УДК 004.056.55

ЗАПЕРЕЧУВАНЕ ШИФРУВАННЯ НА ОСНОВІ ЗАСТОСУВАННЯ ПІДХОДУ ГІБРИДНИХ КРИПТОГРАФІЧНИХ СИСТЕМ

Гальченко А. В. – професіонал з організації інформаційної безпеки Казенного підприємства «Науково-виробничого комплексу «Іскра», м. Запоріжжя, Україна.

Чопоров С. В. – канд. техн. наук, доцент кафедри «Програмна інженерія» Запорізького національного університету, м. Запоріжжя, Україна.

АНОТАЦІЯ

Актуальність. Несанкціонований доступ до добре захищених інформаційно-телекомунікаційних систем є досить актуальною проблемою в галузі інформаційної безпеки [1]. Для вирішення цієї проблеми запропоновано використання механізмів заперечуваного шифрування, які в разі отримання несанкціонованого доступу до інформації дозволяють її розпорядникам як заперечити факт існування даних, так і забезпечити їх конфіденційність та захистити розпорядників інформації від застосування грубої сили з боку зловмисників для отримання ключової інформації. В статті викладено підхід до застосування існуючих алгоритмів заперечуваного шифрування для захисту інформації в великих масивах даних.

Мета. Основна мета дослідження полягає в перевірці гіпотези щодо можливості використання алгоритмів заперечуваного шифрування для роботи з великими масивами даних, оскільки всі алгоритми даного напряму є асиметричними та не адаптовані для роботи з «big data».

Метод. Перевірка гіпотези здійснюється шляхом введення додаткових блоків обробки даних у вихідний алгоритм заперечуваного шифрування з відкритим ключем на основі розширеної криптографічної схеми Рабіна [2], структура та особливості якого найбільш підходять для перевірки висунутої автором гіпотези.

Результати. За результатами експериментів авторами запропоновано прототип алгоритму заперечуваного шифрування, який реалізує блочне шифрування даних, а також зберігає особливості механізму двозначності вихідного алгоритму шифрування. Окрім того, запропоновані авторами зміни забезпечують збільшення продуктивності роботи запропонованого алгоритму, при реалізації певних обчислень, в порівнянні з існуючими підходами [3–6].

Висновки. Авторами вирішено задачу застосування існуючих алгоритмів заперечуваного шифрування для захисту інформації в великих масивах даних, на прикладі алгоритму заперечуваного шифрування з відкритим ключем на основі розширеної криптографічної схеми Рабіна. Запропонований підхід до побудови гібридного алгоритму з механізмом заперечування демонструє не лише збереження основних властивостей базового алгоритму, але й можливість блочного шифрування даних будь-якого розміру з гарними показниками продуктивності. Тобто запропонований алгоритм не лише дозволяє вирішити задачу забезпечення конфіденційності даних, в разі несанкціонованого доступу до них, але й робить його придатним для практичного застосування.

КЛЮЧОВІ СЛОВА: блочне шифрування, груба сила, заперечуване шифрування, інформаційно-телекомунікаційна система, неоднозначність, несанкціонований доступ, обробка даних, псевдоімовірність, публічні дані, розпізнавання, розширена криптографічна схема Рабіна, секретні дані, статичні дані.

АБРЕВІАТУРИ

IDLE – Integrated DeveLopment Environment;
PC – Personal Computer;
ЕОМ – електронно-обчислювальна машина;
ІТС – інформаційно-телекомунікаційна система;
КБ – кілобайт;
МБ – мегабайт;
МБ/сек – мегабайт в секунду;
НСД – несанкціонований доступ;
ОЗП – оперативний запам'ятовуючий пристрій;
ОС – операційна система;
ЦП – центральний процесор;
сек – секунда.

НОМЕНКЛАТУРА

$\wedge$ – логічна операція «І»;
$\rightarrow$ – напрямок дії;
∞ – нескінченість (для додатних чисел);
$||$ – операція склеювання (конкатенація);
$\div$ – діапазон числових значень;
$\|\ldots\|$ – розмір змінної в бітах;
$(p;q)$ – секретний ключ дешифрування;
A – ліва частина криптограми (алгоритм 2);
B – права частина криптограми (алгоритм 2);
C – криптограма в алгоритмах 2–4;
$CRsize$ – реальний розмір криптограми;
$CTsize$ – теоретичний розмір криптограми;
D – дешифровані дані в алгоритмах 3 і 4;
D_{K_T} – позначення функції дешифрування в блочному алгоритмі Молдовяна М. О.;
E_{K_M} – позначення функції шифрування в блочному алгоритмі Молдовяна М. О.;
$K0016$ – умовне позначення 16-байтових ключів;
$K0032$ – умовне позначення 32-байтових ключів;
$K0064$ – умовне позначення 64-байтових ключів;
$K0128$ – умовне позначення 128-байтових ключів;
$K0256$ – умовне позначення 256-байтових ключів;
$K0512$ – умовне позначення 512-байтових ключів;
K_3 – позначення секретного параметру третьої сторони;

DOI 10.15588/1607-3274-2019-1-17

K_M – умовне позначення публічного ключа в блочному алгоритмі Молдовяна М.О.;

K_T – умовне позначення секретного ключа в блочному алгоритмі Молдовяна М.О.;

M – вихідні публічні дані;

M' – дешифровані публічні дані;

$MBM\_A\lg orithm$ – умовне позначення блочного алгоритму Молдовяна М. О.;

$MPdec$ – позначення витрат пам'яті в алгоритмі 3;

$MSdec$ – позначення витрат пам'яті в алгоритмі 4;

$MidBlocksValue$ – середній кількість блоків;

$MidDsize$ – середній розмір вхідних даних;

P_Enc – умовне позначення алгоритму 2;

P_Pdec – умовне позначення алгоритму 3;

P_Sdec – умовне позначення алгоритму 4;

N – відкритий ключ шифрування в прототипі алгоритму;

R – блок випадкових цілих чисел (алгоритм 2);

R' – блок випадкових цілих чисел (алгоритм 3);

T – вихідні секретні дані;

T' – дешифровані секретні дані;

$Tdec$ – позначення витрат часу в алгоритмах 3 і 4;

$Tenc$ – позначення витрат часу в алгоритмі 2;

$TPdec$ – позначення витрат часу в алгоритмі 3;

$TSdec$ – позначення витрат часу в алгоритмі 4;

$VPdec$ – швидкість виконання алгоритму 3;

$VSdec$ – швидкість виконання алгоритму 4;

$Venc$ – швидкість виконання алгоритму 2;

$Vdec$ – швидкість виконання алгоритмів 3 і 4;

X – загальне позначення вихідних даних;

X' – загальне позначення дешифрованих даних;

Y – загальне позначення криптограми;

Z – блок випадкових цілих чисел в алгоритмі 4;

a_k – блок лівої частини криптограми в алгоритмах 2–4;

b_k – блок правої частини криптограми в алгоритмах 2–4;

c_k – блок криптограми в алгоритмах 2–4;

d_k – дешифровані дані в алгоритмах 3 і 4;

f_1 – загальне позначення функції шифрування;

f_2 – загальне позначення функції дешифрування;

f_{AN} – позначення функції аналізу даних в алгоритмах 3 і 4;

f_D – позначення функції дешифрування в алгоритмах 3 і 4;

f_E – позначення функції шифрування в алгоритмі 2;

f_H – позначення функції обчислення контрольної суми в алгоритмах 2–4;

f_M – позначення функції додавання мітки в алгоритмі 2;

f'_M – позначення функції перевірки мітки в алгоритмах 3 і 4;

f_R – позначення функції обчислення випадкових значень в алгоритмі 2;

f_{SB} – позначення функції перестановки байтів в алгоритмах 2;

f'_{SB} – позначення функції зворотної перестановки байтів в алгоритмах 3 і 4;

f'_T – позначення функції перетворення числових значень в дані в алгоритмах 3 і 4;

f_Z – позначення функції обчислення коренів;

m_k – блок вихідних публічних даних;

m'_k – блок дешифрованих публічних даних;

r_k – випадкове ціле число в алгоритмі 2;

r'_k – випадкове ціле число в алгоритмі 3;

t – умовне позначення часу виконання алгоритмів;

t_k – блок вихідних секретних даних в алгоритмах 2–4;

t'_k – блок дешифрованих секретних даних в алгоритмі 4;

u – розмір блоку даних в бітах;

z_k – випадкове ціле число в алгоритмі 4.

ВСТУП

Кожний клієнт або сервер, а особливо ті, що мають підключення до мережі Інтернет, є джерелом інформації. Тому вони є об'єктами для НСД для зловмисників, однією з основних задач яких є отримання інформації з цих пристроїв будь-яким способом.

