МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

ГРИЦАК Анатолій Васильович

УДК 003.26:621.39 (043.3)

МЕТОДИ ПОБУДОВИ ЕФЕКТИВНИХ КРИПТОГРАФІЧНИХ ФУНКЦІЙ ГЕШУВАННЯ

05.13.21 – «Системи захисту інформації»

Автореферат

дисертації на здобуття наукового ступеня
кандидата технічних наук

Київ – 2020

Дисертацією є рукопис.

Робота виконана на кафедрі менеджменту та безпеки інформаційних систем Вінницького національного технічного університету Міністерства освіти і науки України.

Науковий керівник: доктор технічних наук, професор
Яремчук Юрій Євгенович,
Вінницький національний технічний університет, професор кафедри менеджменту та безпеки інформаційних систем.

Офіційні опоненти: доктор технічних наук, професор
Васіліу Євген Вікторович,
директор Навчально-наукового інституту кібербезпеки, комп'ютерних і радіо технологій Одеської національної академії зв'язку ім. О.С. Попова.

кандидат технічних наук
Охріменко Тетяна Олександрівна,
докторант Національного авіаційного університету.

Захист відбудеться «27» листопада 2020 р. о 14^{00} на засіданні спеціалізованої вченої ради Д 26.062.17 при Національному авіаційному університеті за адресою: 03058, м. Київ, пр. Любомира Гузара, 1.

З дисертацією можна ознайомитись у Науково-технічній бібліотеці Національного авіаційного університету за адресою: 03058, м. Київ, пр. Любомира Гузара, 1.

Автореферат розісланий «27» жовтня 2020 р.

Учений секретар
спеціалізованої вченої ради
д.т.н., доцент С.О. Гнатюк

ЗАГАЛЬНА ХАРАКТЕРИСТИКА РОБОТИ

Актуальність. Серед спектру методів захисту інформації особливе місце займають криптографічні методи. На відміну від інших, ці методи спираються лише на властивості самої інформації і не використовують властивості її матеріальних носіїв, особливості вузлів її оброблення, передавання і зберігання. Широке використання і постійне збільшення об'єму інформаційних потоків викликає постійне зростання інтересу до криптографії. Останнім часом збільшується роль програмних криптографічних засобів захисту інформації, які не потребують великих фінансових витрат порівняно з апаратними криптосистемами. Сучасні методи шифрування гарантують надійний захист, але завжди є імовірність знаходження нових методів криптоаналізу, які дозволять послабити стійкість криптоалгоритмів.

На сьогодні існує багато криптографічних алгоритмів, серед яких зустрічаються достатньо вдалі та широко використовувані, що розроблені не тільки спецслужбами, а й приватними особами. Сучасна криптографія застосовується для розв'язання таких задач: 1) забезпечення конфіденційності даних; 2) перевірка справжності відправника (аутентифікація); 3) не заперечення авторства; 4) забезпечення цілісності даних. Остання задача полягає у тому, що отримувач може перевірити несанкціоновану модифікацію в тексті, а зловмисник не може видати змінений текст за справжній. Одним із найбільш ефективних способів розв'язання зазначеної задачі є використання методів гешування, тобто перетворення вхідних даних довільної довжини у вихідні дані (бітовий рядок) фіксованої довжини (процес перетворення називається геш-функцією, а вихідні дані геш-кодом, або дайджестом). Криптографічні функції гешування дають змогу перевірити відповідність вхідних даних дайджесту, проте не дозволяють відновити вхідні дані за наявним дайджестом – саме ця властивість дозволяє забезпечити цілісність даних. Крім зазначеної, ефективна функція гешування має забезпечувати так властивості, як висока швидкість обчислення та стійкість до колізій першого і другого роду. Серед сучасних геш-функцій варто відзначити Keccak, SHA-2, BCA, MD-5, Haval, N-hash, RIPE-MD, Kupyna та інші.

Значний внесок у розвиток теорії й практики побудови ефективних функцій гешування внесли такі вітчизняні та закордонні вчені: Д. Бернштейн, Г. Бертоні, А. Бірюков, І. Горбенко, Й. Даймен, Ю. Женг, О. Король, Т. Ланге, А. Олексійчук, Р. Олійников, Й. Пєпжик, Б. Преніл, Р. Райвест, Дж. Себері, Б. Шнайєр та ін.

Переважна більшість сучасних наукових досліджень, пов'язаних із розробкою криптографічних функцій гешування, є орієнтованими або на забезпечення високого рівня криптостійкості (такі геш-функції потребують підвищення швидкодії), або ж на забезпечення високої швидкодії (такі геш-функції потребують підвищення стійкості до криптоаналітичних атак). З огляду на це, та незважаючи на велику номенклатуру існуючих методів і рішень, розробка та дослідження нових ефективних геш-функцій, які при достатньо високій швидкодії забезпечуватимуть необхідний рівень стійкості є *актуальною науково-технічною задачею,* що має теоретичне і практичне значення.

Зв'язок роботи з науковими програмами, планами, темами. Тематика дисертаційної роботи та одержані результати безпосередньо пов'язані з "Основними науковими напрямами та найважливішими проблемами фундаментальних досліджень у галузі природничих, технічних і гуманітарних наук НАН України на 2014-2018 роки" в частині п.1.2.8.1. "Розробка методів та інформаційних технологій розв'язання задач комп'ютерної криптографії та стеганографії", зі Стратегією кібербезпеки України від 15 березня 2016 року № 96/2016 у контексті п. 4.1 "Розвиток та вдосконалення системи технічного і криптографічного захисту інформації" і Рамковою програмою ЄС з досліджень та інновацій "Горизонт 2020". Результати роботи відображені у звітах держбюджетних НДР Національного авіаційного університету "Методи забезпечення конфіденційності державних інформаційних ресурсів в інформаційно-комунікаційних системах" (№ 61/09.01.08) "Квантово-

криптографічні методи захисту критичної інформаційної інфраструктури держави" (0117U006770), у яких здобувач брав участь у якості виконавця.

Мета і задачі дослідження. Метою дисертаційної роботи є забезпечення цілісності даних в інформаційно-комунікаційних системах за рахунок розробки методів побудови і засобів реалізації ефективних криптографічних функцій гешування.

Для досягнення поставленої мети **необхідно розв'язати такі основні задачі:**

– проаналізувати сучасні методи і алгоритми побудови та реалізації ефективних криптографічних функцій гешування для виявлення їх недоліків і формалізації завдання наукового дослідження;

– розробити та удосконалити методи побудови функцій гешування для ефективного застосування у системах, для яких критичними є параметри швидкості і криптостійкості;

– удосконалити метод побудови генераторів псевдовипадкових послідовностей (ПВП) для формування статистично стійкої гами;

– удосконалити метод криптографічного захисту інформації для забезпечення конфіденційності і цілісності даних в інформаційно-комунікаційних системах;

– розробити спеціалізоване програмне забезпечення та методику для проведення експериментів і верифікації запропонованих методів.

Об'єктом дослідження є процес забезпечення цілісності та конфіденційності даних.

Предметом дослідження є методи, способи та алгоритми побудови ефективних криптографічних функцій гешування.

Методи дослідження. Проведені дослідження базуються на сучасних методах теорії криптографії (дослідження швидкості і стійкості геш-функцій), скінченних полів та елементів теорії чисел (побудова функцій гешування і криптоалгоритмів), об'єктно-орієнтованого програмування та математичної статистики (розробка програмних засобів, проведення експериментів і обробка їх результатів, аналіз колізійних властивостей геш-функцій).

Наукова новизна одержаних результатів полягає у такому:

– *вперше розроблено* метод побудови функцій гешування, який базується на структурі Меркла-Демґарда та за рахунок доповнення вхідного повідомлення розміром цього повідомлення та псевдовипадковою послідовністю salt (розраховується на основі вхідного повідомлення), використання у функції стиснення нової послідовності операцій (на основі 6-ти не лінійних функцій, операцій підстановки, додавання за модулем 2 і 2^n, циклічних і лінійних зсувів), дозволив будувати криптостійкі функції гешування;

– *вперше розроблено* метод побудови функцій гешування, який базується на структурі Меркла-Демґарда та за рахунок доповнення вхідного повідомлення псевдовипадковою послідовністю salt (розраховується на основі вхідного повідомлення та його розміру), використання у функції стиснення додаткового вектору внутрішнього стану та нової послідовності операцій (на основі 4-х не лінійних функцій, операцій підстановки, перестановки, додавання за модулем 2 і 2^n та циклічного зсуву), дозволив будувати швидкісні функції гешування;

– *удосконалено* метод побудови генераторів псевдовипадкових послідовностей, який за рахунок обробки вектора внутрішнього стану та ключового вектору операціями підстановки, циклічного зсуву, складання за модулем 2 і 2^n та 4-ма нелінійними функціями, дозволив будувати ефективні генератори псевдовипадкових послідовностей;

– *удосконалено* метод криптографічного захисту інформації, який за рахунок фіксування інформації про ідентифікатор користувача, ідентифікатор сесії, час відправлення, довжину повідомлення та його порядковий номер, а також використання нової процедури формування сеансового ключа для шифрування, дозволяє забезпечити конфіденційність і цілісність даних в інформаційно-комунікаційних системах.

