КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 2 (10), 2020
ISSN 2663 - 4023
DOI 10.28925/2663-4023.2020.10.2944
УДК 004.056.55

**Гальченко Андрій Віталійович**
аспірант
Запорізький національний університет, Запоріжжя, Україна
ORCID: 0000-0002-2258-9755
*Andream1993@ukr.net*

**Чопоров Сергій Вікторович**
кандидат технічних наук, доцент, старший викладач
Запорізький національний університет, Запоріжжя, Україна
ORCID: 0000-0001-5932-952X
*S.choporoff@znu.edu.ua*

# ВИКОРИСТАННЯ МЕТОДУ «РОЗДІЛЯЙ ТА ВОЛОДАРЮЙ» В АЛГОРИТМАХ ЗАПЕРЕЧУВАНОГО ШИФРУВАННЯ

**Анотація.** Поточне дослідження проведене авторами для перевірки гіпотези щодо можливості збільшення швидкості роботи алгоритмів заперечуваного шифрування. Вказане дослідження є актуальним, оскільки алгоритми заперечуваного шифрування використовують ефективні схеми перетворення для захисту як інформації, так і її користувачів. Разом з тим, структура алгоритмів заперечуваного шифрування досить складна та зосереджена. Це впливає на швидкість їх роботи та робить неможливим їх практичне застосування в галузях з обробки даних і захисту інформації. Основною метою дослідження є пошук методів і засобів, використання яких дозволить зменшити час виконання алгоритмів заперечуваного шифрування. В цій роботі було розглянуто та досліджено застосування методу «розділяй та володарюй». Вказаний метод був застосований до процедур обробки даних алгоритмів заперечуваного шифрування. Оскільки кінцеве рішення повинне бути універсальним для подальшого використання з іншими алгоритмами заперечуваного шифрування, то автори не вносили жодних змін у вихідні алгоритми шифрування. Для перевірки гіпотези автори провели серію експериментів. Першим об'єктом дослідження був алгоритм заперечуваного шифрування побудований на базі багатопотокових обчислень. Недоліки безпеки спричинені його використанням були виявлені та досліджені. Інший алгоритм ґрунтувався на використанні методу «розділяй та володарюй». Вказаний метод був імплементований в систему обробки даних першого алгоритму. Для перевірки ефективності обох алгоритмів в експериментах були використані реальні файли з публічними та секретними даними. Саме дослідження було проведене на стендовому обладнанні, яке імітує типове робоче місце користувача. Результати експериментів демонструють появу приросту у швидкості роботи вихідного алгоритму заперечуваного шифрування даних. Також вказані результати були перевірені з використанням ключів шифрування різного розміру. Отримані результати були порівняні з дослідженнями інших авторів. В кінцевому результаті гіпотеза авторів була підтверджена. Використання методу «розділяй та володарюй» призвело до значного приросту швидкодії алгоритмів заперечуваного шифрування даних.

**Ключові слова:** заперечуване шифрування; захист інформації; конфіденційні дані; метод розділяй та володарюй; несанкціонований доступ; примушування; продуктивність; шифр.

## 1. ВСТУП

Смартфони, персональні комп'ютери та інші пристрої обробки даних стали невід'ємною частиною життя кожної людини. Це призвело до суттєвого розширення інформаційного простору. Конфіденційна інформація, яка має таку цінність для

людини, безконтрольно поширюється від одного пристрою до іншого [1] , [2]. Саме цей аспект і став причиною появи третіх осіб, які зацікавлені в накопиченні та використанні цієї інформації в особистих цілях.

За останні роки розроблено безліч організаційних методик і засобів технічного захисту інформації, які унеможливлюють доступ до конфіденційних даних [3], [4]. Однак більшість власників інформації не надають належної уваги правилам кібербезпеки та не виконують їх в повній мірі [5].

Шифрування даних – це основний спосіб технічного захисту даних, який широко використовується усіма користувачами сьогодні. З цією метою спеціалістами галузі інформаційної безпеки та суміжних галузей розроблена велика кількість криптографічних алгоритмів, які виконую широкий набір завдань щодо комплексного захисту даних [6]. Вказані алгоритми гарантують надійність захисту завдяки їх обчислювальній стійкості та стійкості до алгоритмічних атак [7]. Вказані характеристики не дозволяють стороннім особам втручатися в роботу цих систем або отримати ключі доступу до інформації з обмеженим доступом.

Разом з тим, сучасний рівень розвитку комп'ютерної техніки дозволяє виконати криптографічний аналіз алгоритмів шифрування та виявити алгоритмічні недоліки в їх структурі [8], [9]. Використовуючи вказану інформацію та недбале ставлення до власної інформаційної безпеки, користувачі створюються умови для зменшення надійності криптографічних засобів захисту та витоку інформації з обмеженим доступом внаслідок цього.

Окрім того, завжди існує більш простий спосіб отримати доступ до необхідної інформації, а саме застосування примусу до користувачів вказаних технічних засобів. Подібний підхід дозволяє не лише зберегти кошти на придбання необхідного технічного обладнання, але й час на вивчення об'єкту несанкціонованого доступу та пошуку його вразливостей.

Враховуючи вищевказане вже кілька десятиліть проводяться дослідження в напрямку розробки алгоритмів заперечуваного шифрування даних [10] – [21]. Особливістю вказаних алгоритмів є те, що в їх основі лежить використання вже існуючих схем перетворення даних. В порівнянні з іншими алгоритмами шифрування, вони дозволяють захистити як інформацію в разі її витоку за межі контрольованої зони, так і користувачів цієї інформації від застосування примусу з боку третіх осіб. Разом з тим алгоритми заперечуваного шифрування не набули практичного застосування в галузі інформаційної безпеки через складності з правовим регулюванням щодо їх використання [22], [23], а також складність та зосередженість їх структури, які впливають на швидкість їх роботи [10].

**Постановка проблеми.** Авторами статті створена блочна модель шифрування даних, яка дозволяє застосовувати алгоритми заперечуваного шифрування для обробки файлів з даними [24]. На базі вказаної моделі створено багатопочний алгоритм шифрування даних, який демонструє досить високі показники швидкодії, в порівнянні з подібними алгоритмами [17] – [21].

Однак отримані показники швидкодії виявилися недостатніми для практичного використання алгоритмів цього класу. В даній роботі автори пропонують використати метод «розділяй та володарюй» [25] для поділу вхідних даних на окремі частини та їх подальшу обробку в багатопотоковому режимі. Запропонований рішення повинне збільшити швидкість роботи алгоритмів заперечуваного шифрування та зменшити витрати ресурсів робочих стацій, не змінюючи структури, особливостей механізмів і рівня захисту алгоритмів шифрування.

**Аналіз останніх досліджень і публікацій.** Починаючи з 80-х років та до сьогодні проводяться розробки та дослідження в напрямі заперечуваного шифрування даних. За цей час розроблено та досліджено значну кількість алгоритмів заперечуваного шифрування. Однак вихідні концепції цих алгоритмів включають досить велику кількість важких математичних операцій, оптимізація алгоритмів вирішення яких триває в теперішній час. Тому вихідні алгоритми так і не набули практичного застосування.

В 1984 році Шаффі Голдвассер і Синтія Мікалі представили роботу [10], в якій був продемонстрований прототип алгоритму імовірнісного шифрування даних. Їх розробка дозволяла виконувати розшифровку кількох імовірних варіантів даних з однієї шифрограми. Однак створена ними система не набула практичного застосування, оскільки шифрограми генеровані їх системою значно перевищували розміри вхідних даних.

