С.П. Остапов С.Е., Король О.Г. Кібербезпека: сучасні технології захисту Навчальний посібник для студентів вищих навчальних закладів. Львів: “Новий Світ- 2000”, 2019. – 678.

29. Bierbrauer J. Universal hashing and geometric codes [Electronic resource] / J. Bierbrauer. – Access mode : http://www.math.mtu.edu/~jbierbra/ hashco1.ps.

30. Black J. "UMAC: Fast and provably secure message authentication", Advances in Cryptology. / J. Black, S. Halevi, H. Krawczyk, T. Krovetz, P. Rogaway // CRYPTO '99, LNCS, Springer-Verlag, 1999. – vol. 1666 – P.

216-233.

31. Korol O. G. Development of the Model and Method of Integrity Control and Data Authenticity Codes Generation Based On Modular Transformations / O. G. Korol // Перспективні технології і методи проектування МЕМС : матеріали IX міжнародної конференції MEMSTECH 2013. – Львів : Видавництво Львівської політехніки, 2013. – С. 79–83.

32. Korol O.G. Results of the statistical test security hash algorithms-candidates tender to select standard hash algorithm SHA-3 / O. G. Korol, S. P. Evseev // News of higher technical educational institutions of Azerbaijan. – 2012. – № 2. – С. 73–78.

33. Krawczyk H. UMAC-Message authentication code using universal hashing [Electronic resource] / H. Krawczyk, P. Rogaway. – Access mode : www.cs.ucdavis.edu/~rogaway/umac, 2000.

34. Krovetz T. UMAC – Message authentication code using universal hashing [Electronic resource]. – Access mode : http://www.cs.ucdavis.edu/~rogaway/umac.

35. NESSIE consortium «NESSIE Security report.» Deliverable report D20 – NESSIE, 2002. – NES/DOC/ENS/WP5/D20 [Electronic resource]. – Access mode : http://www.cryptonessie.org/.

36. Simmons G. J. An impersonation-proof identity verification scheme / G. J. Simmons // Computer Science. – 1988. – № 87. – P. 211–215.

37. Stinson D. R. Some constructions and bounds for authentication codes / D. R. Stinson // J. Cryptology. – 1988. – № 1. – P. 37–51.

38. UMAC: Fast and secure message authentication // Advances in Cryptology ‘CRYPTO '99 (1999) / J. Black, S. Halevi, H. Krawczyk at al. – Berlin : Springer-Verlag, 1999. – P. 216–233.

# ДОДАТКИ

# Лістинг коду програми

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Diploma_project
{
    class Umac
    {
        private static string[] zero = {"" ,"0", "00", "000", "0000",
"00000", "000000", "0000000", "00000000", "000000000", "0000000000",
"00000000000", "000000000000", "0000000000000"
                            ,       "00000000000000",       "000000000000000",
"0000000000000000",         "00000000000000000",         "000000000000000000",
"0000000000000000000", "00000000000000000000" };

        public int GetHash(string t)
        {
            string[] KminL1i ={"10011111",
                               "10101110",
                               "01111010",
                               "00010010",
                               "11110100",
                               "01101001",
                               "01101101",
                               "11001101"
                              };
            int HASH = 0;
            int m = 0;
            int count = 0;
            string text = GetDivisionBeTwo(t);
            while (m != text.Length)
            {
                string text_two_char = string.Empty;
                for (int i = m; i < m + 2; i++)
                {
                    text_two_char += text[i];
                                m = m + 2;
```

```
                byte[] byte_mas = GetByteCode(text_two_char);
                int[] Mmini = Get4BitArr(byte_mas);
                int HashMinl1 = 0;
                HashMinl1     =     (HashMinl1     +     (((SumByMod(Mmini[0],
Convert.ToInt32(KminL1i[0],      2),      16))      *      (SumByMod(Mmini[4],
Convert.ToInt32(KminL1i[4], 2), 16))) % 256)) % 256;
                HashMinl1     =     (HashMinl1     +     (((SumByMod(Mmini[0],
Convert.ToInt32(KminL1i[1],      2),      16))      *      (SumByMod(Mmini[5],
Convert.ToInt32(KminL1i[4], 2), 16))) % 256)) % 256;
                HashMinl1     =     (HashMinl1     +     (((SumByMod(Mmini[0],
Convert.ToInt32(KminL1i[2],      2),      16))      *      (SumByMod(Mmini[6],
Convert.ToInt32(KminL1i[4], 2), 16))) % 256)) % 256;
                HashMinl1     =     (HashMinl1     +     (((SumByMod(Mmini[0],
Convert.ToInt32(KminL1i[3],      2),      16))      *      (SumByMod(Mmini[7],
Convert.ToInt32(KminL1i[4], 2), 16))) % 256)) % 256;

                int YminL1 = HashMinl1;


                string[] KminL31i = { "0000", "0010", "0000", "0010" };
                string KminL32i = "0001";


                string[] YminiL2 = GetYminL3Arr(YminL1);



                int YminL3 = (((SumLevel3(YminiL2, KminL31i)) % 17) % 16) ^
(Convert.ToInt32(KminL32i, 2));
                HASH += YminL3;
                count++;

            }


            return HASH;
        }

        private int SumLevel3(string[] Ymin, string[] Kmin)
```

```
   int sum = 0;
            for (int i = 0; i < 4; i++)
            {
                int     mul     =     Convert.ToInt32(Ymin[i],     2)     *
Convert.ToInt32(Kmin[i], 2);
                sum += mul;
            }
            return sum;
        }
        private int SumByMod(int a, int b, int mod)
        {
            int result = (a + b) % mod;
            return result;
        }

        private string GetDivisionBeTwo(string text)
        {
            int text_lenght = text.Length;
            if (text_lenght % 2 != 0)
            {
                int add_char = 2 - (text_lenght % 2);
                for (int i = 0; i < add_char; i++)
                {
                    text += " ";
                }
            }
            return text;
        }

        private byte[] GetByteCode(string text)
        {
            byte[] mas = new byte[4];
            if (text.Length == 2)
            {
                mas = Encoding.Unicode.GetBytes(text);
            }
            return mas;
        }
```

```
   private static int[] Get4BitArr(byte[] arr)
        {

            int[] bitArr = new int[8];
            StringBuilder strBuild = new StringBuilder();
            int i = 0;
            foreach (byte bt in arr)
            {
                string str = Convert.ToString(bt, 2);

                int zeroNeed = 8 - str.Length;
                strBuild.Append(zero[zeroNeed]);
                strBuild.Append(str);

                bitArr[i]  =  Convert.ToInt32(strBuild.ToString().Substring(0,
4), 2);
                bitArr[i                +                1]                =
Convert.ToInt32(strBuild.ToString().Substring(4, 4), 2);
                i = i + 2;
                strBuild.Clear();

            }
            return bitArr;
        }
        private string[] GetYminL3Arr(int Hashl3)
        {
            StringBuilder strBuild = new StringBuilder();
            string[] strArr = new string[4];
            string hash = Convert.ToString(Hashl3, 2);
            int zeroNeed = 16 - hash.Length;
            strBuild.Append(zero[zeroNeed]);
            strBuild.Append(hash);

            strArr[0] = strBuild.ToString().Substring(0, 4);
            strArr[1] = strBuild.ToString().Substring(4, 4);
            strArr[2] = strBuild.ToString().Substring(8, 4);
            strArr[3] = strBuild.ToString().Substring(12, 4);
            return strArr;
        }
```