Практичне значення одержаних результатів. Отримані в дисертаційній роботі результати можуть бути використані для підвищення ефективності забезпечення цілісності даних в ІКС та інших завданнях криптографічного захисту даних. Зокрема, практична цінність роботи полягає у такому:

– розроблено і реалізовано програмно шість нових функцій гешування (стійкі – Oberih-1, Oberih-2, Oberih-3, швидкісні – Barvinok-1, Barvinok-2, Barvinok-3), які дозволяють забезпечити стійкість, підвищити швидкість у 1.15-1.36 разів (Oberih) або у 1.16-1.53 разів (Barvinok) і можуть бути використані для забезпечення цілісності даних в ІКС, блокчейн системах, електронній пошті, системах миттєвого обміну повідомленнями (месенджерах) та інших сучасних застосунках;

– розроблено і реалізовано програмно три генератори ПВП (Viriy-1, Viriy-2, Viriy-3), які є швидшими у 1.02-1.22 разів в порівнянні з аналогами, що можуть бути використанні для криптографічних застосувань (генерування ключів, потокові шифри тощо) для підвищення їх ефективності;

– подано заявку на отримання патенту України на корисну модель "Спосіб побудови стійких функцій гешування" від 27.05.2020 року;

– результати дисертації використовуються у навчальному процесі Вінницького національного технічного університету, науковому процесі Національного авіаційного університету та ННВК "Інформаційно-комунікаційні системи".

Особистий внесок здобувача. Основні положення і результати дисертаційної роботи, що виносяться до захисту, отримані автором самостійно. У роботах, написаних у співавторстві, автору належать: [1,6] – розробка і реалізація стійких генераторів ПВП, [2] – реалізація і дослідження програмної системи для блокчейн; [3,5,7,8] – розробка і дослідження швидкісних та криптостійких функцій гешування; [4] – огляд методів і засобів забезпечення цілісності даних в сучасних ІКС; [9-11] – експериментальне дослідження розроблених методів забезпечення конфіденційності і цілісності даних в ІКС. З робіт, що опубліковані в співавторстві, у дисертаційній роботі використовуються виключно результати, отримані особисто здобувачем.

Апробація результатів дисертації. Основні положення дисертаційної роботи доповідалися та обговорювалися міжнародних наукових конференціях, серед яких: МНПК молодих учених і студентів "Політ. Сучасні проблеми науки" (м. Київ, 2016-2020 роки), МНТК "Проблеми експлуатації та захисту інформаційно-комунікаційних систем" (Київ, 2017 рік), МНПК "Актуальні питання забезпечення кібернетичної безпеки та захисту інформації" (Верхній Студений, 2018 рік), МНТК "Сучасні засоби зв'язку" (Мінськ, 2019 рік).

Публікації. Основні положення дисертації опубліковано в 11 наукових працях, у тому числі – 6 наукових статей (1 – у міжнародному рецензованому періодичному виданні [1], що входить до бази даних Scopus, 5 – у вітчизняних [2-4,6] і закордонних [5] фахових наукових журналах), а також 5 матеріалів і тез доповідей на конференціях [7-11].

Структура роботи та її обсяг. Дисертація складається із анотації, вступу, чотирьох розділів, загальних висновків, додатків, списку використаних джерел і має 123 сторінки основного тексту, 31 рисунок, 17 таблиць, 5 сторінок додатків. Список використаних джерел містить 125 найменувань і займає 14 сторінок. Загальний обсяг роботи 142 сторінки.

ОСНОВНА ЧАСТИНА

У **вступі** подано загальну характеристику роботи, обґрунтовано актуальність, сформульовано мету і задачі досліджень, відображено наукову новизну і практичну цінність отриманих результатів, наведено дані щодо їх апробації та впровадження.

У **першому розділі** проведено аналіз наукової літератури за темою дисертаційної роботи. Встановлено, що сучасна функція гешування має відповідати таким вимогам, як: 1) легкість обчислення геш-значення для будь-якого повідомлення; 2) неможливість утворення

повідомлення для заданого геш-коду; 3) неможливість зміни повідомлення без зміни геш-коду; 4) неможливість знаходження двох різних повідомлень з тим самим геш-кодом.

Визначено, що у сучасній криптографії необхідно виділити такі додатки геш-функцій (рис. 1), як протоколи електронного цифрового підпису, контроль цілісності даних з використанням спільного секрету, контроль цілісності даних без використання спільного секрету, генерація ПВП, протоколи встановлення ключів, протоколи автентифікації за паролем тощо.

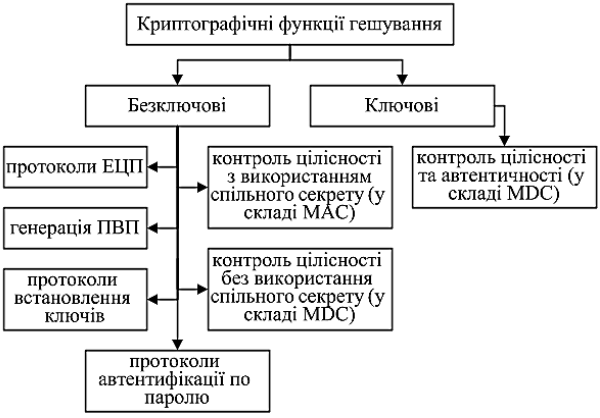

Рисунок 1 – Класифікація геш-функцій у контексті криптографічних застосувань

При обґрунтуванні та узагальненні вимог будемо вважати, що функція гешування дозволяє обчислювати геш-код з довжиною *hlen* при довільній довжині вхідного повідомлення *M*. Проведені дослідження показали, що сучасні та перспективні функції гешування мають неодмінно відповідати таким вимогам (1) – (4):

1) Складність знаходження колізії C_{col}:

$$C_{col} \geq 2^{hlen/2}. \quad (1)$$

2) Складність відновлення прообразу M (C_{prem}):

$$C_{preim} \geq 2^{hlen}. \quad (2)$$

3) Складність знаходження іншого прообразу $C_{\sec\_prem}$:

$$C_{\sec\_preim} \geq 2^{hlen}. \quad (3)$$

4) Складність знаходження колізії, усіченої на *ht* символів C_{tr_col}:

$$C_{tr\_col} \geq 2^{(hlen-ht)/2}. \quad (4)$$

Наведені вимоги дозволяють ввести безумовні критерії оцінки *j*-ї функції гешування для криптографічних застосувань, крім того, можуть бути введені умовні критерії.

На сьогодні геш-функції використовуються для розв'язання різноманітних завдань захисту даних, з огляду на що, їх спектр є досить широкою. У табл. 1 наведено результати аналізу відомих функцій гешування за такими параметрами: *S* – максимальний розмір повідомлення; *L* – довжина геш-коду (у бітах); *V* – швидкість шифрування (у Мбіт/с); *B* – розмір блоку (у бітах); *R* – кількість раундів; *W* – розмір слова; *C* – стійкість до відомих методів криптоаналізу.

Таблиця 1

Базові параметри та характеристики сучасних функцій гешування

№	*Назва*	*S*	*L*	*V*	*B*	*R*	*W*	*C*
1.	MD4	$<2^{64}$	128	2,36	512	48	32	-
2.	MD5	$<2^{64}$	128	1,74	512	64	32	-
3.	SHA-1	$<2^{64}$	160	0,75	512	80	32	-/+
4.	SHA-2/256	$<2^{64}$	256	1,85	512	64	32	+
5.	SHA-2/512	$<2^{128}$	1024	1,76	1024	80	64	+
6.	SHA-3 (Keccak)	-	512	0,12	1600	24	64	+
7.	ГОСТ Р 34.11-2012	$<2^{64}$	256	0,11	512	12	32	+
8.	RIPEMD (160)	$<2^{64}$	128	3,60	512	8	32	-
9.	N-геш (12 етапів)	$<2^{64}$	128	1,66	512	8	128	-
10.	Snerfu	$<2^{64}$	128	1,70	512	64	128	-
11.	Купина-512	$<2^{96}$	512	3,25	1024	10	64	+

У результаті аналізу (див. табл. 1) можна зробити висновок, що більш криптостійкі функції гешування (наприклад, SHA-2/256, SHA-2/512, ГОСТ Р 34.11-2012, Купина-512) потребують підвищення швидкодії. Поряд з тим, функціям гешування з високою швидкодією (для прикладу, RIPEMD (160), Snerfu, MD4, MD5) більше властиво піддаватися атакам криптоаналізу. Отже, не дивлячись на те, що існує багато функцій гешування, розробка нових геш-функцій, які при достатньо високій швидкодії забезпечуватимуть достатній рівень стійкості є актуальною – саме це і визначає завдання дисертаційного дослідження.