Згодом, в 1997 році, Джуліан Ассандж і Вейнман розробили файлову систему Rubberhose, в якій був реалізований механізм заперечуваного шифрування даних [11]. Це була перша система, яка дозволяла відновлювати різні набори даних з однієї шифрограми без обмежень їх розміру. Надійність вказаної системи ґрунтувалася на використанні існуючих на той час симетричних алгоритмів шифрування. В основі перетворень даних був поділ дискового простору на частини та їх послідовне шифрування різними ключами, окремих для кожного набору даних. Це унеможливлювало виявлення будь-яких інших даних після розшифрування кожного нового розділу. Однак вказана розробка не набула поширення через неможливість її використання в мережі.

В тому ж році група дослідників, до якої увійшли Ран Канетті, Синтія Дворк, Монті Наор і Рафаїл Островський, представили алгоритм заперечуваного шифрування даних з публічним ключем [12]. Їх розробка передбачала шифрування окремих бітів даних за допомогою випадкових значень. Однак їх розробка не дозволяла виконувати обробку наборів даних великого розміру, а також захищала лише джерело даних, в окремих випадках. Враховуючи це їх алгоритм не був застосований на практиці.

Сюзана Ряжкова запропонувала схему заперечуваного шифрування даних, яка ґрунтувалася на перетвореннях протоколу шифрування RSA [13]. Її розробка передбачала обмін даними через мережу та в подальшому повинна була використовуватися в системах електронного голосування. Однак з зв’язку з високою надмірністю даних (105 біт службових даних на 1 біт інформації), які необхідно було передавати по відкритих каналах, вказаний алгоритм не був застосований на практиці.

У 2009 році індійський вчений Хамада Ібрахім використовуючи особливості степеневих порівнянь створив новий алгоритм заперечуваного шифрування [14], в повній мірі захищав чутливі дані від витоку. Але його алгоритм мав досить низькі показники продуктивності та не міг застосовуватися на практиці. Він створив інший алгоритм, який ґрунтувався на протоколі Mediated RSA PKI [26]. В основі даного протоколу лежала можливість поділу секретного ключа, що навіть у разі витоку одного з них не давало можливість отримати доступ до захищених даних і заперечувало факт їх існування. Для відновлення даних з шифрограми необхідно було отримувати частину ключа від довіреної сторони. Оскільки вказане могло призвести до витоку даних, то він використовував протокол [27], який ґрунтувався на протоколі RSA [28]. Однак під час досліджень алгоритму виявилося, що він не достатньо опрацював частину щодо заперечування даних. До того ж його алгоритм накладав обмеження на розмір даних,

які можна було зашифрувати. Тому цей алгоритм також не був застосований на практиці.

Того ж року китайські дослідники Jiang Qing Wang та Bo Meng, використовуючи вищевказані алгоритми та схему заперечуваного шифрування запропоновану Михайлом Клоновскі [15], розробили протокол електронного голосування BCP [16]. Вказаний протокол ґрунтувався на двох схемах генерації ключів та використовував схему перетворень даних Ель-Гамаля [29]. Вказаний протокол не мав вад із продуктивністю та був застосований на практиці. Од дослідження протоколу BCP виявило можливі проблеми з безпекою даних, тому їх розробка не набула широкого поширення в більшій частині мереж.

В 2013 році Микола Молдовян розробив декілька варіантів алгоритмів заперечуваного шифрування даних на базі розподілу секретних параметрів [17]. Згідно з його твердженнями надійність вказаних алгоритмів ґрунтувалася на специфіці розподілу ключів шифрування та використанні існуючих симетричних алгоритмів. Від так швидкість розроблених ним алгоритмів повинна була досягати 10 – 100 Мб/с. Однак жодних експериментальних даних, які б підтвердили чи спростували вказані твердження в його роботі відсутні.

Під час своїх досліджень в 2014 році Молдовян розробив алгоритм заперечуваного шифрування даних з публічним ключем, в основі якого лежить використання розширеної криптографічної схеми Рабіна [18]. Вказаний алгоритм досить схожий на алгоритм імовірнісного шифрування даних Шаффі Голдвассера та Синтії Мікалі [10]. Інформація щодо практичної реалізації або дослідження його швидкодії у відкритих джерелах відсутні. Однак його схожість на свого попередника дає можливість припустити, що вказаний алгоритм має подібні переваги та недоліки.

Того ж року він провів дослідження з побудови швидкісного алгоритму заперечуваного шифрування на основі існуючих блочних шифрів [19]. Ідея дослідження полягала в розробці алгоритму заперечуваного шифрування, який виконує шифрування публічного та секретного повідомлень за допомогою одного випадкового параметра та двох різних ключів одночасно. Вказаний підхід згідно його тверджень повинен забезпечити швидкість шифрування на рівні існуючих симетричних алгоритмів (2,602 – 3,904 Мб/с). Однак наведені ним дані щодо практичної перевірки цих тверджень демонструють, що кінцева швидкість складає 0,5 – 1,25 Кб/с. Вказані результати досить далекі від показників симетричних алгоритмів шифрування, тому вказані твердження не можуть бути застосовані на практиці.

Також відомо, що в 2016 року Микола Молдовян продемонстрував потоковий алгоритм заперечуваного шифрування [20]. Цей алгоритм дозволяв виконувати потокову обробку даних та був орієнтований на використання у мережі та був подібним до розробки 2013 року. Однак згідно його тверджень вказаний алгоритм має інші показники швидкодії, експериментальне підтвердження яких не було доведено.

Вже у 2018 році була опублікована його праця щодо можливості заперечуваного шифрування даних за допомогою степеневих порівнянь [21]. Основною ідеєю його підходу було використання порівнянь $N$-ї степені для шифрування даних та відновлення $N$-ї кількості вхідних даних. Вказана робота є досить перспективною з точки зору інформаційної безпеки. Разом з тим в одній із своїх праць він наголошував, що збільшення кількості шифрованих даних неминуче призводить до падіння швидкодії вихідного алгоритму заперечуваного шифрування даних. Експериментальне дослідження показників швидкодії вказаного алгоритму не проводилося.

Огляд вищевказаних алгоритмів заперечуваного шифрування продемонстрував наявність значної кількості варіацій механізмів заперечуваного шифрування даних і їх застосування. Схеми шифрування даних, які не ґрунтуються на використанні симетричних алгоритмів, мають суттєві обмеження швидкодії та розміру вхідних даних. Також встановлено, що не всі з вказаних алгоритмів були перевірені на предмет можливості практичної реалізації, дані щодо їх експериментальних досліджень відсутні.

**Мета статті.** Пошук засобів і методів збільшення продуктивності алгоритмів заперечуваного шифрування даних, що дозволить їх практичне застосування в галузі інформаційної безпеки.

## 2. МЕТОДИКА ДОСЛІДЖЕННЯ

Автори виконали аналіз алгоритмів заперечуваного шифрування, які знаходяться в публічному доступі [10] – [21], та запропонували власну ефективну модель блочного алгоритму заперечуваного шифрування даних.