Проблема захисту статичних даних як на клієнтах користувачів, так і на серверах в мережі Інтернет, зростає щороку [1]. Для їх захисту використовуються різноманітні засоби технічного захисту, але кількість інформації швидко збільшується. Від так неможливо обмежити доступ до важливої інформації в достатній мірі та контролювати цей доступ постійно, оскільки це потребує значних витрат часу та ресурсів.

Шифрування даних – це найбільш доступний спосіб захисту цифрових даних сьогодні. Проте часто користувачі ІТС не користуються засобами шифрування або використовують їх некоректно. Від так користувачі часто втрачають ключі шифрування та, як результат, втрачають доступ до власних даних.

Об'єктом дослідження є алгоритми заперечуваного шифрування, зокрема алгоритм заперечуваного шифрування з відкритим ключем на основі розширеної криптографічної схеми Рабіна [2].

Використання алгоритму [2] зменшує ймовірність НСД до інформації за будь-яких обставин, дозволяє обмежити доступ до неї та її конфіденційність в разі здійснення НСД, оскільки її відновлення досить складний процес. Крім того, після виявлення НСД власни-

DOI 10.15588/1607-3274-2019-1-17

ки ІТС мають час для зміни ключів та повторного шифрування даних для відновлення її захисту.

Предметом дослідження є вирішення проблеми щодо неможливості використання алгоритмів заперечуваного шифрування, зокрема алгоритму заперечуваного шифрування з відкритим ключем на основі розширеної криптографічної схеми Рабіна, для блочного шифрування даних. Вказані алгоритми не передбачають блочного шифрування, тому галузі їх практичного застосування обмежені.

Метою дослідження є пошук підходу та рішення, яке дозволить виконати блочне шифрування даних за допомогою алгоритмів заперечуваного шифрування, зокрема алгоритму [2]. Окрім того, з метою збереження властивостей вихідних алгоритмів заперечуваного шифрування даних, запропоновані авторами підходи та рішення не повинні вносити суттєві зміни в структуру та властивості вихідних алгоритмів заперечуваного шифрування.

1 ПОСТАНОВКА ЗАДАЧІ

Алгоритми заперечуваного шифрування даних забезпечують перетворення вихідних даних в криптограми, значення яких подібні до криптограм сформованих за допомогою алгоритмів імовірнісного шифрування даних [2, 7]. Іншою особливістю цих алгоритмів, яка є найбільш важливою, це можливість отримання кількох варіантів вихідних даних, при дешифруванні криптограм.

Такий метод захисту даних є ефективним проти атак на основі примусу. В його основі лежить виконання наступних перетворень (1):

$$\begin{cases} Y = f_1(X), \\ X' = f_2(Y). \end{cases} \quad (1)$$

Згідно вище наведених виразів на вхід функції f_1 подається набір байтів X, який перетворюється в криптограму Y, значення якої мають псевдо випадковий характер. При зворотному перетворенні криптограма Y подається на вхід функції f_2 і в результаті користувач отримує вихідний набір байтів X', який попередньо обирається системою шифрування або розпізнається користувачем за допомогою функції f_2. Це дає можливість як заперечити факт існування даних, так і ввести зловмисника в оману надавши йому фіктивні дані.

Окрім двозначності відновлених даних, стійкість існуючих алгоритмів заперечуваного шифрування ґрунтується на неможливості вирішення задач факторизації та дискретного логарифмування. Проте саме умови цих задач накладають обмеження на застосування алгоритмів заперечуваного шифрування на практиці.

Одним з таких обмежень є складність їх реалізації та застосування для роботи з великими об'ємами даних (наприклад, в складі «big data»), оскільки розмір вхідних даних суттєво обмежений довжиною ключа шифрування (2):

$$\|M\| \le 2^N. \quad (2)$$

Для вирішення вказаної проблеми в роботі [3] запропоновано підхід до побудови алгоритмів заперечуваного шифрування на основі блочних шифрів, в основі якого лежить наступне:

1) використання двох різних ключів дешифрування даних K_T (секретних) та K_M (публічних);

2) генерації випадкових значень $R: \{r_1, r_2, r_3, \dots r_n\}$, які задовольняють наступній системі виразів (3):

$$\begin{cases} m_i = E_{K_M}(m_i, r_i) \bmod 2^u \\ t_i = D_{K_T}(t_i, r_i') \bmod 2^u \end{cases} \Rightarrow m_i \equiv t_i. \quad (3)$$

Вище вказаний спосіб заперечуваного шифрування даних навіть з використанням блочних алгоритмів не є досить ефективним, оскільки вибір випадкових значень $R: \{r_1, r_2, r_3, \dots r_n\}$ передбачає їх прямий перебір, що по складності схоже на задачу дискретного логарифмування.

Крім того, запропонований спосіб шифрування даних передбачає використання двох різних ключів (K_T та K_M) для дешифрування публічних та секретних даних, що суттєво впливає на час роботи алгоритму.

Таким чином, на сьогодні ефективних реалізацій алгоритмів заперечуваного шифрування на основі блочних перетворень не існує, що суттєво звужує галузь застосування заперечуваного шифрування.

В даній статті, перед авторами стоять наступні завдання: перевірити ефективність запропонованого ними способу побудови блочних алгоритмів заперечуваного шифрування, розробити робочий прототип, який зберігає особливості вихідного алгоритму заперечуваного шифрування, реалізує механізм блочного шифрування даних та має кращі показники продуктивності, аніж попередні алгоритми.

2 ОГЛЯД ЛІТЕРАТУРИ

Розробка та дослідження алгоритмів заперечуваного шифрування досить актуальні серед науковців в галузі інформаційної безпеки [2–13].

Алгоритм заперечуваного шифрування розроблений Раном Каретті [7] передбачає побітове шифрування даних. Значення кожного біта даних варіюється в залежності від граничного значення, яке встановлюється користувачем алгоритму. Вказане рішення не потребує блочної реалізації, оскільки шифрує кожний біт окремо та може працювати з даними будь-якого розміру. Проте вказаний алгоритм шифрування потребує значних витрат часу та пам'яті для інформації з значним розміром, оскільки n-операцій даного ал-

DOI 10.15588/1607-3274-2019-1-17

горитму повинні виконуватися для кожного біту окремо.

В протоколі RD-PKE [4], який розроблений Хамадою Ібрахімом, застосовується подібна методика шифрування даних. Проте в даному алгоритмі розмір даних, які можливо зашифрувати за одну ітерацію алгоритму, обмежений розміром контейнера KD та публічного ключа N (4):

$$\|KD\| < \sqrt{N}. \quad (4)$$

Хоча Хамада Ібрахім запропонував спосіб для подолання обмеження (4) за допомогою склеювання зашифрованих контейнерів, але через незначний розмір даних, які може вмістити контейнер KD, вказаний алгоритм також потребує значних витрат часу та пам'яті для роботи з реальними наборами даних.

В основі перетворень алгоритму заперечуваного шифрування [8], який запропонували Джин-Квін Ванг та Бо Менг, лежить використання схем шифрування RSA та Ель-Гамаля. Як наслідок, розмір вхідних даних різко обмежується довжиною публічного ключа згідно умови (4). Таким чином, як і в випадку з протоколом RD-PKE [4], даний алгоритм є асиметричним і не може працювати з великими розмірами даних. Тому галузі його застосування обмежені.

Інформація щодо існування або розробки варіацій авторами алгоритму [8], яка дозволяє реалізувати блочне шифрування даних, в відкритих джерелах відсутня.

Криптографічна система GM [3], яка розроблена Ш. Голвасером та С. Мікалі, реалізує непроникний псевдо імовірний варіант шифрування даних. Вказаний алгоритм не дає зловмиснику можливості отримати будь-які фрагменти інформації, яка була зашифрована даним алгоритмом.

В криптографічній системі GM використовується схема генерації ключів подібна до схеми RSA, проте її авторами запропоновано підхід для реалізації блочного шифрування даних шляхом склеювання зашифрованих блоків з даними.

Для дешифрування одного або кількох з вказаних блоків використовується один і той самий секретний ключ. Таким чином, при отриманні зловмисником даного ключа, криптографічна система GM втрачає свою надійність оскільки інформація з криптограми однозначно дешифрується та переходить в розпорядження зловмисника.

М. А. Молдовян запропонував алгоритм заперечуваного шифрування [2], який грунтується на використанні розширеної криптографічної схеми Рабіна [2]. Використання вищевказаної схеми шифрування надає можливість дешифрувати декілька варіантів публічних та секретних вихідних даних з використанням лише одного секретного ключа. Проте можливість застосування вказаного алгоритму обмежується розміром секретного ключа, так і особливостями криптографічної схеми Рабіна. Інформація щодо подальшого дослідження авторами вказаного алгоритму та створення схеми блочного шифрування на його основі в відкритих джерелах відсутня.