Другий розділ присвячено розробці двох методів побудови функцій гешування – перший метод орієнтований на застосування у системах, для яких критичним є параметр стійкість, а другий – у системах, для яких критичним є параметр швидкість.

Перший метод побудови функцій гешування (криптостійкий)

Нехай M – вхідне повідомлення, $M \in V_N$, $V_N \in \{0,1\}^N$, $N \in Z_+$, $N < 2^{128}$, H – дайджест повідомлення M, $H \in V_L$, $L = 256 \cdot l$, $l \in Z_+$. Тоді, обчислення H з M виконується у два етапи (рис. 2):

Етап 1. Етап попередньої обробки. На даному етапі вхідне повідомлення M доповнюється додатковою інформацією, таким чином, щоб результуюча довжина повідомлення була кратна $2 \cdot L$:

$$M_{rez} = (M, D, salt),$$

де M_{rez} – результуюче повідомлення, з якого буде обраховуватись H, $M_{rez} \in V_{NN}$, $NN = N + N_D + N_{salt} = 2 \cdot L \cdot t$, $t \in Z_+$, D – довжина повідомлення M, $D \in V_{N_D}$, $N_D = 128$, $salt$ – ПВП, що формується на основі M, $salt = F_{Gen}(M)$, $salt \in V_{N_{salt}}$, $N_{salt} = 4L - \left((N + N_D) \bmod 2L\right)$, F_{Gen} – деяка функція генерування ПВП на основі M.

Етап 2. Визначення дайджесту повідомлення. Спочатку повідомлення M_{rez}, $M_{rez} \in V_{NN}$, розбивається на t $2 \cdot L$ – бітних блоків:

$$M_{rez} = (m_1, m_2 .... m_t),$$

де $m_i \in V_{2 \cdot L}$, $i = \overline{1,t}$, $t = NN / 2L$.

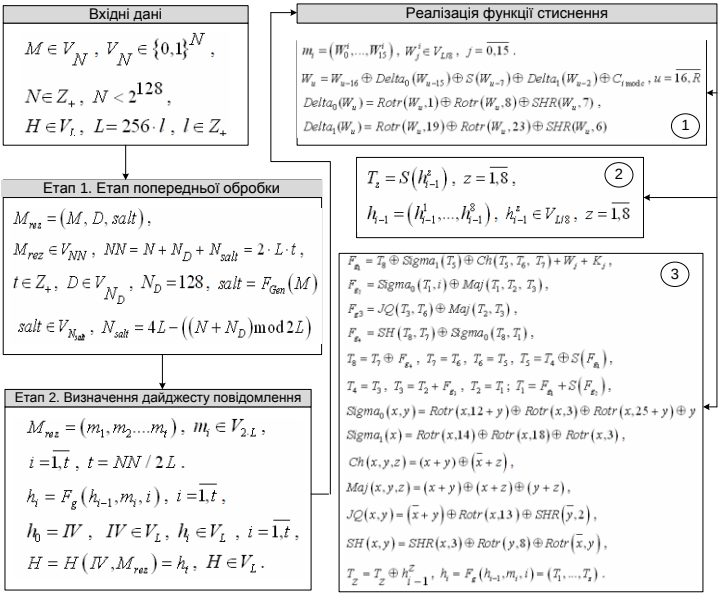

Рисунок 2 – Схема реалізації криптостійкого методу побудови функцій гешування

Далі послідовно обробляється функцію стиснення F_g кожен i-й блок повідомлення M_{rez}, проміжний дайджест $(i-1)$-го блоку та індекс i:

$$h_i = F_g\left(h_{i-1}, m_i, i\right),\ i=\overline{1,t},$$

де $h_0 = IV$, IV – вектор ініціалізації, $IV \in V_L$, h_i – проміжні значення дайджесту, $h_i \in V_L$, $i=\overline{1,t}$, F_g – функція стиснення.

Результат обробки останнього блоку t і буде дайджестом повідомлення M:

$$H = H\left(IV, M_{rez}\right) = h_t,$$

де H – дайджест повідомлення M, $H \in V_L$.

Функція стиснення F_g i-го блоку повідомлення M_{rez} виконується в три етапи: 1) розбиття блоків на слова; 2) ініціалізація змінних; 3) безпосереднє стиснення.

Етап 1 функції стиснення F_g. Кожен m_i блок повідомлення M_{rez}, $m_i \in V_{2\cdot L}$, $i=\overline{1,t}$, розкладається на 16 слів:

$$m_i = \left(W_0^i, ..., W_{15}^i\right),$$

де $W_j^i \in V_{L/8}$, $j=\overline{0,15}$.

На основі слів W_j^i, $j=\overline{0,15}$, розраховуються слова W_u^i, $W_u^i \in V_{L/8}$, $u=\overline{16,R}$:

$$W_u = W_{u-16} \oplus Delta_0\left(W_{u-15}\right) \oplus S\left(W_{u-7}\right) \oplus Delta_1\left(W_{u-2}\right) \oplus C_{i \bmod c},\ u=\overline{16,R}$$

де $Delta_0(W_u) = Rotr(W_u,1) \oplus Rotr(W_u,8) \oplus SHR(W_u,7)$,

$Delta_1(W_u) = Rotr(W_u,19) \oplus Rotr(W_u,23) \oplus SHR(W_u,6)$, $Rotr(x,l)$ – правий побітовий циклічний зсув аргументу x на l – біт; $SHR(x,l)$ – лівий зсув аргументу x на l – біт, $S(x)$ – деяка операція підстановки, C_v - наперед визначені константи, $C_v \in V_{L/8}$, $v = \overline{0,c-1}$, $c \in Z_+$, R – кількість раундів стиснення, $R \in Z_+$.

Етап 2 функції стиснення F_g. Виконується ініціалізація векторів внутрішнього стану T, $T = (T_1,...,T_8)$, $T_z \in V_{L/8}$, $z = \overline{1,8}$:

$$T_z = S\left(h_{i-1}^z\right),\ z = \overline{1,8},$$

де $h_{i-1} = \left(h_{i-1}^1,...,h_{i-1}^8\right)$, h_{i-1} – значення дайджесту $(i-1)$-го блоку, що подається на вхід функції F_g, $h_{i-1}^z \in V_{L/8}$, $z = \overline{1,8}$, $S(x)$ – деяка операція підстановки.

Етап 3 функції стиснення F_g. На даному етапі відбувається безпосереднє стиснення блоку даних $m_i \in V_{2 \cdot L}$, $i = \overline{1,t}$, $t = NN / 2L$, при цьому у кожному j-му раунді ($j = \overline{0,R}$, $R \in Z_+$) буде змінюватись значення векторів внутрішнього стану $T = (T_1,...,T_8)$, $T_z \in V_{L/8}$, $z = \overline{1,8}$, за допомогою їх перемішування із векторами W_j та константами K_j.

У кожному j-му раунді послідовно будуть виконуватися наступні дії, приведені нижче, $j = \overline{0,R}$:

$$F_{g_1} = T_8 \oplus Sigma_1(T_5) \oplus Ch(T_5, T_6, T_7) + W_j + K_j,\ F_{g_2} = Sigma_0(T_1, i) \oplus Maj(T_1, T_2, T_3),$$

$$F_{g3} = JQ(T_3, T_6) \oplus Maj(T_2, T_3),\ F_{g_4} = SH(T_8, T_7) \oplus Sigma_0(T_8, T_1),$$

$$T_8 = T_7 \oplus F_{g_4},\ T_7 = T_6,\ T_6 = T_5,\ T_5 = T_4 \oplus S\left(F_{g_1}\right),$$

$$T_4 = T_3,\ T_3 = T_2 + F_{g_3},\ T_2 = T_1;\ T_1 = F_{g_1} + S\left(F_{g_2}\right),$$

де T_z – вектори внутрішнього стану, $T_z \in V_{L/8}$, $z = \overline{1,8}$, W_j – слова, на які розбивається кожен m_i блок, K_j – наперед визначені константи, $K_j \in V_{L/8}$,

$Sigma_0(x,y) = Rotr(x,12+y) \oplus Rotr(x,3) \oplus Rotr(x,25+y) \oplus y$,

$Sigma_1(x) = Rotr(x,14) \oplus Rotr(x,18) \oplus Rotr(x,3)$, $Ch(x,y,z) = (x+y) \oplus \left(\bar{x}+z\right)$,

$Maj(x,y,z) = (x+y) \oplus (x+z) \oplus (y+z)$, $JQ(x,y) = \left(\bar{x}+y\right) \oplus Rotr(x,13) \oplus SHR\left(\bar{y},2\right)$,

$SH(x,y) = SHR(x,3) \oplus Rotr(y,8) \oplus Rotr\left(\bar{x},y\right)$, $S(x)$ – деяка операція підстановки.