Вихідна модель шифрування даних авторів вирішувала завдання щодо можливості обробки файлів з даними алгоритмами заперечуваного шифрування. До того ж метод запропонований авторами дозволяв використовувати будь-які з алгоритмів заперечуваного шифрування, незважаючи на різність їх структур та механізмів перетворення даних. Структурна схема вихідної моделі на ведена на рис. 1:

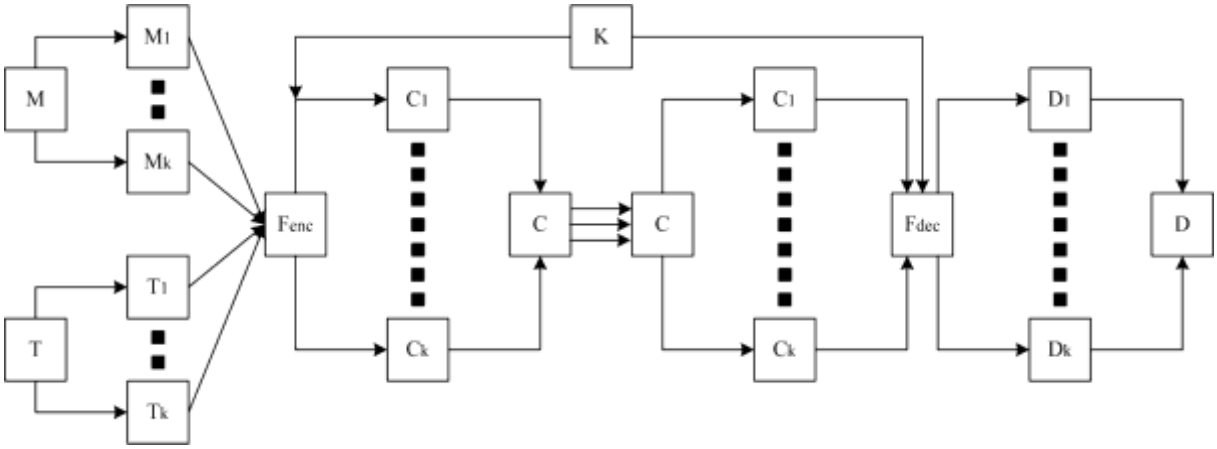



*Рис. 1 – Базова модель шифрування даних з використанням алгоритмів заперечуваного шифрування: K – ключ шифрування/дешифрування даних, M – файл с фіктивними даними, $M_{1...k}$ – набір блоків з фіктивними даними, $M_i$ – блок з фіктивними даними, T – файл з секретними даними, $T_{1...k}$ – набір блоків з секретними даними, $T_i$ – блок з секретними даними, $F_{enc}(...)$ – функція дешифрування даних, C – шифровані дані, $C_{1...k}$ – набір блоків з шифрованими даними, $C_i$ – блок з шифрованими даними, $F_{dec}(...)$ – функція дешифрування даних, D – дешифровані дані, $D_{1...k}$ – набір блоків з дешифрованими даними, $D_i$ – блок з дешифрованими даними*

Метод використаний у вищевказаній моделі був достатньо продуктивним, в порівнянні з подібними алгоритмами [17], [19]. Однак, його продуктивності досі не вистачало для практичного застосування, в порівнянні з блочними алгоритмами

шифрування [30]. Дослідивши базову модель автори виявили, що швидкість її роботи $V$ досить чутлива до розміру вхідних даних $\|M\|$ (1):

$$\begin{cases} \|M\| \to \infty \\ V \to \infty \end{cases} \tag{1}$$

З метою усунення виявленого недоліку автори використали багатопотокові обчислення [30], які дозволили скоротити кількість блоків з даними та загальний час функціонування моделі в цілому. Структурна з використанням багатопотокових обчислень наведена на рис. 2:

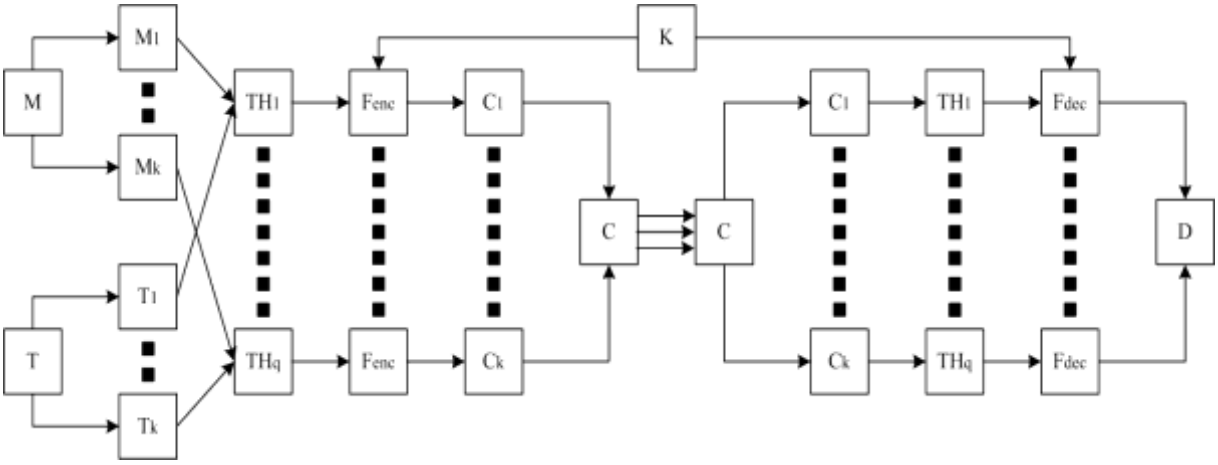



*Рис. 2 – Багатопотокова схема моделі шифрування даних з використанням алгоритмів заперечуваного шифрування: M – файл с фіктивними даними, $M_{1...k}$ – набір блоків з фіктивними даними, $M_i$ – блок з фіктивними даними, T – файл з секретними даними, $T_{1...k}$ – набір блоків з секретними даними, $T_i$ – блок з секретними даними, $F_{enc}(...)$ – функція дешифрування даних, C – шифровані дані, $C_{1...k}$ – набір блоків з шифрованими даними, $C_i$ – блок з шифрованими даними, $F_{dec}(...)$ – функція дешифрування даних, D – дешифровані дані, $D_{1...k}$ – набір блоків з дешифрованими даними, $D_i$ – блок з дешифрованими даними, $TH_{1...k}$ – потоки обробки даних, q – кількість потоків обробки даних, $th_i$ – поток обробки даних*

Під час проведення серії експериментів з вищевказаною моделлю автори підтвердили появу прискорення обчислень в процедурах шифрування та дешифрування даних. Однак отримані показники досі не могли порівнюватися з показниками існуючих блочних алгоритмів шифрування даних.

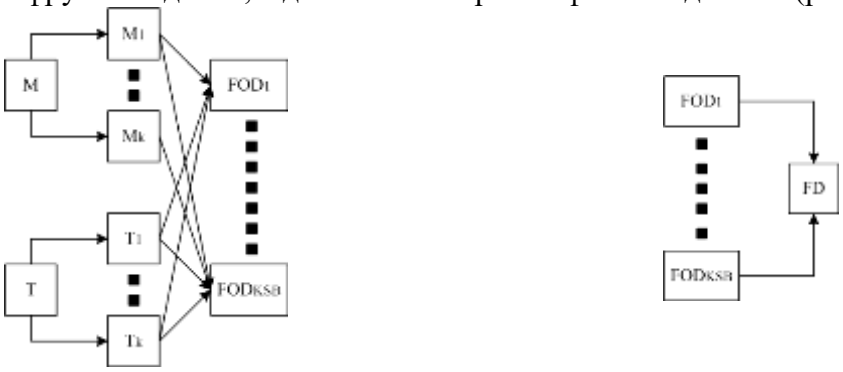
З метою усунення недоліків, які автори виявили в багатопотоковій моделі, та збільшення швидкості її роботи автори переглянули структуру базової моделі. Вони встановили, що значну частину часу функціонування моделі відводиться на введення та виведення даних. Багатопотокова моделі шифрування зменшує кількість блоків даних, які обробляються програмою за одиницю часу. Але для оптимальної роботи моделі необхідно виділити окремий потік обробки даних для кожного блоку даних, що не є можливим згідно з виразом (2):

$$\begin{cases} DBQ \to \infty \\ 1 < THQ \le GILQ \end{cases} \quad (2)$$

Де *DBQ* – кількість блоків з даними; *THQ* – кількість потоків обробки даних; *GILQ* – граничне значення кількості потоків обробки даних.