Проте М.А. Молдовян запропонував декілька подібних алгоритмів, в основі яких лежить використання блочного шифрування даних [5, 6]. Але дослідивши вказані алгоритми автори даної статті дійшли висновку, що вказані алгоритми не реалізують заперечуване шифрування вихідних даних, оскільки для дешифрування публічних та фіктивних даних М.А. Молдовян використовує два різних ключа дешифрування K_T та K_M. Таким чином, в разі з застосування грубої сили зловмисник має можливість отримати як один, так і обидва ключа, що дає йому змогу самостійно дешифрувати секретні дані користувача. Вище вказане не відповідає вихідним ідеям заперечуваного шифрування.

Крім того, для побудови блочних алгоритмів шифрування даних М. А. Молдовян використав існуючі блочні алгоритми шифрування. Проте в порівнянні з ними продуктивність алгоритмів [5, 6] є досить низькою, що ставить під сумнів можливість їх практичного застосування. Також в [2, 5, 6] не зазначено ані умов в яких проводилися експерименти, ані особливості реалізації вказаних алгоритмів шифрування.

3 МАТЕРІАЛИ І МЕТОДИ

Враховуючи недоліки способу заперечуваного шифрування даних на основі блочних шифрів [5, 6], які підтверджуються експериментами, авторами проведено аналіз вказаного способу заперечуваного шифрування даних та запропоновано інший підхід до вирішення цієї проблеми.

Таким чином, з урахуванням особливостей структури вихідного алгоритму [2], при внесенні змін автори розділили його на 4 окремих частини: генерація ключів (алгоритм 1), шифрування даних (алгоритм 2), дешифрування публічних даних (алгоритм 3) та дешифрування секретних даних (алгоритм 4).

Алгоритм генерація ключів передбачає наступну послідовність дій:

– адресант (абонент Б) генерує u-бітовий секретний ключ $(p;q)$;

– адресант (абонент Б) обчислює відкритий ключ за допомогою виразу: $N = p \cdot q$;

– адресант (абонент Б) передає відкритий ключ N адресату (абонент А).

Алгоритм шифрування даних передбачає наступну послідовність дій (рис. 1):

– адресант (абонент Б) генерує секретний набори публічних M та секретних T даних;

– адресант (абонент Б) вирівнює набори публічних M та секретних T даних, доки $\|M\| \neq \|T\|$;

– адресант (абонент Б) розділяє набори публічних M та секретних T даних на блоки m_k та t_k розміром u-біт;

DOI 10.15588/1607-3274-2019-1-17

– адресант (абонент Б) обчислює контрольні суми блоків m_k та t_k за допомогою функції f_H;

– адресант (абонент Б) склеює блоки (m_k, t_k) та їх контрольні суми f_H;

– адресант (абонент Б) перемішує байти в блоках (m_k, t_k) за допомогою функції f_{SB};

– адресант (абонент Б) фіксує блоки (m_k, t_k) за допомогою функції f_M;

– адресант (абонент Б) обчислює випадкові значення r_k для блоків t_k за допомогою функції f_R;

– адресант (абонент Б) обчислює блок криптограми c_k для пар (m_k, r_k) за допомогою функції f_E.

Алгоритм дешифрування публічних даних передбачає наступну послідовність дій (рис. 2):

– адресат (абонент А) розділяє криптограму на блоки c_k;

– адресат (абонент А) розділяє блоки c_k на субблоки A та B;

– адресат (абонент А) обчислює корені рівняння $x^2 \pm A \cdot x - B = 0 \pmod N$ для пар (A, B) за допомогою функції f'_R;

– адресат (абонент А) обчислює варіанти вихідних даних m'_k для наборів (r_k, a_k);

– адресат (абонент А) знімає фіксацію блоків m'_k за допомогою функції f'_M;

– адресат (абонент А) здійснює зворотну перестановку байтів в блоках m'_k за допомогою функції f'_{SB};

– адресат (абонент А) виконує аналіз варіантів вихідних даних m'_k за допомогою функції f_{AN};

– адресат (абонент А) обчислює блоки вихідних даних d_k за допомогою функції f_D.

Алгоритм дешифрування секретних даних передбачає наступну послідовність дій (рис. 3):

– адресат (абонент А) виконує кроки 3.1–3.7 алгоритму 3;

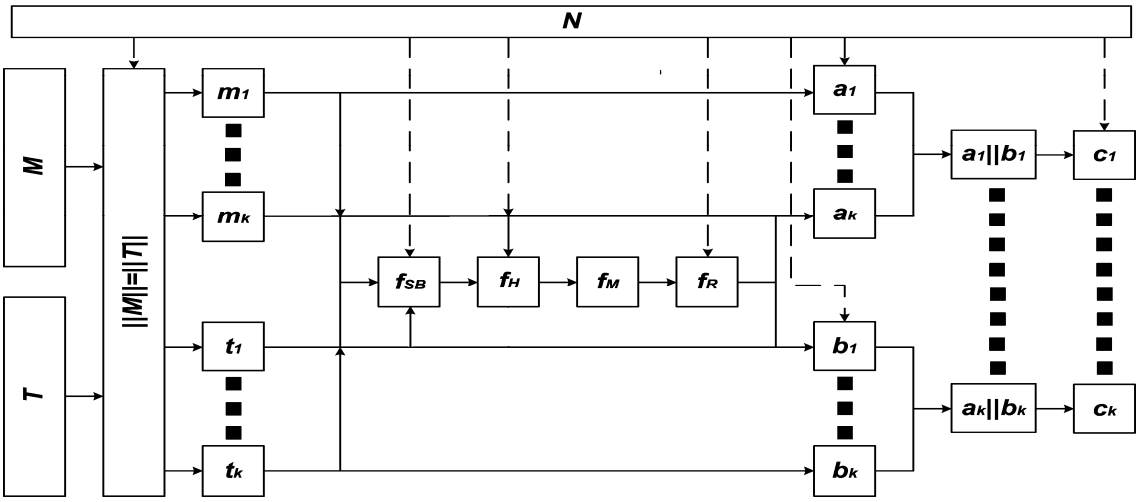


Рисунок 1 – Схема блочного шифрування даних (алгоритм 2)

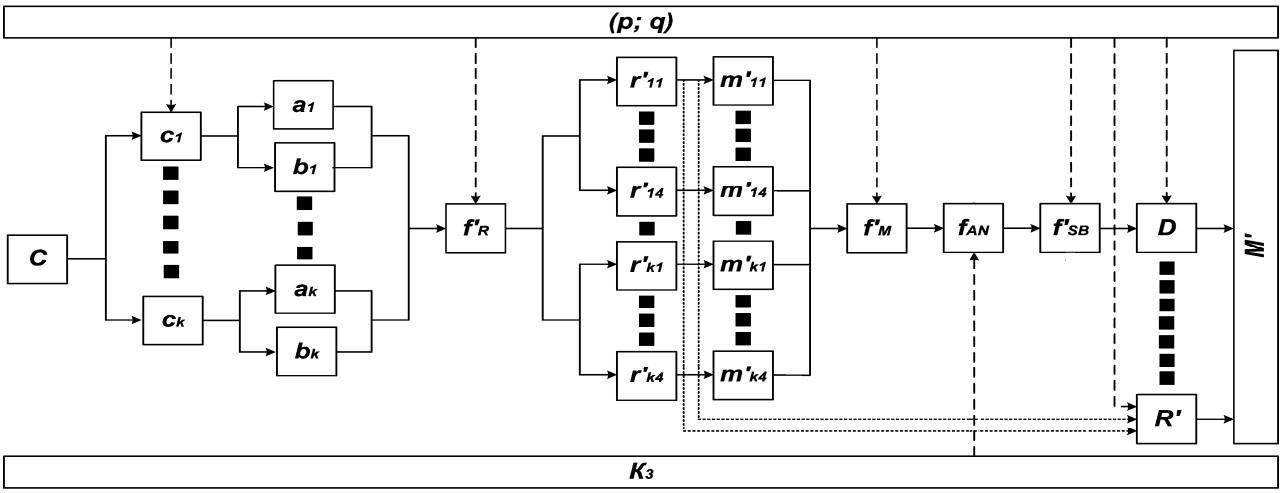


Рисунок 2 – Схема блочного дешифрування публічних даних (алгоритм 3)