Після виконання останнього R-го раунду значення векторів внутрішнього стану $T = (T_1,...,T_8)$, $T_z \in V_{L/8}$, $z = \overline{1,8}$, остаточно змінюються наступним чином:

$$T_z = T_z \oplus h_{i-1}^z,$$

де h_{i-1} – попереднє значення дайджесту, що подається на вхід функції F_g $h_{i-1}=\left(h_{i-1}^1,...,h_{i-1}^8\right)$, $h_{i-1}^z \in V_{L/8}$, $z=\overline{1,8}$.

Виходом функції стиснення F_g буде вектор h_i, $h_i \in V_L$ складений із $T_z \in V_{L/8}$, $z=\overline{1,8}$:

$$h_i = F_g\left(h_{i-1}, m_i, i\right) = \left(T_1, ..., T_z\right).$$

Другий метод побудови функцій гешування (швидкісний)

Нехай M – вхідне повідомлення, $M \in V_N$, $V_N \in \{0,1\}^N$, $N \in Z_+$, $N < 2^{128}$, H – дайджест повідомлення M, $H \in V_L$, $L = 32 \cdot p \cdot l$, $p \in Z_+$, $l \in Z_+$. Тоді, обчислення H з M виконується у два етапи:

Етап 1. Етап попередньої обробки. На даному етапі вхідне повідомлення M доповнюється додатковою інформацією, таким чином, щоб результуюча довжина повідомлення була кратна $10 \cdot L$:

$$M_{rez} = \left(M, salt\right),$$

де M_{rez} – результуюче повідомлення, з якого буде обраховуватись H, $M_{rez} \in V_{NN}$, $NN = N + N_{salt} = 10 \cdot L \cdot t$, $t \in Z_+$, $salt$ – ПВП, що формується на основі M і довжини D повідомлення M, $salt = F_{Gen}\left(M, D\right)$, $salt \in V_{N_{salt}}$, $N_{salt} = 20L - \left(N \bmod 10L\right)$, D – довжина повідомлення M, $D \in V_{N_D}$, $N_D = 128$, F_{Gen} – деяка функція генерування ПВП на основі M і D.

Етап 2. Визначення дайджесту повідомлення. Спочатку повідомлення M_{rez}, $M_{rez} \in V_{NN}$, розбивається на t $10 \cdot L$ – бітних блоків:

$$M_{rez} = \left(m_1, m_2 .... m_t\right),$$

де $m_i \in V_{10 \cdot L}$, $i = \overline{1,t}$, $t = NN / 10L$.

Далі послідовно обробляється функцію стиснення F_g кожен i-й блок повідомлення M_{rez}, проміжний дайджест $(i-1)$-го блоку, допоміжний вектор v_{i-1} та індекс i:

$$\left(h_i, v_i\right) = F_g\left(h_{i-1}, v_{i-1}, m_i, i\right),\ i = \overline{1,t},$$

де $h_0 = IV$, IV – вектор ініціалізації, $IV \in V_L$, $v_0 = 0$, h_i – проміжні значення дайджесту, $h_i \in V_L$, $i = \overline{1,t}$, v_i – значення допоміжного вектору, $v_i \in V_L$, $i = \overline{1,t}$, F_g – функція стиснення.

Результат обробки останнього блоку t і буде дайджестом повідомлення M:

$$H = H\left(IV, 0, M_{rez}\right) = h_t,$$

де H - дайджест повідомлення M, $H \in V_L$.

Функція стиснення F_g i-го блоку повідомлення M_{rez} виконується в три етапи: 1) розбиття блоків на слова; 2) ініціалізація змінних; 3) безпосереднє стиснення.

Етап 1 функції стиснення F_g. Кожен m_i блок повідомлення M_{rez}, $m_i \in V_{10 \cdot L}$, $i = \overline{1,t}$, розкладається на 40 слів:

$$m_i = \left(W_1^i, ..., W_{40}^i\right),$$

де $W_j^i \in V_{L/4}$, $j = \overline{1,40}$.

Етап 2 функції стиснення F_g. F_g. Виконується ініціалізація векторів внутрішнього стану T, $T = (T_1, ..., T_8)$, $T_z \in V_{L/4}$, $z = \overline{1,8}$:

$$T_j = h_{i-1}^j, \ j = \overline{1,4},$$

$$T_y = S\left(v_{i-1}^{y-4}\right), \ y = \overline{5,8},$$

де $h_{i-1} = \left(h_{i-1}^1, ..., h_{i-1}^4\right)$, h_{i-1} – значення дайджесту $(i-1)$-го блоку, що подається на вхід функції F_g, $h_{i-1}^j \in V_{L/4}$, $j = \overline{1,4}$, $v_{i-1} = \left(v_{i-1}^1, ..., v_{i-1}^4\right)$, v_{i-1} – значення допоміжного вектору, що подається на вхід функції F_g, $v_{i-1}^y \in V_{L/4}$, $y = \overline{1,4}$, $S(x)$ – деяка операція підстановки.

Етап 3 функції стиснення F_g. На даному етапі відбувається безпосереднє стиснення блоку даних $m_i \in V_{10 \cdot L}$, $i = \overline{1,t}$, $t = NN / 10L$, при цьому у кожному j-му раунді ($j = \overline{0,R}$, $R = 5 \cdot o$, $o \in Z_+$) буде змінюватись значення векторів внутрішнього стану $T = (T_1, ..., T_8)$, $T_z \in V_{L/4}$, $z = \overline{1,8}$, за допомогою їх перемішування із векторами W_j.

У кожному j-му раунді послідовно будуть виконуватися наступні дії, приведені нижче, $j = \overline{0,R} = \overline{0,5 \cdot o}$:

$$T_1 = F_1\left(T_1, T_2, T_3, T_4, T_5, W_{(8j+0) \bmod 40}, i, j\right), \ T_2 = F_2\left(T_2, T_3, T_4, T_5, T_6, W_{(8j+1) \bmod 40}, i, j\right),$$

$$T_3 = F_3\left(T_3, T_4, T_5, T_6, T_7, W_{(8j+2) \bmod 40}, i, j\right), \ T_4 = F_4\left(T_4, T_5, T_6, T_7, T_8, W_{(8j+3) \bmod 40}, i, j\right),$$

$$T_5 = F_1\left(T_5, T_6, T_7, T_8, T_1, W_{(8j+4) \bmod 40}, i, j\right), \ T_6 = F_2\left(T_6, T_7, T_8, T_1, T_2, W_{(8j+5) \bmod 40}, i, j\right),$$

$$T_7 = F_3\left(T_7, T_8, T_1, T_2, T_3, W_{(8j+6) \bmod 40}, i, j\right), \ T_8 = F_4\left(T_8, T_1, T_2, T_3, T_4, W_{(8j+7) \bmod 40}, i, j\right),$$

де T_z – вектори внутрішнього стану, $T_z \in V_{L/4}$, $z = \overline{1,8}$, W_e – слова, на які розбивається кожен m_i блок, $W_e \in V_{L/4}$,

$$F_1(A, B, C, D, E, W, i, j) = \left(\left(A + \overline{E}\right) \oplus P(B) \oplus S\left(\overline{C} \oplus W\right) \oplus i\right) <<< (D + j),$$

$$F_2(A, B, C, D, E, W, i, j) = \left((A + C) \oplus P\left(B \oplus S(D + j + \overline{E})\right) \oplus W \oplus i\right) <<< C,$$

$$F_3(A, B, C, D, E, W, i, j) = \left(A \oplus S\left(S\left(B \oplus W \oplus D\right) + j + C + \overline{W}\right) \oplus i\right) <<< E,$$

$$F_1(A, B, C, D, E, W, i, j) = \left(A \oplus S\left(P\left(C + D + \overline{W}\right) \oplus E\right) \oplus i\right) <<< (B + j),$$

$S(x)$ – деяка операція підстановки, $P(x)$ – деяка операція перестановки.

Виходом функції стиснення F_g буде вектори h_i і v_i, $h_i \in V_L$ складений із $T_j \in V_{L/4}$, $j = \overline{1,4}$, $v_i \in V_L$ складений із $T_y \in V_{L/4}$, $y = \overline{5,8}$:

$$(h_i, v_i) = F_g(h_{i-1}, v_{i-1}, m_i, i) = ((T_1, ..., T_4), (T_5, ..., T_8)).$$

Таким чином, у другому розділі роботи наведено опис швидкісного і криптостійкого методів побудови функцій гешування.