Кількість блоків *DBQ* з даними залежить від розміру буфера обміну даними *BS* та розміру ключа шифрування *KS*, тому для підвищення швидкості роботи багатопотокової моделі шифрування даних автори запропонували використати метод «розділяй та володарюй».

Вказаний метод є однією з парадигм програмування та дозволяє підвищувати швидкість роботи програм шляхом її розкладання на прості операції, які робоча станція виконує швидше. Однак ідея авторів полягає у застосуванні положень даного методу не до моделі шифрування даних, а до системи обробки файлів з даними (рис. 3):




А Б

*Рисунок 3 – Схеми декомпозиції (а) та відновлення (б) файлів системою обробки даних: M – файл с фіктивними даними, $M_{1...k}$ – набір блоків з фіктивними даними, $M_i$ – блок з фіктивними даними, T – файл з секретними даними, $T_{1...k}$ – набір блоків з секретними даними, $T_i$ – блок з секретними даними, $FOD_{1...KSB}$ – фрагменти файлу з даними, FD – файл з даними*

Згідно зі схемою рис. 3а, декомпозиція файлу з даними виконується згідно з алгоритмом 1:

1) Знаючи розміри файлів з публічними $\|M\|$ та секретними $\|T\|$ даними, обчислити максимальний розмір файлів *SMI* за допомогою виразу (3):

$$SMI = F_{\max}(\|M\|, \|T\|) \quad (3)$$

2) Знаючи розмір службових даних моделі $\|ADI\|$, обчислити початкову кількість блоків з даними *BQ*, вираз (4):

$$1 < BQ \le \left\lceil \frac{SMI}{KS - \|ADI\|} \right\rceil \quad (4)$$

3) Обрати кількість потоків обробки даних *KSB* згідно з умовою $1 < KSB \le GILQ$.

4) Розділити файл з даними *FD* на фрагменти *FOD* за допомого системи (5):

$$FOD_j = FD_{(i\ldots i+1)\cdot\frac{BQ\cdot(KS-\|ADI\|)}{KSB}} \quad (5)$$

Де $fod_j$ – фрагмент даних, $i$ – номер фрагменту, $j$ – номер фрагменту в файлі з даними.

Процедура відновлення файлу з даними (рис. 3б) передбачає виконання алгоритму 2:

1) Обчислити максимальний розмір вихідних даних *SMO*, вираз (6):

$$SMO = 2 \cdot KSB \cdot KS \quad (6)$$

2) Обчислити максимальну кількість блоків з даними *MBQ*, вираз (7):

$$MBQ = \frac{SMO}{2 \cdot BQ} \quad (7)$$

3) Згрупувати блоки даних та зберегти в окремі файли $fod_j = fod_{j,0} \,||\, fod_{j,1} \,||\, \ldots \,||\, fod_{j,MBQ-1}$.

4) Згрупувати окремі файли та зберегти в вихідний файл з даними $FD = FOD_0 \,||\, FOD_1 \,||\, \ldots \,||\, FOD_{KSB-1}$.

Після введення вказаних алгоритмів обробки даних в багатопотокову модель, вона зазнала змін, які наведені на рис. 4:

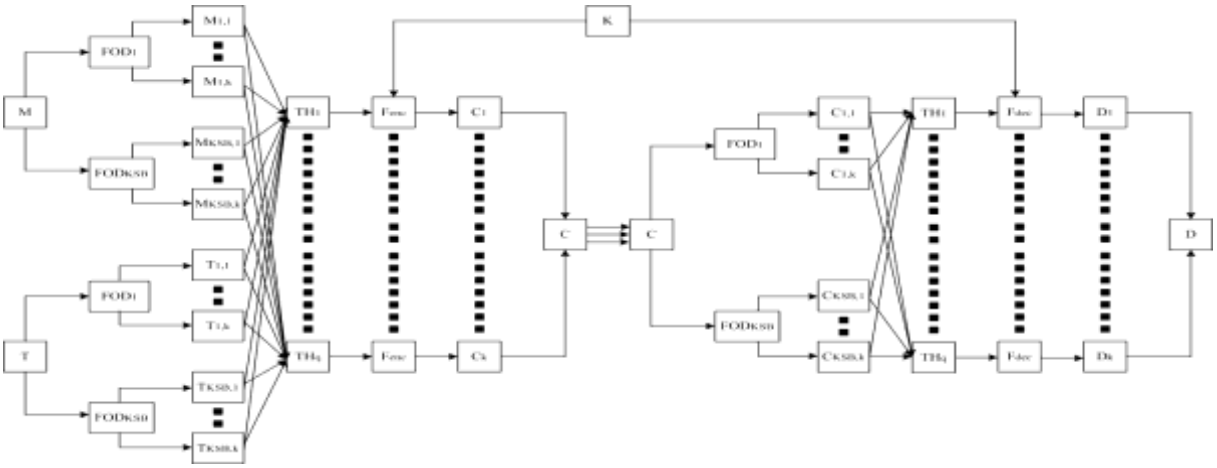



*Рис. 4 – Модифікована багатопотокова схема моделі шифрування даних з використанням алгоритмів заперечуваного шифрування: M – файл с фіктивними даними, T – файл з секретними даними, $FOD_{1\ldots KSB}$ – набір фрагментів даних, $M_{1,1\ldots k}$ – набір блоків з фіктивними даними, $M_{i,j}$ – блок з фіктивними даними, $T_{1,1\ldots k}$ – набір блоків з секретними даними, $T_{i,j}$ – блок з секретними даними, $TH_{1\ldots k}$ – потоки обробки даних, q – кількість потоків обробки даних, $th_i$ – поток обробки даних, $F_{enc}(\ldots)$ – функція дешифрування даних, C – шифровані дані, $C_{1\ldots k}$ – набір блоків з шифрованими даними, $C_i$ – блок з шифрованими даними, $F_{dec}(\ldots)$ – функція дешифрування даних, D – дешифровані дані, $D_{1\ldots k}$ – набір блоків з дешифрованими даними, $D_i$ – блок з дешифрованими даними*

Запропоновані авторами зміни не вносять суттєвих змін в ядро базової моделі, при цьому їх використання, в теорії, повинне збільшити швидкість роботи моделі в *KSB* разів.

Для перевірки істинності запропонованої гіпотези проведено серію експериментів з використанням підготовленого стендового обладнання, яке імітує роботу типової робочої станції:

1) Апаратне забезпечення – ЦП Intel(R) Core(TM) i5-8250, оперативна пам'ять DDR4 на 8 ГБ та жорсткий диск об'ємом 500 ГБ.

2) Програмне забезпечення – операційна система Windows 10 (x64), середовище програмування моделі Python IDLE 3.7. 3.