DOI 10.15588/1607-3274-2019-1-17

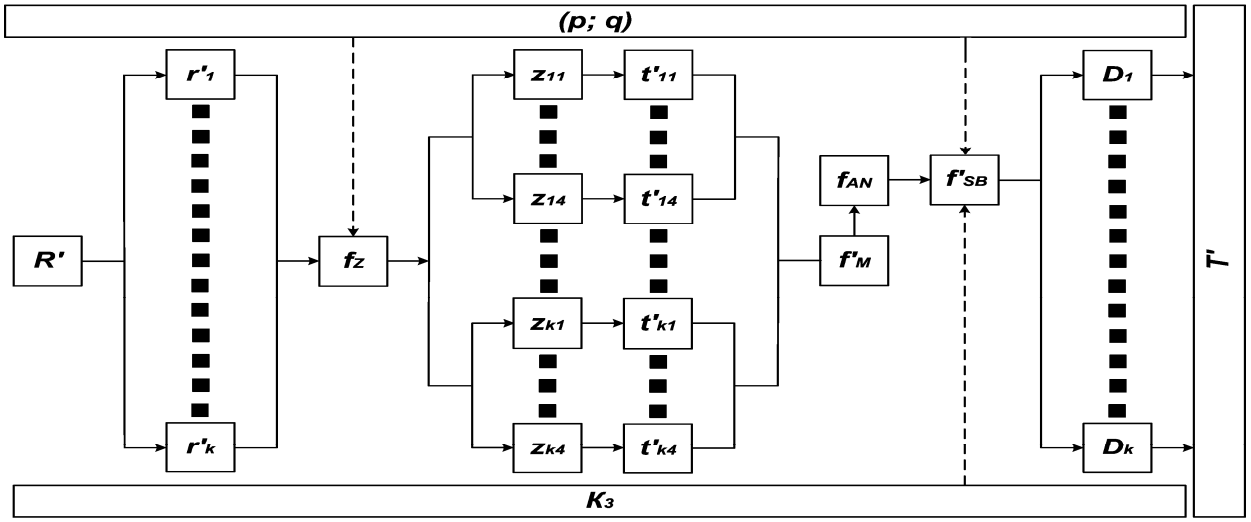


Рисунок 3 – Схема блочного дешифрування секретних даних (алгоритм 4)

– адресат (абонент А) отримує підтвердження від третьої сторони K_3;

– адресат (абонент А) обчислює корені рівняння $\sqrt{r'_k}\,(\mathrm{mod}\,N)$ за допомогою функції f_Z;

– адресат (абонент А) обчислює варіанти вихідних даних t'_k для наборів (z_k, a_k);

– адресат (абонент А) знімає фіксацію блоків t'_k за допомогою функції f'_M;

– адресат (абонент А) здійснює зворотну перестановку байтів в блоках t'_k за допомогою функції f'_{SB};

– адресат (абонент А) виконує аналіз варіантів вихідних даних t'_k за допомогою функції f_{AN};

– адресат (абонент А) обчислює блоки вихідних даних d_k за допомогою функції f_D.

Вище запропоновані алгоритми ґрунтуються лише на описі вихідного алгоритму. Але в ході реалізації як вихідного, так і запропонованих алгоритмів у авторів виникли проблеми:

1) з вирівнюванням потоків фіктивних та секретних даних;

2) генерацією та застосуванням процедури f_{SB};

3) розпізнаванням дешифрованих даних [2];

4) втратою частини даних, при дешифруванні криптограми;

5) витратами часу та пам'яті на виконання.

Так для вирішення виявлених проблем автори внесли додаткові зміни, які відображені на рис. 1–3.

Окрім того, з метою дослідження робото спроможності розробленого алгоритму та встановлення його оптимальних параметрів, при яких його робота найбільш ефективна, авторами проведено оцінки додаткових параметрів алгоритму (наприклад, кількість блоків, швидкість шифрування-дешифрування тощо) та проведено серії експериментів з оцінки часу і витрат пам'яті на виконання алгоритмів 2–4.

4 ЕКСПЕРИМЕНТИ

Для проведення експериментів автори використали наступне технічне устаткування: x86-based PC, ЦП x64 Family 16 Model 6 Stepping 3 Authentic AMD ~1679 МГц, ОЗП 3067МБ, ОС Microsoft Windows 7 Ultimate 6.1.7601 Service Pack 1 збірка 7601, IDLE Python v3.6.2, бібліотеки line_profiler та mem_profiler, MS Excel 2003.

З метою повної та усесторонньої оцінки ефективності роботи запропонованих алгоритмів автори відкинув оцінку алгоритму 1 та зосередився на алгоритмах 2–4.

В ході проведення експериментів авторами встановлено наступні змінні, залежність від яких буде досліджуватися:

1) формати вхідних даних: текстові (*.dat, *.txt, *.doc, *.xls), графічні (*.pdf, *.jpg, *.gif, *.png, *.bmp), медіа (*.avi, *.3gp, *.mp3) та інші (*.exe, *.dll, *.zip).

2) розмір вхідних даних: 1КБ ÷ 100МБ;

3) розмір ключа шифрування: 128, 256, 512, 1024, 2048 та 4096 біт;

4) розмір хешу даних: 1 ÷ 2 байти;

5) мітки довільного змісту, розмір яких становить по 3 байти кожна (наприклад, «!block!»).

Узагальнена схема проведення експериментів наведена на рис. 4.

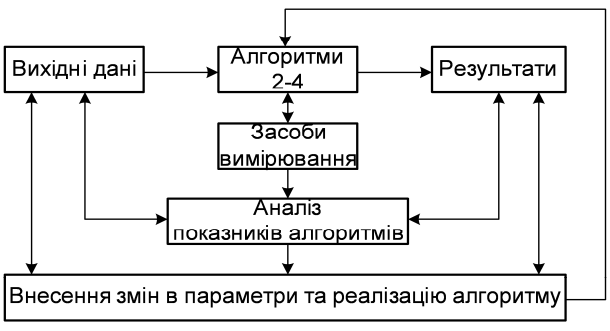


Рисунок 4 – Схема проведення експериментів

DOI 10.15588/1607-3274-2019-1-17

Проведення експериментів з оцінки ефективності виконання та пошуку найбільш оптимальних параметрів в алгоритмах 2–4 передбачають наступний порядок дій:

1) вибір вхідних параметрів та областей їх значень;
2) виконання алгоритмів 2–4;
3) вимірювання часу виконання алгоритмів 2–4;
4) вимірювання витрат пам'яті на виконання алгоритмів 2–4;
5) аналіз отриманих результатів та пошук найбільш оптимальних значень для кожного з алгоритмів;
6) зміна параметрів та підходів до реалізації алгоритмів.

5 РЕЗУЛЬТАТИ

За результатами експериментів автори отримали дані, які описують роботу та ефективність запропонованих ним алгоритмів, та наведені на рис. 5–10.

Так, з метою оцінки можливості застосування вихідного алгоритму для блочного шифрування даних та його реалізації на основі сучасних програмно-апаратних рішень, авторами проведено вимірювання витрат часу та пам'яті, які необхідні для виконання алгоритму 2, від розміру вхідних даних (рис. 5).

В залежності від розміру публічного ключа N та різниці розмірах публічних M і секретних T даних, практичний розмір вихідної криптограми C в алгоритмі 2 зростає на 5–49 % в порівнянні з теоретичними оцінками (рис. 6).

Іншим ключовим питанням було встановлення залежності часу виконання алгоритмів 2–4 від розміру ключа шифрування, оскільки в запропонованій авторами версії вихідного алгоритму розмір ключа безпосередньо впливає як на кількість ітерацій в алгоритмах, так і на витрати пам'яті для їх виконання (рис. 6–7). Також вплив вказаних обставин (рис. 5) на витрати часу та пам'яті в алгоритмі 2 незначний, але він зростає, при виконанні в алгоритмів 3 і 4 (рис. 7–9).

Крім того, з метою пошуку оптимального розміру ключа авторами проведено додаткове дослідження для виявлення залежності часу виконання алгоритмів 2–4 від ступеню розкладання вихідних даних на окремі блоки, які й обробляються за 1 ітерацію роботи алгоритмів 2–4 (рис. 10).