У **третьому розділі** наведено розробку методу побудови генераторів псевдовипадкових послідовностей та методу криптографічного захисту інформації

Метод побудови генераторів ПВП

Нехай $n, t \in Z_+$, тоді для генерації ПВП M, $M \in V_N$, $V_N \in \{0,1\}^N$, довжиною $N = n \cdot t$ біт, потрібно сформувати t послідовностей довжиною n біт кожна:

$$M = (m_1, m_2, ..., m_{t-1}, m_t),\ m_i \in V_n,\ i = \overline{1,t}.$$

Процес генерації кожного m_i, $m_i \in V_n$, $i = \overline{1,t}$, відбувається наступним чином:

$$m_i, E_i = F_{gen}(E_{i-1}, K, i),\ i = \overline{1,t},$$

де E_i – вектор внутрішнього стану генератора після генерації i-го m_i, $E_i \in V_e$, $e \in Z_+$, $E_0 = IV$, IV – вектор ініціалізації, $IV \in V_e$, K – ключовий вектор для генерації послідовності, $K \in V_k$, $k \in Z_+$, F_{gen} - функція генерації послідовності m_i.

Функція $F_{gen}(E, K, i)$ виконується в два етапи: 1) ініціалізація змінних; 2) формування послідовності.

Етап 1 функції $F_{gen}(E, K, i)$. На початку виконується обробка вектора внутрішнього стану E, $E \in V_e$:

$$E = S(E) <<< i,$$

де $x <<< y$ – операція правого побітового циклічного зсуву аргументу x на y – біт, $S(x)$ – деяка операція підстановки.

Далі вектор внутрішнього стану генератора E і ключовий вектор K розкладаються на 4 частини:

$$E = (E_a, E_b, E_c, E_d),\ E \in V_e,\ e = a + b + c + d,$$
$$E_a \in V_a,\ E_b \in V_b,\ E_c \in V_c,\ E_d \in V_d,\ a, b, c, d \in Z_+,$$
$$K = (K_a, K_b, K_c, K_d),\ K \in V_k,\ k = a' + b' + c' + d',$$
$$K_a \in V_{a'},\ K_b \in V_{b'},\ K_c \in V_{c'},\ K_d \in V_{d'},\ a', b', c', d' \in Z_+.$$

Вектори E_a, E_b, E_c, E_d і K_a, K_b, K_c, K_d будуть використовуватись в наступному етапі функції $F_{gen}(E, K, i)$.

Етап 2 функції $F_{gen}(E, K, i)$. На даному етапі виконується формування послідовності m, $m \in V_n$. Для цього використовуються чотири додаткових функції $F_A(E_a, K_a, i)$,

$F_B(E_b, K_b, i)$, $F_C(E_c, K_c, i)$ і $F_D(E_d, K_d, i)$, функції F_A, F_B, F_C і F_D – деякі функції, що на вхід приймають значення певного вектора внутрішнього стану і ключового вектора, а на вихід передається послідовність довжини n біт (ці функції можуть бути побудовані на основі нелінійних регістрів зсуву, блокових і потокових шифрів, геш-функцій тощо).

Тоді, процес генерації послідовності m, $m \in V_n$ та нового значення вектора внутрішнього стану E, $E \in V_e$ в функції $F_{gen}(E, K, i)$ буде таким:

Крок 1. Сформувати додаткові вектори A, B, C і D, та отримати нові значення векторів E_a, E_b, E_c і E_d:

$$A, E_a = F_A(E_a, K_a, i),\ A \in V_n,\ E_a \in V_a,$$
$$B, E_b = F_B(E_b, K_b, i),\ B \in V_n,\ E_b \in V_b,$$
$$C, E_c = F_C(E_c, K_c, i),\ C \in V_n,\ E_c \in V_c,$$
$$D, E_d = F_D(E_d, K_d, i),\ D \in V_n,\ E_d \in V_d.$$

Крок 2. Розрахувати нове значення вектору внутрішнього стану E, $E \in V_e$:

$$E = (E_b, E_d, E_a, E_c).$$

Крок 3. Сформувати послідовності m, $m \in V_n$:

$$AB = A <<< B,\ AB \in V_n,$$
$$CD = C <<< D,\ CD \in V_n,$$
$$BC = B + CD,\ BC \in V_n,$$
$$AD = AB \oplus D,\ AD \in V_n,$$
$$m = \overline{AD} \oplus S(BC),\ m \in V_n,$$

де $\oplus$ і $+$ відповідають відповідно операціям додавання за модулем 2 і 2^n, $S(x)$ – операція підстановки.

Виходом функції $F_{gen}(E, K, i)$ будуть вектори m, $m \in V_n$ і E, $E \in V_e$:

$$(m, E) = F_{gen}(E, K, i).$$

Метод криптографічного захисту інформації

Нехай маємо повідомлення M, $M \in V_m$, $V_m \in \{0,1\}^m$, яке потрібно зашифрувати для передавання.

Тоді для передавання підготовлюється наступне повідомлення (містить окрім повідомлення M, інформацію про ідентифікатор користувача та ідентифікатор сесії (S, $S \in V_s$, $s \in Z_+$), інформацію про час відправлення і довжину повідомлення (ID, $ID \in V_{id}$, $id \in Z_+$) та порядковий номер повідомлення (PD, $PD \in V_{pd}$, $pd \in Z_+$)):

$$P = (S, ID, M, PD),$$

де $P \in V_p$, $p = m + s + id + pd$.

Розглянемо поетапно схему роботи алгоритму шифрування (рис. 3).

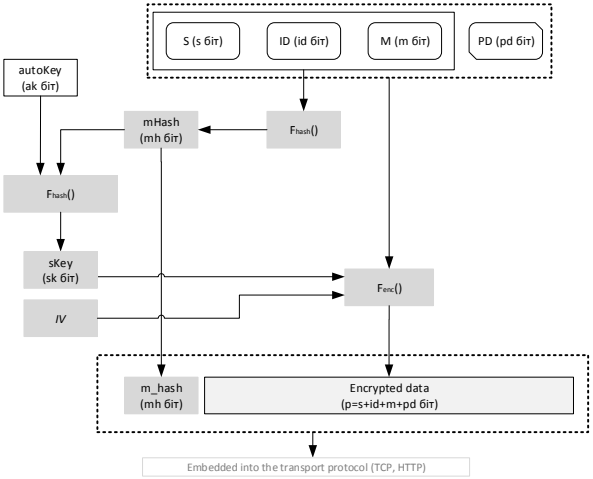

Рис. 3. Схема роботи розробленого методу криптографічного захисту інформації

Етап 1. Формування блоку даних DB для обрахунку геш значення:

$$DB = (S, ID, M),$$

де $DB \in V_{db}$, $db = m + s + id$, S – ідентифікатор користувача та ідентифікатор сесії, $S \in V_s$, $s \in Z_+$, ID – інформація про час відправлення і довжину повідомлення, $ID \in V_{id}$, $id \in Z_+$, M – саме повідомлення, $M \in V_m$.

Етап 2. Формування геш значення повідомлення DB:

$$mHash = F_{hash}(DB),$$

де $mHash$ – геш значення DB, $DB \in V_{db}$, $mHash \in V_{mh}$, $mh \in Z_+$, $F_{hash}(x)$ – деяка функція гешування.

Етап 3. Формування ключа сеансу $sKey$:

$$sKey = F_{hash}(authKey, mHash),$$

де $sKey$ – сеансовий ключ, $sKey \in V_{sk}$, $sk \in Z_+$, $authKey$ – ключ автентифікації, $authKey \in V_{ak}$, $ak \in Z_+$, $F_{hash}(x)$ – деяка функція гешування.

Етап 4. Шифрування за допомого криптографічного алгоритму:

$$EncP = F_{enc}(P, sKey, IV),$$

де $EncP$ – зашифроване повідомлення P, $EncP \in V_p$, P – повідомлення із додатковою інформацією, $P \in V_p$, $sKey$ – сеансовий ключ, $sKey \in V_{sk}$, IV – вектор ініціалізації, $IV \in V_{iv}$, $iv \in Z_+$, $F_{enc}(P, sKey, IV)$ – деяка функція шифрування (може бути побудована на основі блокових і потокових шифрів, геш-функцій тощо).

Етап 5. Формування кінцевого повідомлення:

$$EncMes = (mHash, EncP),$$

де $EncMes$ – кінцеве повідомлення, $EncMes \in V_{p+mh}$, $mHash$ – геш значення DB, $DB \in V_{db}$, $EncP$ – зашифроване повідомлення P, $EncP \in V_p$.

Таким чином, у третьому розділі описано методи, що можна використовувати в криптографічних системах захисту інформації для формування ключових даних, дайджестів повідомлень та передавання конфіденційних повідомлень.