Крім того, враховуючи значну кількість обчислень, які виконуються моделлю, та час необхідний на їх виконання, для прискорення перевірки гіпотези автори встановили наступні обмеження для тестових даних:

1) Розмір файлів з даними ~15 МБ та файлів з шифрованими даними ~30 МБ;

2) Форматом тестових файлів обрано EXE (дозволяє перевірити валідність відновлених даних на етапі дешифрування без додаткових засобів);

3) Розміри тестових ключів шифрування 1024 біт (оптимальний з точки зору захисту даних) та 8192 біт (наближений до оптимального з точки зору кількості блоків з даними).

Встановлені авторами обмеження жодним чином не вплинуть на отримані результати, оскільки механізми перетворення даних модифікованої моделі не зазнали змін.

## 3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

В даному розділі наведені результати проведених експериментів, їх аналіз та порівняння з досягненнями інших авторів.

### 3.1 Результати експериментів

Результати проведених авторами експериментів наведені в таблицях 1-3. У вказаних таблицях автори використали наступні позначення: i – номер експерименту, TE1024 та TME1024 – час відведений на шифрування даних в вихідній і модифікованій моделях ключем 1024 біти, TE8192 та TME8192 – час відведений на шифрування даних в вихідній і модифікованій моделях ключем 8192 біти, TPD1024 та TMPD1024 – час відведений на дешифрування даних в вихідній і модифікованій моделях ключем 1024 біти, TPD8192 та TMPD8192 – час відведений на дешифрування даних в вихідній і модифікованій моделях ключем 8192 біти, маркери «min», «max» та «avg» – нижнє, верхнє та середнє значення швидкодії процедур шифрування/дешифрування даних, KE1024 та KE8192 – коефіцієнти прискорення процедур шифрування даних ключами 1024 та 8192 біти, KPD1024 та KPD8192 – коефіцієнти прискорення процедур дешифрування публічних даних ключами 1024 та 8192 біти, KSD1024 та KSD8192 – коефіцієнти прискорення процедур дешифрування секретних даних ключами 1024 та 8192 біти. У разі наявності прискорення значення коефіцієнтів KE1024, KPD1024, KSD1024, KE8192, KPD8192, KSD8192 знаходяться на проміжку *[0; 100]*, інакше на проміжку *[-∞; 0)*.

ISSN 2663 - 4023

*Таблиця 1*

**Дані щодо швидкості шифрування даних вихідної та модифікованої моделей**

| I | TE1024, c | TME1024, c | TE1024, c | TME1024, c |
|---|---|---|---|---|
| 1 | 4,705 | 6,405 | 11,268 | 8,108 |
| 2 | 4,387 | 6,655 | 11,784 | 6,874 |
| 3 | 4,252 | 7,155 | 11,908 | 8,179 |
| 4 | 4,055 | 7,108 | 11,938 | 7,889 |
| 5 | 4,572 | 6,389 | 11,838 | 7,326 |
| 6 | 4,370 | 6,608 | 11,688 | 7,873 |
| 7 | 4,230 | 7,303 | 11,952 | 7,873 |
| 8 | 4,181 | 7,264 | 11,936 | 7,624 |
| 9 | 4,360 | 7,108 | 11,853 | 7,998 |
| 10 | 4,268 | 6,936 | 11,820 | 6,874 |
| Min | 4,055 | 6,389 | 11,268 | 6,874 |
| Max | 4,705 | 7,303 | 11,952 | 8,179 |
| Avg | 4,338 | 6,893 | 11,799 | 7,662 |

*Таблиця 2*

**Дані швидкості дешифрування даних ключем 1024 біти**

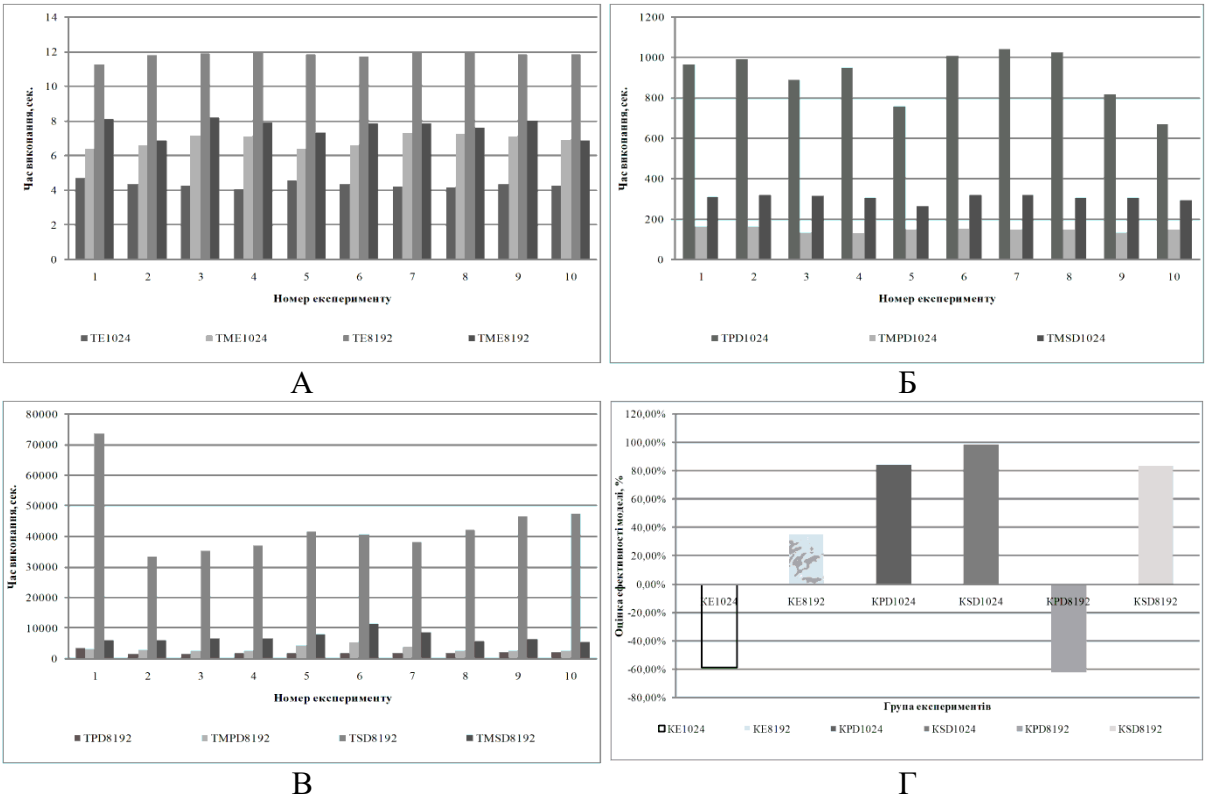
| I | TPD1024, c | TMPD1024, c | TSD1024, c | TMSD1024, c |
|---|---|---|---|---|
| 1 | 965,09 | 163,30 | 19504,40 | 309,98 |
| 2 | 991,12 | 162,60 | 20030,50 | 321,36 |
| 3 | 888,59 | 136,19 | 17958,30 | 317,68 |
| 4 | 946,76 | 130,24 | 19134,00 | 309,36 |
| 5 | 759,84 | 150,54 | 15356,20 | 265,57 |
| 6 | 1005,41 | 154,58 | 20319,30 | 318,98 |
| 7 | 1039,46 | 150,96 | 21007,50 | 317,91 |
| 8 | 1023,15 | 151,75 | 20677,80 | 309,03 |
| 9 | 815,20 | 135,23 | 16475,20 | 306,36 |
| 10 | 669,60 | 150,18 | 13532,60 | 295,74 |
| Min | 669,60 | 130,24 | 13532,60 | 265,57 |
| Max | 1039,46 | 163,30 | 21007,50 | 321,36 |
| Avg | 910,42 | 148,56 | 18399,58 | 307,20 |