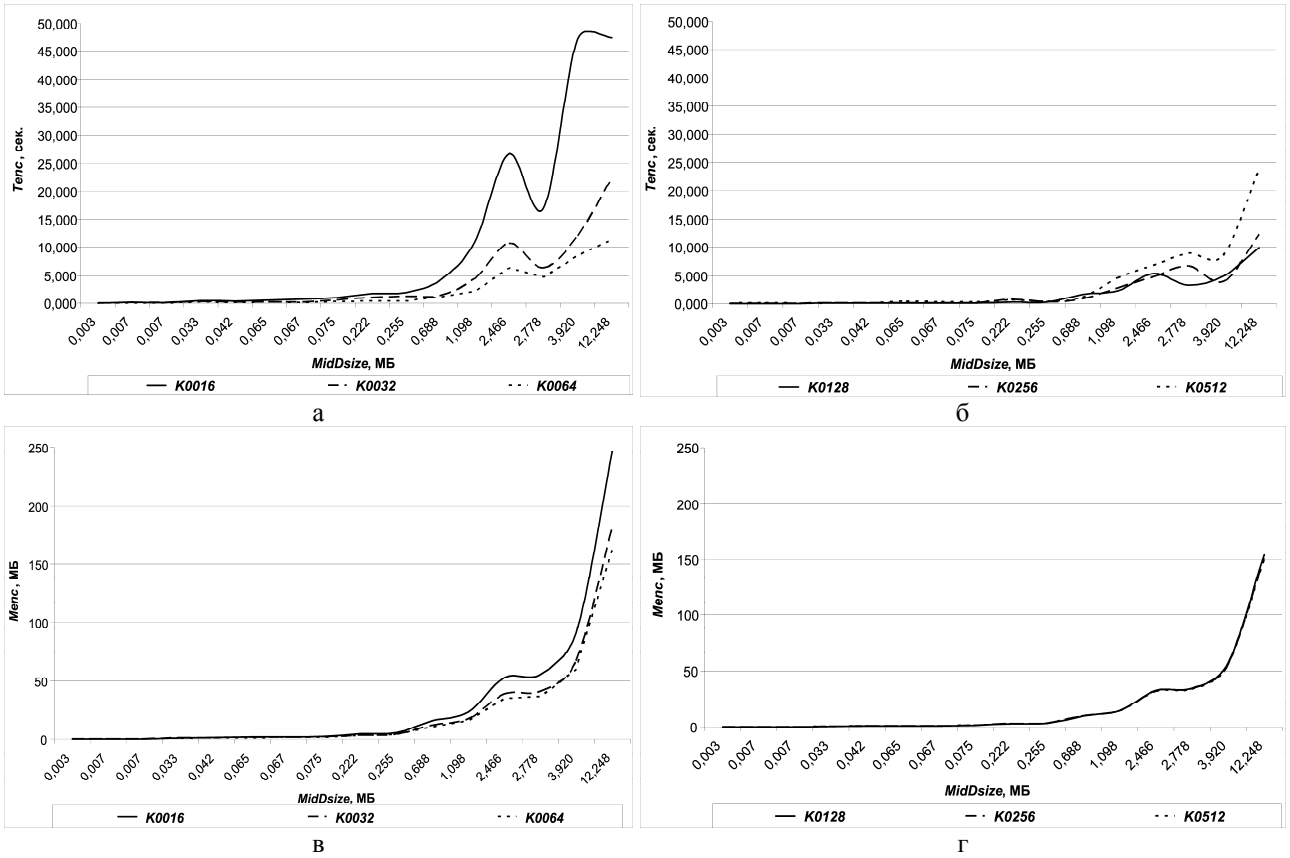
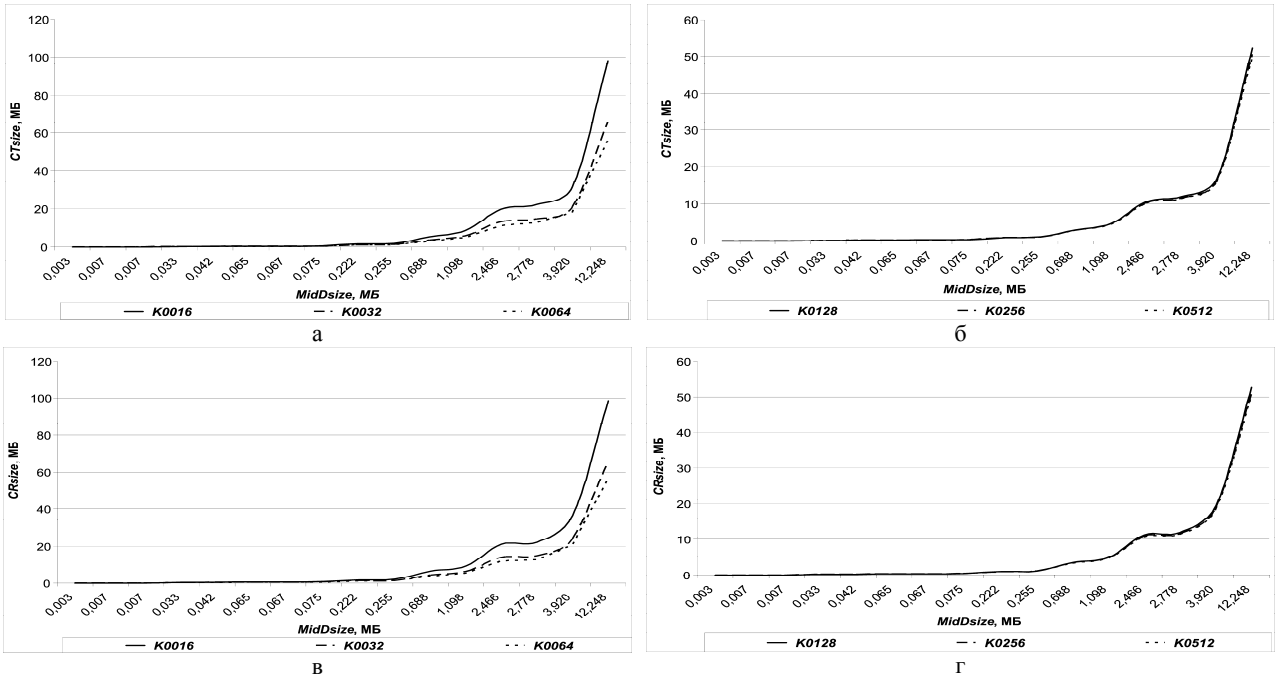


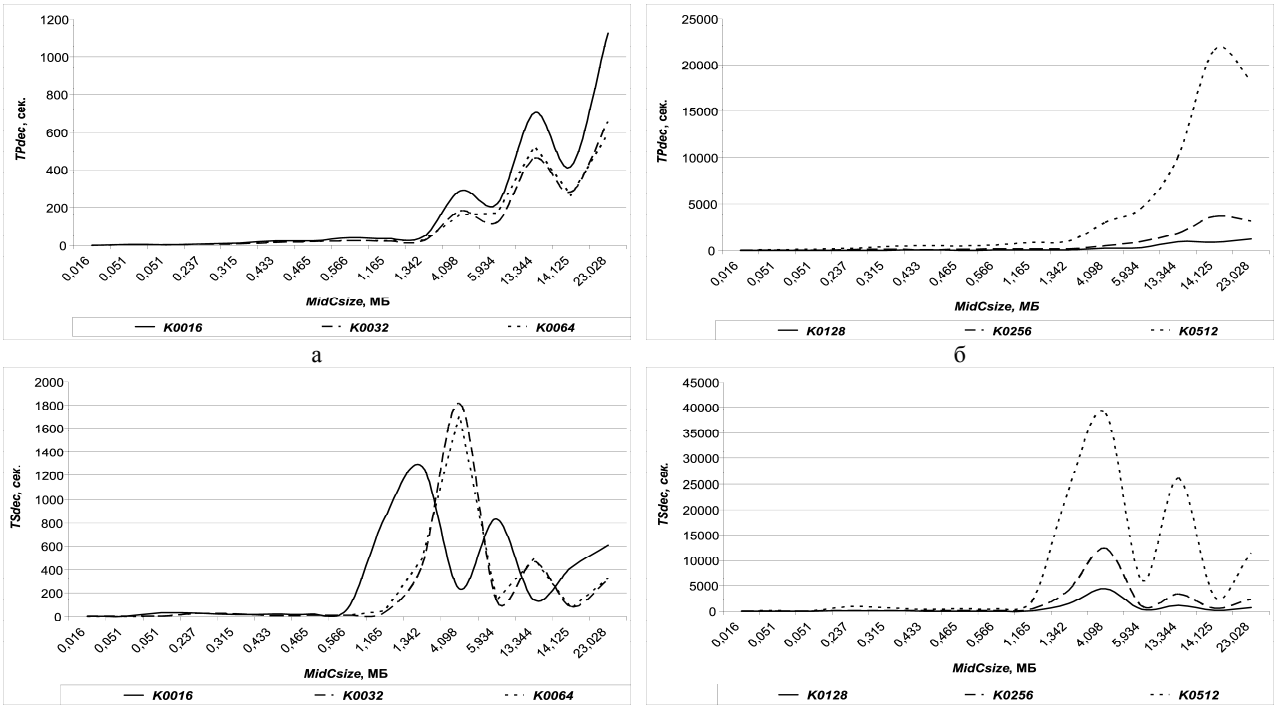
Рисунок 5 – Оцінка витрат часу та пам'яті на виконання алгоритму 2:
а, в – при використанні ключів розміром 16–64 байти, б, г – при використанні ключів розміром 128–512 байт