Четвертий розділ присвячено практичним реалізаціям та експериментальним дослідженням розроблених рішень. Розроблено методику проведення експерименту, обґрунтовано доцільність вибору бази експерименту, визначено мету та задачі експерименту, вхідні та вихідні параметри, гіпотезу і критерії дослідження, достатність експериментальних об'єктів та послідовність необхідних дій. Для проведення експериментів на базі розроблених у другому розділі методів побудови функцій гешування було розроблено шість функцій гешування, а на базі розробленого у третьому розділі методу побудови генераторів ПВП було розроблено три генератори. Над розробленими криптографічними алгоритмами були проведені експериментальні дослідження згідно розробленій методиці.

На базі першого методу побудови функцій гешування (криптостійкого) було розроблено три функції гешування з такими параметрами:

Oberih-1 – $l = 1$, $L = 256 \cdot 1 = 256$, $H \in V_{256}$, у якості операції $S(x)$ – використовується операція виду: $S(x) = (s_0(x_1), s_0(x_0))$, де $x_j \in V_{16}$, $j = \overline{0,1}$, s_0 – підстановка на множині V_{16}, що спроектована згідно (5).

Oberih-2 – $l = 2$, $L = 256 \cdot 2 = 512$, $H \in V_{512}$, у якості операції $S(x)$ – використовується операція виду: $S(x) = (s_1(x_3), ..., s_0(x_0))$, де $x_j \in V_{16}$, $j = \overline{0,3}$, s_b – підстановка на множині V_{16}, $b = \overline{0,1}$ (почергово використовуються 2 різні таблиці замін, що спроектована згідно (5)).

Oberih-3 – $l = 4$, $L = 256 \cdot 4 = 1024$, $H \in V_{1024}$, у якості операції $S(x)$ – використовується операція виду: $S(x) = (s_3(x_7), ..., s_0(x_0))$, де $x_j \in V_{16}$, $j = \overline{0,7}$, s_b – підстановка на множині V_{16}, $b = \overline{0,3}$ (почергово використовуються 4 різні таблиці замін, що спроектована згідно (5)).

На базі другого методу побудови функцій гешування (швидкісного) було розроблено три функцій гешування з такими параметрами:

Barvinok-1 – $p = 8$, $l = 1$, $L = 32 \cdot p \cdot l = 256$, $H \in V_{256}$, у якості операції $S(x)$ – використовується операція виду: $S(x) = (s_0(x_3), ..., s_0(x_0))$, де $x_j \in V_8$, $j = \overline{0,3}$, s_0 – підстановка на множині V_8, що спроектована згідно (5).

Barvinok-2 – $p = 8$, $l = 2$, $L = 32 \cdot p \cdot l = 512$, $H \in V_{512}$, у якості операції $S(x)$ – використовується операція виду: $S(x) = (s_1(x_7), ..., s_0(x_0))$, де $x_j \in V_8$, $j = \overline{0,7}$, s_b – підстановка на множині V_8, $b = \overline{0,1}$ (почергово використовуються 2 різні таблиці замін, що спроектована згідно (5)).

Barvinok-3 – $p = 8$, $l = 4$, $L = 32 \cdot p \cdot l = 1024$, $H \in V_{1024}$, у якості операції $S(x)$ – використовується операція виду: $S(x) = (s_3(x_{15}), ..., s_0(x_0))$, де $x_j \in V_8$, $j = \overline{0,15}$, s_b – підстановка на множині V_8, $b = \overline{0,3}$ (почергово використовуються 4 різні таблиці замін, що спроектована згідно (5)).

Для генерації таблиці замін на множині V_a, було обраховано для кожного X, $X \in V_a$ при фіксованих C, V, M ($C \in V_a$, $V \in V_a$, M – квадратна не вироджена матриця над полем $GF(2)$ розміром $a \times a$) усі значення таблиці підстановок:

$$s(X) = M \cdot (C / X)^{-1} \oplus V. \qquad (5)$$

На базі методу побудови генераторів ПВП було розроблено три генератори з такими параметрами:

Viriy-1 – $n = 128$, $a = 128$, $b = 100$, $c = 111$, $d = 173$, $e = a + b + c + d = 512$, $a' = 128$, $b' = 128$, $c' = 128$, $d' = 128$, $k = a' + b' + c' + d' = 512$. F_A, F_B, F_C і F_D – функції, що побудовані на основі нелінійних регістрів зсуву. У якості операції $S(x)$ – використовується операція виду: $S(x) = (s_0(x_{31}), ..., s_0(x_0))$, де $x_j \in V_{16}$, $j = \overline{0,31}$, s_0 – підстановка на множині V_{16}, що спроектована згідно (5).

Viriy-2 – $n = 256$, $a = 138$, $b = 120$, $c = 116$, $d = 138$, $e = a + b + c + d = 512$, $a' = 128$, $b' = 128$, $c' = 128$, $d' = 128$, $k = a' + b' + c' + d' = 512$. F_A, F_B, F_C і F_D – функції, що побудовані на основі нелінійних регістрів зсуву. У якості операції $S(x)$ – використовується операція виду: $S(x) = (s_7(x_{63}), ..., s_0(x_0))$, де $x_j \in V_8$, $j = \overline{0,63}$, s_b – підстановка на множині V_8, $b = \overline{0,7}$ (почергово використовуються 8 різних таблиць замін, що спроектовані згідно (5)).

Viriy-3 – $n = 128,\ a = 128,\ b = 128,\ c = 128,\ d = 128,\ e = a + b + c + d = 512,$ $a' = 128,\ b' = 128,\ c' = 128,\ d' = 128,\ k = a' + b' + c' + d' = 512.$ F_A, F_B, F_C і F_D – функції, що побудовані на основі AES-128. У якості операції $S(x)$ – використовується операція виду: $S(x) = (s_0(x_{63}), ..., s_0(x_0))$, де $x_j \in V_8$, $j = \overline{0,63}$, s_0 – підстановка на множині V_8, що спроектована згідно (5).

Для проведення експериментальних досліджень запропоновані алгоритми були програмно реалізовані у вигляді консольних додатків на мові програмування C++ (середовище розробки Microsoft Visual Studio 2013 (Release Version)).

Таблиця 2

Результати дослідження за методикою NIST STS

Генератор ПВП	*Кількість тестів, у яких тестування пройшло*	
	99% послід.	*96% послід.*
BBS	133.4 (70.96%)	188 (100%)
SHA-256	132.2 (70.32%)	187.9 (99.94%)
SHA-512	134.3 (71.44%)	188 (100%)
Snow	134.8 (71.70%)	188 (100%)
Trivium	130.1 (69.20%)	187.6 (99.78%)
Oberih-1	133.0 (70.74%)	188 (100%)
Oberih-2	134.6 (71.59%)	188 (100%)
Oberih-3	136.1 (72,39%)	188 (100%)
Barvinok-1	132.3 (70.37%)	187.9 (99.94%)
Barvinok-2	131.7 (70.05%)	188 (100%)
Barvinok-3	135.5 (72,07%)	188 (100%)
Viriy-1	134.1 (71,32%)	188 (100%)
Viriy-2	136.4 (72,55%)	187.8 (99.89%)
Viriy-3	137.3 (73,03%)	188 (100%)

Спочатку досліджувались статистичні характеристики запропонованих функцій гешування і генераторів ПВП за методикою NIST STS. Результати порівнювались із результатами генератора псевдовипадкових послідовностей BBS, функціями гешування SHA-256, SHA-512, потоковими шифрами Snow і Trivium. Зауважимо, що для цього дослідження, на основі розроблених геш-функцій і функцій SHA-256 та SHA-512, були побудовані генератори ПВП для створення файлів необхідної довжини для статистичних тестів NIST STS.

Для кожного алгоритму генерувалось 10 файлів із ПВП розміром 100 Мбіт, які і досліджувались за методикою NIST STS. У табл. 2 наведено усереднені результати експериментальних досліджень. Як видно з результатів, розроблені алгоритми пройшли комплексний контроль за методикою NIST STS, та показали не гірші, а в деяких випадках і кращі, результати ніж відомі алгоритми. Також, у роботі досліджувались статистичні характеристики запропонованих функцій гешування і генераторів ПВП за методикою DIEHARD, що теж були успішними.

Також, були проведені дослідження швидкісних характеристик розроблених функцій гешування. Результати порівнювались із функцією гешування SHA-512. Для дослідження були випадковим чином обрані кілька файлів різного розміру, для кожного з яких

визначався його дайджест досліджуваним алгоритмом, при цьому замірявся час обробки. Результати експериментальних досліджень наведено у табл. 3.