*Таблиця 3*

**Дані щодо швидкості дешифрування даних ключем 8192 біти**

| I | TPD8192, c | TMPD8192, c | TSD8192, c | TMSD8192, c |
|---|---|---|---|---|
| 1 | 3641,80 | 309,98 | 73600,58 | 6265,39 |
| 2 | 1658,84 | 321,36 | 33525,22 | 6309,97 |
| 3 | 1744,55 | 317,68 | 35257,46 | 6761,95 |
| 4 | 1840,00 | 309,36 | 37186,41 | 6799,40 |
| 5 | 2050,14 | 265,57 | 41433,45 | 8034,68 |
| 6 | 2012,41 | 318,98 | 40670,80 | 11429,71 |
| 7 | 1894,31 | 317,91 | 38284,02 | 8566,11 |
| 8 | 2078,19 | 309,03 | 42000,38 | 5799,39 |
| 9 | 2299,06 | 306,36 | 46464,00 | 6437,79 |
| 10 | 2343,82 | 295,74 | 47368,68 | 5688,39 |
| Min | 1658,84 | 265,57 | 33525,22 | 5688,39 |
| Max | 3641,80 | 321,36 | 73600,58 | 11429,71 |
| Avg | 2156,31 | 307,20 | 43579,10 | 7209,28 |

Для можливості надання об'єктивної оцінки отриманим результатам вони були представлені в графічному вигляді (рис. 5):




А Б

В Г

*Рис. 5 – Оцінка витрат часу на шифрування та дешифрування даних: а – шифрування ключами 1024 та 8192 біти; б – дешифрування ключем 1024 біти; в – дешифрування ключем 8192 біти; г – ефективність шифрування/дешифрування*

### 3.2 Обговорення результатів

За результатами проведених експериментів встановлено можливість прискорення блочної моделі заперечуваного шифрування даних шляхом декомпозиції файлів з даними. Однак вказане прискорення можливо отримати лише за певних умов.

Під час аналізу експериментальних даних автори виявили падіння швидкості на етапі шифрування даних ключем 1024 біти. Аналіз даних таблиці 1 та структурний аналіз моделі показали, що причиною падіння продуктивності є внесені авторами модифікації в процедури обробки даних перед їх шифруванням. Більш глибокий аналіз показав, що процедури шифрування в алгоритмах заперечуваного шифрування даних включають в себе незначну кількість легких обчислень, які виконуються більшістю робочих станцій за оптимальний час та не потребують прискорення. До того ж паралельне обчислення шифрограми у вихідній багато потоковій моделі вже зменшило кількість блоків з даними вхідних файлів. Тому подальше прискорення процедури шифрування можливе лише шляхом збільшення *BS* між системою та програмою, який залежить від *KS*.

Вищевказане твердження знаходить своє підтвердження згідно з даними табл. 1. Дані отримані в результаті шифрування файлів ключем 8192 біти модифікованою

багатопоточною моделлю шифрування даних свідчать про незначний приріст швидкості під час шифрування блоків з даними. Вказане підтверджує гіпотезу авторів та доводить ефективність запропонованих ними модифікацій для прискорення роботи вихідної моделі. Однак в порівнянні зі швидкістю шифрування даних ключем 1024 біти, отримані показники швидкодії свідчать про неефективність подібного рішення. Від так модифікована модель шифрування даних дає приріст у швидкості шифрування даних лише у випадку використання більш довшого ключа шифрування.

Досить критичною для алгоритмів заперечуваного шифрування даних є час відновлення (дешифрування) даних, оскільки в подібних алгоритмах використовуються важкі обчислення за одиницю машинного часу. Саме вказана обставина впливає на продуктивність існуючих алгоритмів заперечуваного шифрування та не дозволяє їх практичне використання в галузі інформаційної безпеки.

Разом з тим дані таблиці 2 та графіків (рис. 5) свідчать, що запропоновані авторами модифікації призвели до значного прискорення роботи тестової моделі на етапі дешифрування даних. Середнє значення прискорення процедури дешифрування становить 6-8 разів для відновлення даних, в порівнянні з показниками продуктивності вихідної багатопотокової моделі алгоритмів заперечуваного шифрування.

Так під час дешифрування публічних та секретних даних ключем 1024 біти, автори отримали значне прискорення роботи тестового алгоритму заперечуваного шифрування. Вказане пояснюється вибором оптимального з точки зору обчислень секретного ключа та розміру буфера обробки даних. Скорочення кількості блоків за рахунок розкладання файлів на частини та їх наступне обробка в паралельному режимі потребували витрат часу, але вони стали меншими за тривалість дешифрування даних вихідної моделі.

Також автори перевірили можливість покращення показників швидкодії дешифрування даних використовуючи для обчислень більш довгий ключ (8192 біти). Вказаний підхід повинен був значно скоротити кількість блоків з даними та в сукупності з іншими модифікаціями прискорити роботу алгоритму.

Однак результати експериментів свідчать про наявність незначного прискорення в процедурі дешифрування. Від так отриманий результат свідчить про неможливість отримання приросту швидкодії шляхом збільшення розміру ключа в модифікованій моделі шифрування даних.

Для об’єктивної оцінки отриманих результатів виконано порівняння отриманих результатів прискорення моделі з даними швидкодії подібних алгоритмів [19] (таблиця 4):

*Таблиця 4*

**Порівняння результатів прискорення моделі з існуючими алгоритмами**

| Предмет оцінки | Продуктивність алгоритмів (Мб/с) | | | | |
|---|---|---|---|---|---|
| Об’єкти оцінки | Симетричні алгоритми | Алгоритм Молдовяна | Модель авторів | | |
| Процедури | Шифрування | Шифрування | Шифрування | Дешифрування | |
| | | | | Публічні | Секретні |
| $V_{min}$ | 2,602 | 0,001 | 16,628 | 0,423 | 0,012 |
| $V_{max}$ | 3,904 | 0,010 | 21,287 | 1,044 | 0,423 |

Таким чином, вищевказані показники прискорення в порівнянні з дослідженнями інших авторів є кращими (прискорення до 42 разів) та підтверджують гіпотезу авторів. Разом з тим автори зазначають, що для виключення випадків отримання негативних показників прискорення необхідно комбінувати використання вихідної та модифікованої моделей шифрування даних.

## 4. ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

В даній роботі автори запропонували та дослідили метод, який дозволяє частково вирішити проблеми продуктивності в алгоритмах заперечуваного шифрування даних.

Наукова новизна роботи полягає не в розробці нового методу шифрування даних, а дослідженні існуючих алгоритмів і пошуку засобів для їх практичного використання. Використання методу «розділяй та володарюй» в системі обробки даних вказаних алгоритмів дозволило значно підвищити продуктивність алгоритмів заперечуваного шифрування. Також встановлено, що особливості окремих алгоритмів заперечуваного шифрування не впливають на кінцевий результат, що робить метод універсальним.

Практичне значення отриманих результатів полягає в застосуванні методів динамічного програмування для реалізації алгоритмів заперечуваного шифрування, розробці та імплементації програмного забезпечення для автоматичної декомпозиції і відновлення файлів в базову блочну модель заперечуваного шифрування даних. Результати проведених експериментів свідчать про зростання продуктивності вихідних алгоритмів заперечуваного шифрування даних, попри незначні витрати часу на обробку даних. Також досліджено залежність коефіцієнту зростання продуктивності вихідних алгоритмів заперечуваного шифрування даних від розміру ключа шифрування та обрано оптимальний розмір ключа для забезпечення максимальної продуктивності процедур шифрування/дешифрування даних.