DOI 10.15588/1607-3274-2019-1-17



а б в г

Рисунок 6 – Залежність розміру криптограми від розмірів вхідних даних та публічного ключа 16–512 байт в алгоритмі 2: а, б – теоретичний розмір криптограми; в, г – практичний розмір криптограми



а б в г

Рисунок 7 – Витрати часу на виконання алгоритмів 3 і 4 з використанням ключів 16–512 байт: а, б – при дешифруванні публічних даних; в, г – при дешифруванні секретних даних

DOI 10.15588/1607-3274-2019-1-17

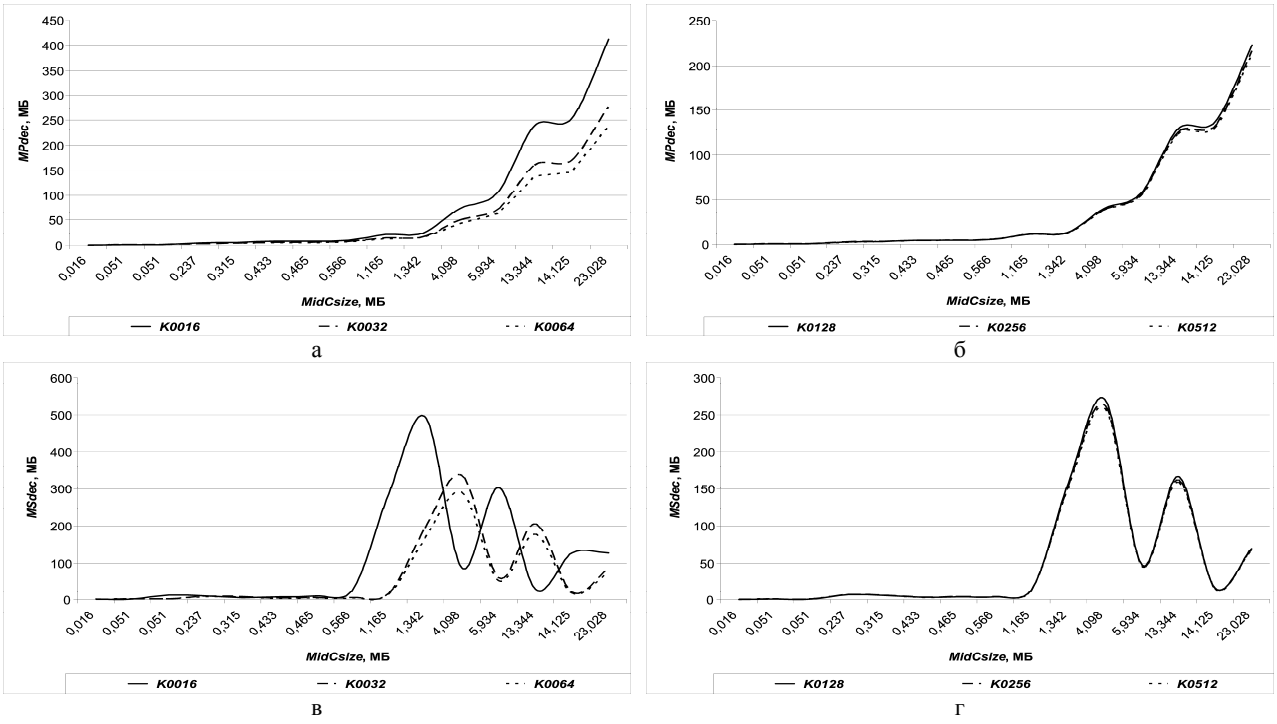


Рисунок 8 – Витрати пам’яті на виконання алгоритмів 3 і 4 з використанням ключів 16–512 байт:
а, б – при дешифруванні публічних даних; в, г – при дешифруванні секретних даних

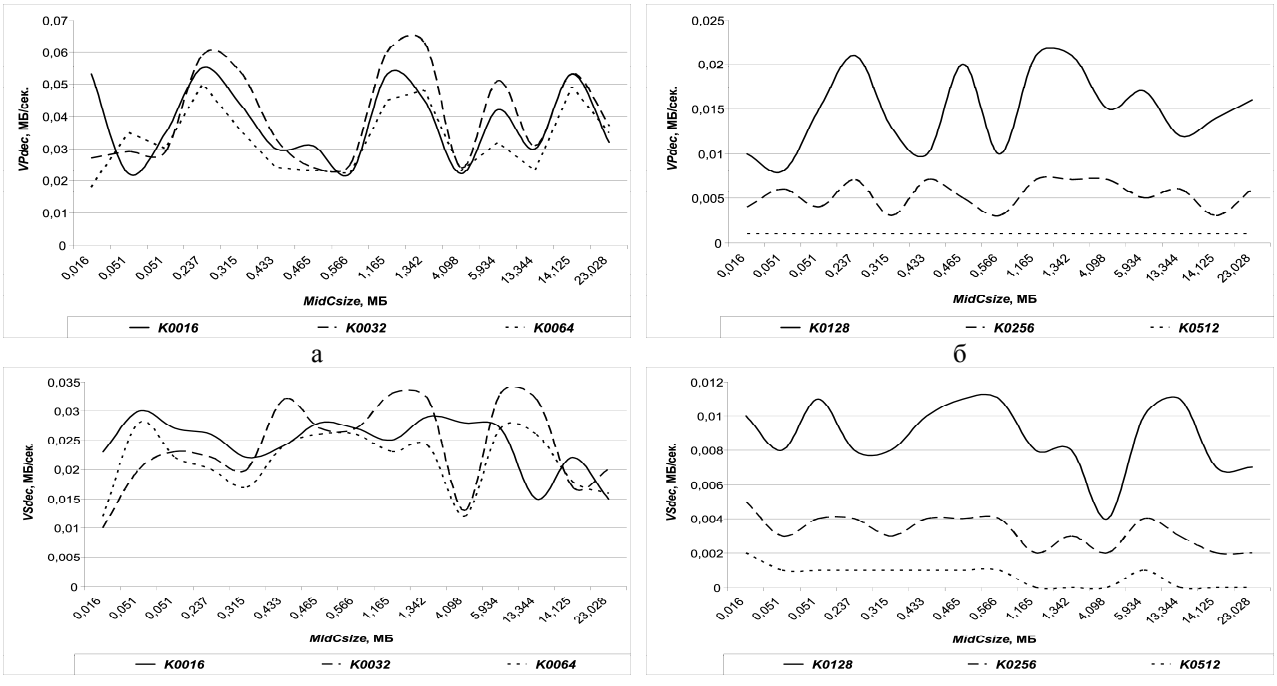
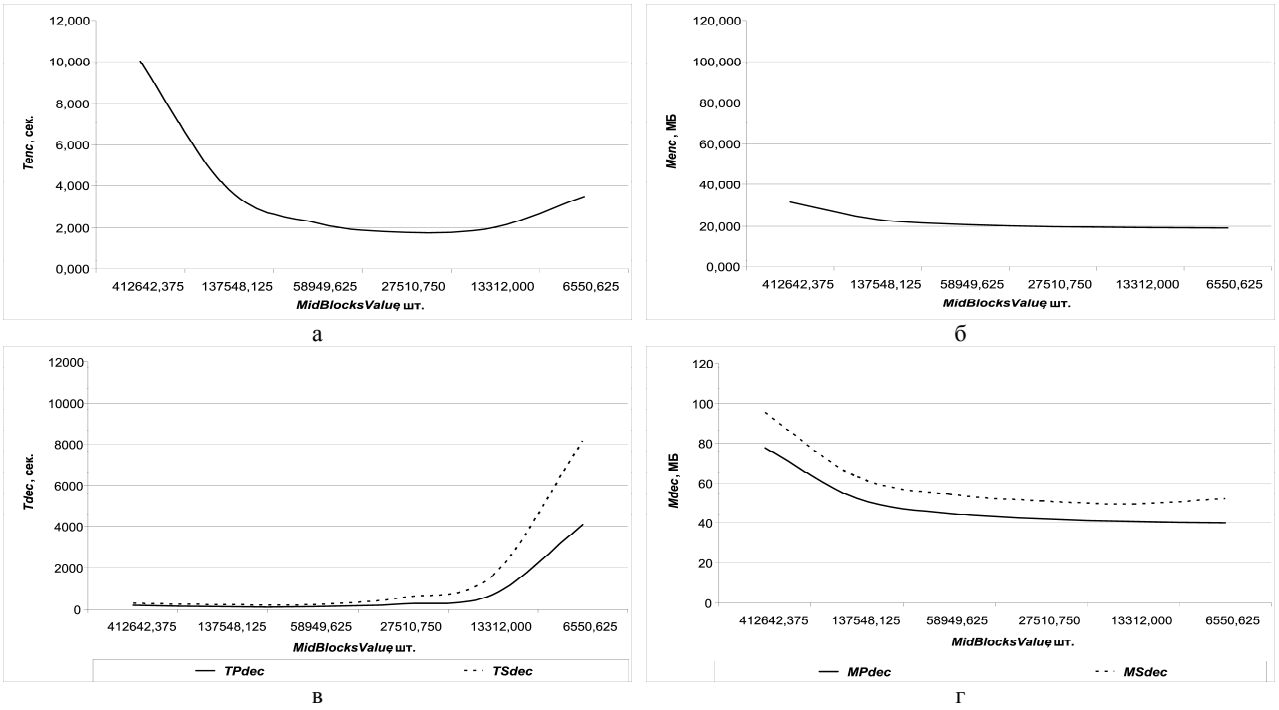