Таблиця 3

Результати дослідження швидкісних характеристик функцій гешування

Функцій гешування	*Файл 1, 1 МБ*		*Файл 2, 10 МБ*		*Файл 3, 100 МБ*	
	t, с	v, МБ / с	t, с	v, МБ / с	t, с	v, МБ / с
SHA-512	0.015	66.67	0.145	68.96	1.38	72,46
Oberih-1	0.012	83.33	0.114	87.72	1.07	93.45
Oberih-2	0.011	90.91	0.109	91.74	1.01	99.01
Oberih-3	0.013	76.92	0.121	82.64	1.13	88.50
Barvinok-1	0.010	100.00	0.096	104.16	0.90	111.11
Barvinok-2	0.011	90.91	0.105	95.23	0.96	104.17
Barvinok-3	0.013	76.92	0.117	85.47	1.12	89.29

Згідно з результатами дослідження швидкість обробки розробленими функціями гешування кращі ніж у функції SHA-512, так алгоритми Oberih швидші за SHA-512 у 1.15-1.36 рази, а алгоритми Barvinok – у 1.16-1.53 рази.

Також були проведені дослідження швидкісних характеристик розроблених генераторів ПВП. Результати порівнювались із алгоритмом Snow. Для дослідження генерувались файли різного розміру, при цьому замірявся час обробки. Результати експериментальних досліджень наведено у табл. 4. Згідно з результатами дослідження швидкість алгоритму Viriy краща за Snow у 1.02-1.22 рази (за виключенням двох результатів алгоритму Viriy-3).

Таблиця 4

Результати дослідження швидкісних характеристик генераторів ПВП

Функцій гешування	*Файл 1, 1 МБ*		*Файл 2, 10 МБ*		*Файл 3, 100 МБ*	
	t, с	v, МБ / с	t, с	v, МБ / с	t, с	v, МБ / с
Snow	0.011	90.91	0.107	93.46	1.01	99.01
Viriy-1	0.009	111.11	0.091	109.89	0.88	113.64
Viriy-2	0.010	100.00	0.098	102.04	0.92	108.70
Viriy-3	0.014	71.43	0.112	89.29	0.99	101.01

Також, у розділі проводяться дослідження колізійних властивостей розроблених функцій гешування (дослідження проводились на зменшених версіях функцій гешування, згідно раніше відомої методики). Для цього вводяться статистичні показники, що характеризують колізійні властивості функцій гешування, що й дозволяють, використовуючи методи теорії імовірності й математичної статистики, одержувати оцінки із заданими довірчим інтервалом і необхідною точністю. У табл. 5 наведено результати даного дослідження (назви розроблених геш-функцій скорочено).

Таблиця 5

Результати дослідження колізійних властивостей функцій гешування на міні версіях

	SHA-512	Ob.-1	Ob.-2	Ob.-3	Bar.-1	Bar.-2	Bar.-3
$\tilde{m}(n_1)$	4.01	3.11	2.72	2.99	3.51	4.07	3.81
$\tilde{D}(n_1)$	0.17	0.23	0.32	0.30	0.28	0.27	0.25
$P_\partial = P(\lvert\tilde{m}(n_1) - m(n_1)\rvert < 0.1)$	0.96	0.96	0.96	0.97	0.95	0.96	0.94
$\tilde{m}(n_2)$	4.41	4.02	3.89	4.11	4.38	4.29	4.35
$\tilde{D}(n_2)$	0.39	0.42	0.40	0.39	0.37	0.45	0.44

$P_{\partial} = P\left(\left\|\tilde{m}(n_2) - m(n_2)\right\| < 0.1\right)$	0.88	0.93	0.95	0.96	0.97	0.95	0.93
$\tilde{m}(n_3)$	5.42	4.87	4.75	4.99	5.31	5.39	5.48
$\tilde{D}(n_3)$	0.23	0.31	0.30	0.25	0.24	0.31	0.39
$P_{\partial} = P\left(\left\|\tilde{m}(n_3) - m(n_3)\right\| < 0.1\right)$	0.95	0.93	0.98	0.92	0.93	0.95	0.96

У табл. 5 перший показник $n_1(x_1, x_2)$ характеризує кількість правил гешування, при яких для заданих $x_1, x_2 \in A$, $x_1 \neq x_2$ виконується рівність (6), тобто кількість ключів, при яких існує колізія для двох вхідних послідовностей x_1 і x_2.

$$n_1(x_1, x_2) = \left|\left\{h \in H : h(x_1) = h(x_2)\right\}\right|, x_1, x_2 \in A, x_1 \neq x_2 \quad (6)$$

У табл. 5 другий показник $n_2(x_1, y_1)$ характеризує кількість правил гешування, при яких для заданих $x_1 \in A$, $y_1 \in B$ виконується рівність (7), тобто кількість ключів, при яких для вхідної послідовності x_1 значення геш-коду y_1 не змінюється.

$$n_2(x_1, y_1) = \left|\left\{h \in H : h(x_1) = y\right\}\right|, x_1 \in A, y_1 \in B \quad (7)$$

У табл. 5 третій показник $n_3(x_1, x_2, y_1, y_2)$ характеризує кількість правил гешування, при яких для заданих $x_1, x_2 \in A$, $x_1 \neq x_2$, $y_1, y_2 \in B$ виконується рівність (8), тобто кількість ключів, при яких для двох вхідних послідовностей x_1 і x_2 відповідні їм значення геш-кодів y_1 і y_2 не змінюються.

$$n_3(x_1, x_2, y_1, y_2) = \left|\left\{h \in H : h(x_1) = y_1, h(x_2) = y_2\right\}\right|, x_1, x_2 \in A, x_1 \neq x_2, y_1, y_2 \in B \quad (8)$$

де $m(n_1)$, $m(n_2)$ та $m(n_3)$ – математичні сподівання максимумів кількості правил гешування при яких виконуються рівності (6), (7) і (8) відповідно.

Дисперсії $D(n_1)$, $D(n_2)$ і $D(n_3)$, що характеризують розсіювання значень кількості правил гешування, при яких виконуються рівності (6), (7) і (8), щодо їх математичних сподівань $m(n_1)$, $m(n_2)$ і $m(n_3)$ відповідно.

Під час експериментальних дослідженнях колізійних властивостей гешування оцінювалось середнє арифметичне спостережуваних значень максимумів $\tilde{m}(n_i)$, дисперсію $\tilde{D}(n_i)$, $i = \overline{1,3}$ і довірчою вірогідністю отриманих середньостатистичних оцінок P_{∂}.

На основі отриманих результатів можна стверджувати, що запропоновані функції гешування мають не гірші колізійні властивості ніж у функції гешування SHA-512.

У **додатках** вміщено акти впровадження результатів дисертаційної роботи.

ВИСНОВКИ

Результатом виконаної роботи є розв'язання актуальної і важливої науково-технічної задачі розробки та дослідження нових ефективних геш-функцій, які при достатньо високій швидкодії забезпечуватимуть необхідний рівень стійкості.

У процесі виконання дисертаційної роботи отримані такі вагомі результати:

1. Проведено аналіз сучасних методів і алгоритмів побудови та реалізації ефективних криптографічних функцій гешування, що дозволило виявити їх недоліки і формалізувати завдання наукового дослідження. Серед виявлених недоліків основними є уразливість відомих геш-функцій до криптоаналітичних атак, низька швидкість шифрування і високі вимоги до обчислювальних засобів.

2. Розроблено метод побудови функцій гешування, який за рахунок доповнення вхідного повідомлення розміром цього повідомлення та ПВП salt (розраховується на основі вхідного повідомлення), використання у функції стиснення нової послідовності операцій (на основі 6-ти нелінійних функцій, операцій підстановки, додавання за модулем 2 і 2^n , циклічних і лінійних зсувів), дозволив будувати криптостійкі функції гешування. На основі цього методу було розроблено, реалізовано програмно і досліджено три нові стійкі геш-функції, які дозволяють підвищити швидкість у 1.15-1.36 разів та можуть застосовуватись у системах, для яких критичним є параметр стійкості.

3. Розроблено метод побудови функцій гешування, який за рахунок доповнення вхідного повідомлення ПВП salt (розраховується на основі вхідного повідомлення та його розміру), використання у функції стиснення додаткового вектору внутрішнього стану та нової послідовності операцій (на основі 4-ьох не лінійних функцій, операцій підстановки, перестановки, додавання за модулем 2 і 2^n та циклічного зсуву), дозволив будувати швидкісні функції гешування. На основі цього методу було розроблено, реалізовано програмно і досліджено три нові швидкісні геш-функції, які дозволяють підвищити швидкість у 1.16-1.53 разів і можуть застосовуватись у системах, для яких критичним є параметр швидкості.