Перспективами для подальших досліджень є пошук подібних способів маніпуляції даними, оскільки подібний підхід дозволив суттєво вплинути на продуктивність вихідних алгоритмів заперечуваного шифрування даних, розробка та дослідження захищеності розділеної системи заперечуваного шифрування даних для вирішення проблем практичного застосування алгоритмів заперечуваного шифрування.

## ПОДЯКА

Дослідження та експерименти проводилися в межах держбюджетної теми «Розробка математичного забезпечення для інженерного аналізу об’єктів аерокосмічної техніки на базі хмарних технологій» (№ держреєстрації 0117U007204).

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1] Грошева Е. К., Невмержицкий П. И. Информационная безопасность: Современные реалии. Бизнес-образование в экономике знаний. 2017. Т. 3. С. 35—38.

[2] Miloslavskaya N. G., Tolstoy A. I. Internet of Things: information security challenges and solutions. Cluster Computing. 2018. Vol. 22. P. 103—119.

[3] ДСТУ ISO/IEC 27001:2015. Інформаційні технології. Методи захисту системи управління інформаційною безпекою. Чинний від 2015-12-18. Вид. Офіц. Київ: ДП «укрнднц», 2016. 28 с.

[4] Хорев П. Б. Методы и средства защиты информации в компьютерных системах: Учеб. Пособие для студ. Высш. Учеб. Заведений. Москва: Издательский центр «Академия», 2005. 256 с.
[5] Развитие информационных угроз в первом квартале 2020 года. Статистика / В. Чебышев та ін. URL: https://securelist.ru/it-threat-evolution-q1-2020-statistics/96202 (дата звернення: 01.08.2020).
[6] Barker E. Guideline for Using Cryptographic Standards in the Federal Government: Cryptographic Mechanisms. Gaithersburg: NIST, 2020. 82 p.
[7] Kölbl S. Design and analysis of cryptographic algorithms: Ph.D Thesis. Lyngby, 2017. 273 p.
[8] Genkin D., Shamir A., Tromer E. Acoustic Cryptanalysis. Journal of Cryptology. 2017. Vol. 30. P. 392—443.
[9] Posch M. Quantum computing and the end of encryption. Hackaday. URL: https://hackaday.com/2020/06/11/quantum-computing-and-the-end-of-encryption (date of access: 01.08.2020).
[10] Goldwasser S., Micali C. Probabilistic encryption. Journal of Computer and System Sciences. 1984. Vol. 28. P. 277—299.
[11] Assange J., Weinmann R. Rubberhose filesystem. Rubberhose. URL: https://web.archive.org/web/20110628084231/http://iq.org/~proff/rubberhose.org (date of access: 01.08.2020).
[12] Deniable Encryption / R. Canetti et al. Advances in Cryptology – CRYPTO. 1997. P. 90—104.
[13] Rjazhkova Z. Electronic Voting Schemes: Master thesis. Bratislava, 2002. 64 p.
[14] Ibrahim H. Receiver-Deniable Public-Key Encryption. International Journal of Network Security. 2009. Vol. 8, no. 2. P. 159—165.
[15] Klonowski M., Kubiak P., Kutyłowsk M. Practical Deniable Encryption. SOFSEM 2008: 34th Conference on Current Trends in Theory and Practice of Computer Science, Nový Smokovec, Jan 19, 2008. 2008. P. 599—609.
[16] Meng B., Wang J. Q. A Receiver Deniable Encryption Scheme. International Symposium on Information Processing: ISIP09. 2009. P. 254—257.
[17] Морозова Е. В., Мондикова Я. А., Молдовян Н. А. Способы отрицаемого шифрования с разделяемым ключем. Информационно-управляющие системы. 2013. № 6. С. 73—78.
[18] Молдовян Н. А., Вайчикаускас М. А. Расширение криптосхемы Рабина: алгоритм отрицаемого шифрования по открытому ключу. Вопросы защиты информации. 2014. № 2. С. 12—16.
[19] Молдовян Н. А., Биричевский А. Р., Мондикова Я. А. Отрицаемое шифрование на основе блочных шифров. Информационно-управляющие системы. 2014. № 5. С. 80—86.
[20] Stream Deniable-Encryption Algorithms / N. A. Moldovyan et al. Computer Science Journal of Moldova. 2016. Vol. 24, no. 1. P. 68—82.
[21] Молдовян Н. А., Вайчикаускас М. А. Генерация степенных сравнений как способ открытого шифрования и протокол отрицаемого шифрования. Интеллектуальные технологии на транспорте. 2018. № 1. С. 25—30.
[22] Кримінальний процесуальний кодекс України: від 18.04.2010 р. № 4651-6: станом на 13 серп. 2020 р. Київ, 2010. URL: https://zakon.rada.gov.ua/laws/show/4651-17#Text (дата звернення: 14.08.2020).
[23] Кобылянский А. Как Apple из-за ФБР открыла СБУ и Нацполиции путь к данным украинцев. Ліга.Tech. URL: https://tech.liga.net/technology/article/perelomnyy-moment-apple-otkazyvaetsya-ot-shifrovaniya-dannyh-polzovateley-chego-nam-jdat (дата звернення: 01.08.2020).
[24] Гальченко А. В., Чопоров С. В. Заперечуване шифрування на основі застосування підходу гібридних криптографічних систем. Радіоелектроніка, інформатика, управління. 2019. № 1. С. 178—191.
[25] Алгоритмы: построение и анализ / Т. Х. Кормен та ін. 3-тє вид. Москва: Вильямс, 2020. 1328 с.
[26] A Method for Fast Revocation of Public Key Certificates and Security Capabilities / D. Boneh et al. USENIX: 10th USENIX Security Symposium, Washington, Aug 13, 2001. 2001. P. 1—13.
[27] Chou T., Orlandi C. The Simplest Protocol for Oblivious Transfer. LATINCRYPT 2015: 4th International Conference on Cryptology and Information Security, Guadalajara, Aug 23, 2015. 2015. P. 40—58.
[28] Sirajuddin A. The RSA Algorithm. 2019. P. 1—23.
[29] Abdul-Hassan M. S., Irtefaa A. N. Modification of elgamal Cryptosystem using Statistical Methods. European Journal of Scientific Research. 2015. Vol. 133, no. 1. P. 20—25.
[30] Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. Москва: Триумф, 2012. 816 с.