Рисунок 9 – Швидкість виконання алгоритмів 3 і 4 з використанням ключів 16–512 байт:
а, б – при дешифруванні публічних даних; в, г – при дешифруванні секретних даних

DOI 10.15588/1607-3274-2019-1-17



а б в г

Рисунок 10 – Залежність витрат часу та пам'яті на виконання алгоритмів 2–4 від ввід кількості блоків:
а, б – при шифруванні даних; в, г – при дешифруванні публічних та секретних даних

6 ОБГОВОРЕННЯ

В результаті аналізу запропонованого алгоритму заперечуваного шифрування даних встановлено, що суттєвий вплив на ефективність роботи та можливість його застосування створюють витрати часу та пам'яті, які необхідні для його виконання, розміри вхідних даних та ключа, а також ступінь розбиття вхідних даних на блоки.

При узагальненому аналізі роботи алгоритму 2 встановлено, що суттєвий вплив на розмір кінцевої криптограми створює різниця в розмірах вихідних файлів з публічними та секретними даними. Таким чином, при різниці до 30%, розмір вихідної криптограми може зрости в 1,5 рази порівняно з теоретичними розрахунками (5):

$$\frac{\left\|\|M\|-\|T\|\right\|}{\max\left(\|M\|,\|T\|\right)} \approx 0{,}3 \cong 1{,}5\cdot\|C\|. \qquad (5)$$

Вказане пояснюється неможливістю підбору вихідних даних з однаковими розмірами та необхідністю їх особливістю алгоритму в частині вирівнювання файлів на етапі їх шифрування.

Також подібні коливання спостерігаються, при виконанні алгоритмів 2–4, оскільки під час експериментів виконувалися наступні умови (6):

$$\begin{cases} t \to \infty, & \text{при } \|\mathrm{N}\| \to \infty \wedge \|\mathrm{M}\| \to 0 \\ t \to \infty, & \text{при } \|\mathrm{N}\| \to \infty \wedge \|\mathrm{T}\| \to 0 \\ t \to 0, & \text{інакше} \end{cases} \qquad (6)$$

При аналізі алгоритму 2 встановлено, що за будь-яких обставин його часові характеристики в більшій степені рівномірні та носять експоненціальний характер, що робить його придатним для роботи на будь-яких ЕОМ. Також, при подальшому аналізі показників, встановлено, що час виконання алгоритму 2 спадає на деяких проміжках графіку (рис. 5), через коливання в різниці розмірів вхідних даних. Таким чином, підчас тестування алгоритму 2 встановлено, що різниця в розмірах вхідних даних, від 1 МБ та більше, призводить до 10-кратного зростання часу шифрування даних.

Крім того, час шифрування публічних та секретних даних в значній мірі залежить від розміру ключа шифрування. При оцінці впливу розмірів ключа на виконання алгоритму 2 встановлено, що найбільший час виконання алгоритму спостерігається, при використанні 128-бітового ключа, та середні, при використанні 512–2048-бітових ключів. Таким чином, експериментально встановлено, що найбільш оптимальний час шифрування даних забезпечується, при використанні ключа в 1024 біти.

Цілком інша ситуація спостерігається при виконанні алгоритмів 3 і 4, оскільки різниця в часі їх виконання досягає 150–200% (рис. 7–9). Так, при розмірі криптограми від 3 МБ, спостерігається збільшення часу на виконання алгоритмів 3 і 4, а при 11 МБ та більше спостерігається незначне спадання часу виконання алгоритму 3 та його значне зростання для алгоритму 4. Вказане є наслідком подвоєння кількості обчислень в алгоритмі 3 порівняно з алгоритмом.

DOI 10.15588/1607-3274-2019-1-17

Також час виконання алгоритмів 3 і 4 так само залежать від розміру ключа, як і алгоритм 2. Так при детальному аналізі роботи алгоритмів 3 і 4 встановлено, що середні показники часу їх виконання спостерігаються, при використанні 128, 256, 512 та 1024-бітових ключів, та гірші, при 2048 і 4096-бітового ключах. Проте порівняно з алгоритмом 2 присутні деякі розбіжності, оскільки:

1) в алгоритмі 2 найбільш оптимальний розмір ключа становить 1024 біти (рис. 5);

2) в алгоритмі 3 найбільш оптимальний розмір ключа становить 512 біт (рис. 7–9);

3) в алгоритмі 4 найбільш оптимальний розмір ключа становить 256 біт (рис. 7–9).

Враховуючи вище вказане, з метою пошуку причин збільшення часу виконання алгоритмів 2–4 та їх усунення, авторами проведено оцінку витрат пам'яті, які відведені на їх виконання.

Так в ході експериментів з алгоритмом 2 встановлено, що витрати пам'яті на шифрування вхідних даних зростають пропорційно розміру вхідних даних та є значними, при використанні ключів до 512 біт включно. Вказане пояснюється тим, що при використанні ключів малого розміру кількість блоків даних зростає, що призводить до збільшення кількості обчислень з ними. Інше спостерігається, при аналізі алгоритмів 3 і 4.

Так при дешифруванні публічних та секретних даних витрати пам'яті значно збільшуються порівняно з алгоритмом 2, але в межах алгоритмів 3 і 4 є майже однакові для ключів 512–2048 біт. Вказане є наслідком вирівнювання вихідних даних в алгоритмі 2 (рис. 1).

Таким чином, автори запропонували ідею блочного шифрування даних для реалізації механізмів заперечуваного шифрування, в якому не використовуються існуючі блочні алгоритми шифрування, лише їх елементи. Так в порівнянні з ідеями, які запропоновані в [5, 6], підхід авторів до вирішення проблеми забезпечує збільшення швидкості шифрування даних до 200 разів та дешифрування даних до 10 разів (залежно від розміру ключа).

Порівняння продуктивності запропонованого авторами алгоритму з першочерговою ідеєю та блочним алгоритмом шифрування даних [5, 6] наведені в табл. 1 та на рис. 11.

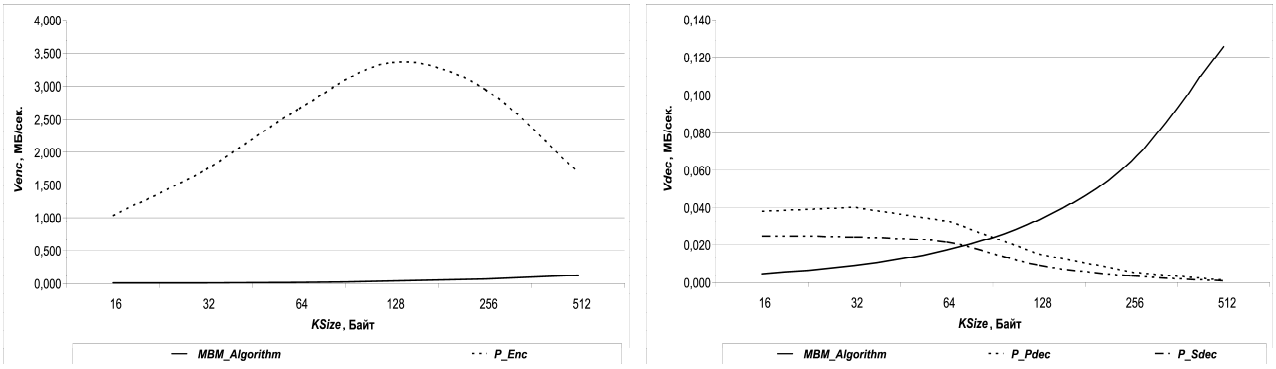
Вище вказані результати доводять можливість практичної реалізації ідеї авторів для здійснення блочного шифрування даних. Також отримані результати свідчать про можливість використання запропонованого алгоритму для захисту статичних даних та перспективи його використання в «big data».