4. Удосконалено метод побудови генераторів ПВП, який за рахунок обробки вектора внутрішнього стану та ключового вектору операціями підстановки, циклічного зсуву, складання за модулем 2 і 2^n та 4-ма нелінійними функціями, дозволив будувати ефективні генератори ПВП. На основі цього методу розроблено і реалізовано програмно три генератори ПВП, які будуть корисними як для функцій гешування, так і для інших криптографічних застосувань (генерування ключів, потокові шифри тощо). Крім того, розроблені генератори ПВП Viriy є більш швидкими за аналоги (зокрема, у 1.02-1.22 разів в порівнянні з алгоритмом Snow).

5. Удосконалено метод криптографічного захисту інформації, що за рахунок фіксування інформації про ідентифікатор користувача, ідентифікатор сесії, час відправлення, довжину повідомлення та його порядковий номер, а також використання нової процедури формування сеансового ключа для шифрування, дає можливість забезпечити конфіденційність і цілісність даних в ІКС.

6. Розроблено спеціалізоване програмне забезпечення у вигляді консольних додатків на мові програмування C++ (середовище розробки Microsoft Visual Studio 2013 (Release Version)) та методику, що дозволило провести експерименти і верифікувати запропоновані методи. Результати дисертаційної роботи використовуються у навчальному процесі Вінницького національного технічного університету, науковому процесі Національного авіаційного університету та ННВК "Інформаційно-комунікаційні системи", що підтверджено відповідними актами впровадження.

ПУБЛІКАЦІЇ ЗДОБУВАЧА ЗА ТЕМОЮ ДИСЕРТАЦІЇ

1. S. Gnatyuk, A. Hrytsak, V. Kinzeryavyy, N. Seilova et al, "Modern Method and Software Tool for Guaranteed Data Deletion an Advanced Big Data Systems", *Advances in Intelligent Systems and Computing*, Vol. 902, pp. 581-590, 2019, ISSN 2194-5357 *(Scopus)*.

2. А. Грицак, І. Березовий, І. Гринь, В. Кінзерявий, "Програмна система захисту засобів зберігання криптовалют", *Вісник Інженерної академії України,* №1, с. 128-139, 2018.

3. Н. Остапенко, В. Кінзерявий, А. Грицак, К. Кириченко, "Удосконалена функція гешування MD4", *Безпека інформації,* Том 24, №2, 2018.

4. Ю. Яремчук, А. Грицак, Д. Присяжний, "Сучасні підходи до побудови і реалізації ефективних криптографічних функцій гешування", *Вісник Інженерної академії України,* №4, с. 221-228, 2017.

5. A. Hrytsak, V. Kinzeryavyy, D. Prysiazhnyi, Yu. Burmak and Ye. Samoylik, “High-Speed and Secure Hash Function for Blockchain Security Mechanisms”, *Scientific and Practical Cyber Security Journal (SPCSJ)*, Vol. 4, Issue 1, pp. 65-70, 2020.

6. А. Грицак, В. Катаєв, В. Леонтьєв, Н. Ляховченко, “Проблеми активного захисту інформації від витоку через віброакустичні канали”, *Реєстрація, зберігання і обробка даних,* Том 18, №3, с.54-59, 2016.

7. А.В. Грицак, “Застосування алгоритмів гешування в технології блокчейн”, *тези доповідей XVII міжнар. наук.-практ. конф. молодих учених і студентів “Політ. Сучасні проблеми науки”*, 5-7 квітня 2017 р., К., с. 77, 2017.

8. К.С. Кириченко, В.М. Кінзерявий, А.В. Грицак, М.Б. Александер, “Перспективна криптографічна функція гешування”, *Матеріали міжнар. наук.-практ. конф. “Актуальні питання забезпечення кібернетичної безпеки та захисту інформації”,* 21-24 лютого 2018 р., Верхній Студений, с. 68-69, 2018.

9. А.В. Грицак, “Дослідження методу побудови функції гешування на основі алгоритму MD4”, *тези доповідей XVI міжнар. наук.-практ. конф. молодих учених і студентів “Політ. Сучасні проблеми науки”,* 04-05 квітня 2016 р., К., с. 104, 2016.

10. А.В. Грицак, “Экспериментальное исследование криптостойкости разработанных функций хеширования”, *материалы XXIII международной научно-технической конференции “Современные средства связи”,* 17-18 октября 2019 р., Минск, с. 56-59, 2019.

11. А.В. Грицак, “Дослідження удосконаленого метода забезпечення конфіденційності та цілісності даних в інформаційно-телекомунікаційних системах”, *тези доповідей Міжнар. наук.-практ. конф. молодих учених і студентів “Політ. Сучасні проблеми науки”,* 01-03 квітня 2020 р., К., с. 159, 2020.

АНОТАЦІЯ

Грицак А.В. Методи побудови ефективних криптографічних функцій гешування. – Рукопис.

Дисертація на здобуття наукового ступеня кандидата технічних наук за спеціальністю 05.13.21 – Системи захисту інформації. – Національний авіаційний університет, Київ, 2020.

Дисертаційна робота присвячена розв’язанню актуальної наукової задачі розробки та дослідження нових ефективних геш-функцій, які при достатньо високій швидкодії забезпечуватимуть необхідний рівень стійкості.

Проведено аналіз сучасних методів і алгоритмів побудови та реалізації ефективних криптографічних функцій гешування, що дозволило виявити їх недоліки і формалізувати завдання наукового дослідження. Розроблено методи побудови функцій гешування, які дозволили підвищити стійкість і швидкість криптографічної обробки даних. Удосконалено метод побудови генераторів ПВП, що дозволило формувати статистично стійку гаму для криптографічних застосувань. Удосконалено метод криптографічного захисту інформації, що дало можливість забезпечити конфіденційність і цілісність даних. Розроблено спеціалізоване програмне забезпечення у вигляді консольних додатків на мові програмування С++ та методику, що дозволило провести експерименти і верифікувати запропоновані методи. результати дисертації використовуються у навчальному процесі Вінницького національного технічного університету, науковому процесі Національного авіаційного університету та ННВК “Інформаційно-комунікаційні системи”,

Ключові слова: захист інформації, гешування, цілісність, конфіденційність, блокчейн, геш-функція, генератор ПВП.

ABSTRACT

Hrytsak A. Methods for effective cryptographic hash functions construction. – Manuscript.

Thesis for a Candidate of Technical Science degree in specialty 05.13.21 – Information security systems. – National Aviation University, Kyiv, 2020.

The dissertation is devoted to solving the actual scientific problem of developing and researching new effective hash functions that will provide the necessary level of the security with a sufficiently high speed.

The analysis of modern methods and algorithms for the construction and implementation of effective cryptographic hashing functions was carried out, which made it possible to identify their shortcomings and formalize the tasks of scientific research. Among the shortcomings identified are the vulnerabilities of known hash functions to cryptoanalytic attacks, low encryption speed and high requirements for computing facilities. It has allowed formalizing problem and defining task of research study.

A method of constructing hashing functions was developed, which made it possible to increase the speed of cryptographic data processing. Based on this method, three new stable hash functions have been developed, programmatically and investigated, which can increase the speed in 1.15-1.36 times and provide resistance to cryptanalytic attacks and can be used in systems for which the stability parameter is critical (this method was titled “secure method”).

A method of constructing hashing functions was developed, which made it possible to provide resistance to cryptanalytic attacks. Based on this method, three new high-speed hash functions have been developed, programmatically and investigated, which increase the speed in 1.16-1.53 times and can be applied to systems for which the speed parameter is critical (in the work this method was titled “high-speed method”).

The method of pseudorandom number generators construction has been improved, which allowed forming a statistically stable range for cryptographic applications. Based on this method, three pseudorandom number generators have been developed and implemented software that will be useful for both hashing functions and other cryptographic applications (key generation, stream ciphers, etc.). In addition, the developed Viriy pseudorandom number generators are faster than their counterparts;

The method of cryptographic protection of information has been improved, which by means of fixing information on user ID, session ID, time of sending, length of message and its serial number, as well as use of the new procedure of formation of session key and encryption, made it possible to ensure confidentiality and integrity of data in the modern information and communication systems and technologies.

Last chapter of the dissertation contains research study devoted to collision characteristics of proposed hash functions using so-called “baby versions” of hashing functions based on the existed experimental technique (relevant in cryptography).

Specialized software was developed in the form of console applications in C ++ programming language (Microsoft Visual Studio 2013 (Release Version)) and a technique that allowed us to conduct experiments and verify the proposed methods. the results of the dissertation are used in the educational process of Vinnytsa National Technical University (to increase the efficiency of training of specialists in the specialty 125 “Cybersecurity”) as well as in scientific process of National Aviation University and Educational & Research Complex “Information and Communication Systems”. It was confirmed by the acts of implementation.

Keywords: information security, hashing, integrity, privacy, blockchain, hash function, pseudo random number generator.