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 2 (10), 2020
ISSN 2663 - 4023

**Andrii Halchenko**
Phd
Department of Software Engineering, Zaporizhzhya National University, Zaporizhzhya, Ukraine
ORCID: 0000-0002-2258-9755
*Andream1993@ukr.net*

**Sergiy Choporov**
Dr. Sc., Senior lecturer
Department of Software Engineering, Zaporizhzhya National University, Zaporizhzhya, Ukraine
ORCID: 0000-0001-5932-952X
*S.choporoff@znu.edu.ua*

# THE DIVIDE AND CONQUER METHOD IN THE DENIABLE ENCRYPTION ALGORITHMS

**Abstract.** The deniable encryption algorithms productivity increasing is investigated in this paper. This investigation is relevant because of effective schemes for information and its users protection. But these algorithms is very complex and lumped. It really affects them. That's why deniable encryption algorithms have not been widespread in data processing and information security systems. The execution time reducing methods and tools exploration is the main goal of this work. The divide and conquer method has been discussed and investigated in this paper. It has been implemented into the data processing system of the deniable encryption algorithms. Nothing modifies have been implemented into the base algorithm. It allows to make it universal and apply to other deniable encryption algorithms. The series of experiments have been completed by authors to verify the hypothesis. The base deniable encryption algorithm discussing is the first stage of investigation. Its vulnerabilities have been found and investigated. Another algorithm is based on the divide and conquer method applying. It has been implemented into the modified data processing system. The both algorithms efficiency has been investigated by the experiments with the real with public and secret information files. The experiments have been completed on the prepared equipment. This equipment simulates the user's workplace with real hardware and software. According to the results the deniable encryption algorithms productivity has been reached by the divide and rule method. Also the method has been verified by the different size encryption keys. The base deniable encryption algorithms have not been modified. The results have been compared with other authors' investigations. In the end authors' hypothesis has been proved. But some restrictions of this results reaching have been set by the authors.

**Keywords: d**eniable encryption; information security; sensitive data; the divide and conquer method; unauthorized access; coercion; productivity; cipher.

## REFERENCES (TRANSLATED AND TRANSLITERATED)

[1] Grosheva E. K., Nevmerzhickij P. I. Informacionnaja bezopasnost': Sovremennye realii. Biznes-obrazovanie v jekonomike znanij. 2017. T. 3. S. 35—38.
[2] Miloslavskaya N. G., Tolstoy A. I. Internet of Things: information security challenges and solutions. Cluster Computing. 2018. Vol. 22. P. 103—119.
[3] DSTU ISO/IEC 27001:2015. Informatsiini tekhnolohii. Metody zakhystu cystemy upravlinnia informatsiinoiu bezpekoiu. Chynnyi vid 2015-12-18. Vyd. Ofits. Kyiv: DP «ukrndnts», 2016. 28 s.
[4] Horev P. B. Metody i sredstva zashhity informacii v komp'juternyh sistemah: Ucheb. Posobie dlja stud. Vyssh. Ucheb. Zavedenij. Moskva: Izdatel'skij centr «Akademija», 2005. 256 s.
[5] Razvitie informacionnyh ugroz v pervom kvartale 2020 goda. Statistika / V. Chebyshev ta in. URL: https://securelist.ru/it-threat-evolution-q1-2020-statistics/96202 (data zvernennja: 01.08.2020).
[6] Barker E. Guideline for Using Cryptographic Standards in the Federal Government: Cryptographic Mechanisms. Gaithersburg: NIST, 2020. 82 p.
[7] Kölbl S. Design and analysis of cryptographic algorithms: Ph.D Thesis. Lyngby, 2017. 273 p.
[8] Genkin D., Shamir A., Tromer E. Acoustic Cryptanalysis. Journal of Cryptology. 2017. Vol. 30. P. 392—443.

[9] Posch M. Quantum computing and the end of encryption. Hackaday. URL: https://hackaday.com/2020/06/11/quantum-computing-and-the-end-of-encryption (date of access: 01.08.2020).
[10] Goldwasser S., Micali C. Probabilistic encryption. Journal of Computer and System Sciences. 1984. Vol. 28. P. 277—299.
[11] Assange J., Weinmann R. Rubberhose filesystem. Rubberhose. URL: https://web.archive.org/web/20110628084231/http://iq.org/~proff/rubberhose.org (date of access: 01.08.2020).
[12] Deniable Encryption / R. Canetti et al. Advances in Cryptology – CRYPTO. 1997. P. 90—104.
[13] Rjazhkova Z. Electronic Voting Schemes: Master thesis. Bratislava, 2002. 64 p.
[14] Ibrahim H. Receiver-Deniable Public-Key Encryption. International Journal of Network Security. 2009. Vol. 8, no. 2. P. 159—165.
[15] Klonowski M., Kubiak P., Kutyłowsk M. Practical Deniable Encryption. SOFSEM 2008: 34th Conference on Current Trends in Theory and Practice of Computer Science, Nový Smokovec, Jan 19, 2008. 2008. P. 599—609.
[16] Meng B., Wang J. Q. A Receiver Deniable Encryption Scheme. International Symposium on Information Processing: ISIP09. 2009. P. 254—257.
[17] Morozova E. V., Mondikova Ja. A., Moldovjan N. A. Sposoby otricaemogo shifrovanija s razdeljaemym kljuchem. Informacionno-upravljajushhie sistemy. 2013. # 6. S. 73—78.
[18] Moldovjan N. A., Vajchikauskas M. A. Rasshirenie kriptoshemy Rabina: algoritm otricaemogo shifrovanija po otkrytomu kljuchu. Voprosy zashhity informacii. 2014. # 2. S. 12—16.
[19] Moldovjan N. A., Birichevskij A. R., Mondikova Ja. A. Otricaemoe shifrovanie na osnove blochnyh shifrov. Informacionno-upravljajushhie sistemy. 2014. # 5. S. 80—86.
[20] Stream Deniable-Encryption Algorithms / N. A. Moldovyan et al. Computer Science Journal of Moldova. 2016. Vol. 24, no. 1. P. 68—82.
[21] Moldovjan N. A., Vajchikauskas M. A. Generacija stepennyh sravnenij kak sposob otkrytogo shifrovanija i protokol otricaemogo shifrovanija. Intellektual'nye tehnologii na transporte. 2018. # 1. S. 25—30.
[22] Kryminalnyi protsesualnyi kodeks Ukrainy: vid 18.04.2010 r. № 4651-6: stanom na 13 serp. 2020 r. Kyiv, 2010. URL: https://zakon.rada.gov.ua/laws/show/4651-17#Text (data zvernennia: 14.08.2020).
[23] Kobyljanskij A. Kak Apple iz-za FBR otkryla SBU i Nacpolicii put' k dannym ukraincev. Liga.Tech. URL: https://tech.liga.net/technology/article/perelomnyy-moment-apple-otkazyvaetsya-ot-shifrovaniya-dannyh-polzovateley-chego-nam-jdat (data zvernennja: 01.08.2020).
[24] Halchenko A. V., Choporov S. V. Zaperechuvane shyfruvannia na osnovi zastosuvannia pidkhodu hibrydnykh kryptohrafichnykh system. Radioelektronika, informatyka, upravlinnia. 2019. № 1. S. 178—191.
[25] Algoritmy: postroenie i analiz / T. H. Kormen ta in. 3-tє vid. Moskva: Vil'jams, 2020. 1328 s.
[26] A Method for Fast Revocation of Public Key Certificates and Security Capabilities / D. Boneh et al. USENIX: 10th USENIX Security Symposium, Washington, Aug 13, 2001. 2001. P. 1—13.
[27] Chou T., Orlandi C. The Simplest Protocol for Oblivious Transfer. LATINCRYPT 2015: 4th International Conference on Cryptology and Information Security, Guadalajara, Aug 23, 2015. 2015. P. 40—58.
[28] Sirajuddin A. The RSA Algorithm. 2019. P. 1—23.
[29] Abdul-Hassan M. S., Irtefaa A. N. Modification of elgamal Cryptosystem using Statistical Methods. European Journal of Scientific Research. 2015. Vol. 133, no. 1. P. 20—25.
[30] Shnajer B. Prikladnaja kriptografija. Protokoly, algoritmy, ishodnye teksty na jazyke Si. Moskva: Triumf, 2012. 816 s.