В результаті проведених експериментів встановлено, в порівнянні з показниками швидкодії блочних алгоритмів шифрування [5, 18], запропоновані авторами алгоритм в 200000 разів програє в швидкості. Але так само, за певних умов, запропонований алгоритм демонструє кращі показники продуктивності в порівнянні з подібними алгоритмами [5, 6].

Таким чином, враховуючи недоліки запропонованого алгоритму та особливості його реалізації, а також з метою перевірки запропонованої авторами ідеї

Таблиця 1 – Порівняння продуктивності алгоритмів шифрування

Найменування алгоритмів шифрування	Продуктивність алгоритмів, МБ/с				
	$K0032$	$K0064$	$K0128$	$K0256$	$K0512$
$MBM\_Algorithm$	0,005	0,009	0,017	0,033	0,065
P_Enc	1,028	1,735	2,658	3,347	2,946
P_Pdec	0,038	0,040	0,033	0,015	0,005
P_Sdec	0,025	0,024	0,021	0,009	0,003



а б

Рисунок 11 – Порівняння швидкості виконання блочних алгоритмів заперечуваного шифрування, які запропоновані Молдовяном М. А. та авторами статті, при використанні ключів розміром 16–512 байт: а – порівняння швидкості шифрування; б – порівняння швидкості дешифрування

DOI 10.15588/1607-3274-2019-1-17

на інших алгоритмах заперечуваного шифрування даних, необхідно провести подальші дослідження алгоритму в наступних напрямках:

1) криптоаналіз запропонованого алгоритму заперечуваного шифрування даних для визначення його надійності та можливості практичного застосування;

2) зменшення часу виконання алгоритмів заперечуваного шифрування за рахунок використання багатопоточних обчислень;

3) зменшення витрат пам'яті на виконання алгоритмів заперечуваного шифрування за рахунок використання розподілених систем;

4) пристосування інших алгоритмів заперечуваного шифрування для блочного шифрування даних, проведення їх криптоаналізу та оцінки продуктивності;

5) дослідження блочних алгоритмів шифрування та їх елементів для побудови не гібридних, а блочних алгоритмів, які реалізують механізми заперечуваного шифрування.

ВИСНОВКИ

Згідно з метою поставленою на початку статті, авторами вирішено проблему використання алгоритмів заперечуваного шифрування, зокрема алгоритму заперечуваного шифрування з відкритим ключем на основі розширеної криптографічної схеми Рабіна, для блочного шифрування даних.

Вищевказані експерименти проводилися в межах заздалегідь визначених параметрів (розміри ключа, хешу даних, мітки, типи даних, тощо). Отримані авторами результати свідчать про перспективи застосування запропонованого ними підходу побудови алгоритмів заперечуваного шифрування даних, який дозволяє більш ефективно захищати як статичні набори даних на персональних комп'ютерах користувачів, так і набори даних в системах типу «big data».

За результатами проведених досліджень та експериментів вирішено наступні завдання:

– виконано огляд рішень проблеми, які дозволяють застосування алгоритмів заперечуваного шифрування для блочних перетворень;

– виконано аналіз елементів блочних алгоритмів для їх впровадження в обраний алгоритм заперечуваного шифрування;

– виконано модифікацію алгоритму заперечуваного шифрування з відкритим ключем на основі розширеної криптографічної схеми Рабіна;

– проведено експерименти для визначення оптимальних параметрів роботи запропонованого алгоритму;

– виконано порівняння результатів експериментів з даними аналогічних підходів.

Наукова новизна полягає в розробці підходу до реалізації блочних алгоритмів заперечуваного шифрування та його застосування до існуючих алгоритмів заперечуваного шифрування даних, зокрема до алгоритму [2].

Як наслідок, використовуючи вказаний підхід та алгоритм заперечуваного шифрування на основі розширеної криптографічної схеми Рабіна, автори синтезували прототип блочного алгоритму заперечуваного шифрування даних, який згідно з результатами проведених експериментів має кращі показники витрат часу та пам'яті, аніж їх аналоги [5, 6].

Практичне значення полягає в застосуванні нового підходу до побудови блочних алгоритмів заперечуваного шифрування даних, розробці прототипу блочного алгоритму заперечуваного шифрування даних на його основі, збереження базових властивостей алгоритмів заперечуваного шифрування. Також запропонований авторами підхід не вносить суттєвих змін в механізм роботи вихідного алгоритму, а отже залишає його привабливим для практичного застосування, при забезпеченні конфіденційності та доступності інформації.

ПОДЯКИ

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

ЛІТЕРАТУРА / ЛИТЕРАТУРА

1. Аналитический центр компании InfoWatch [Електронный ресурс]. – Москва : InfoWatch, 2017–2018. – Режим доступу: www.infowatch.ru/analytics/digest/19546.
2. Молдовян Н. А. Расширение криптосхемы Рабина: алгоритм отрицаемого шифрования по открытому ключу / Н. А. Молдовян, М. А. Вайчикаускас // Научно-технический центр оборонного комплекса «Компас»: Вопросы защиты информации. – 2014. – № 2. – С. 12–16.
3. Goldwasser S. Probabilistic encrypton / S. Goldwasser, S. Micali // Journal of Computer and System Sciences. – 1984. – Vol. 28. – P. 270–299.
4. Ibrahim H. Receiver-Deniable Public-Key Encryption / H. Ibrahim // International Journal of Network Security. – 2009. – Vol. 8, No. 2. – P. 159–165.
5. Молдовян Н. А. Отрицаемое шифрование на основе блочных шифров / Н. А. Молдовян, А. Р. Биричевский, Я. А. Мондикова // Информационно-управляющие системы. – 2014. – № 5. – С. 80–86.
6. Молдовян А. А. Способы псевдовероятностного блочного шифрования / А. А. Молдовян, Я. А. Татчина // Интеллектуальные технологии на транспорте. – 2018. – № 1. – С. 25–30.
7. Canetti R. Deniable Encryption / R. Canetti, C. Dwork, M. Naor, R. Ostrovsky // Advances in Cryptology: CRYPTO. – 1997. – Procedings. – P. 90–104.
8. Bo Meng. A Receiver Deniable Encryption Scheme / Bo Meng, Jiang Qing Wang // Proceedings of the 2009 International Symposium on Information Processing (ISIP'09). – 2009. – P. 254–257.
9. Гальченко А. В. Перспективи використання заперечуваного шифрування в галузі авіаперевезень / А. В. Гальченко // АВІА 2017: XIII Міжнародна науково-технічна конференція, Київ, 19–21 квітня 2017 р. : тези доповідей. – Київ : НАУ, 2017. – С. 24–28.
10. Гальченко А. В. Захист персональних даних з використанням алгоритмів неоднозначного шифрування / А. В. Гальченко // Вісник ЗНУ: Математичне моделювання та прикладна механіка. – 2017. – № 2. – С. 19–32.

DOI 10.15588/1607-3274-2019-1-17

11. Молдовян Н. А. Протокол отрицаемого шифрования по открытому ключу, включающий процедуру аутентификации пользователей / Н. А. Молдовян, М. С. Михтеев, К. Т. Нгуен // Вопросы защиты информации. – 2016. – № 3. – С. 9–15.
12. Молдовян Н. А. Протокол поточного отрицаемого шифрования с разделяемым ключом / Н. А. Молдовян, З. С. Баширов, Ж. А. Солнышкин // Вопросы защиты информации. – 2015. – № 3. – С. 27–31.
13. Шнайер Б. Прикладная криптография: Протоколы, алгоритмы, исходные тексты на языке Си / Б. Шнайер. – М. : Триумф, 2012. – 816 с.

Стаття надійшла до редакції 18.06.2018.
Після доробки 23.09.2018.

УДК 004.056.55

ОТРИЦАЕМОЕ ШИФРОВАНИЕ НА ОСНОВЕ ПРИМЕНЕНИЯ ПОДХОДОВ ГИБРИДНЫХ КРИПТОГРАФИЧЕСКИХ СИСТЕМ

Гальченко А. В. – профессионал по организации информационной безопасности Казенного предприятия «Научно-производственного комплекса« Искра », г. Киев, Украина.

Чопоров С. В. – канд. техн. наук, доцент кафедры «Программная инженерия» Запорожского национального университета, г. Киев, Украина.

АННОТАЦИЯ

Актуальность. Несанкционированный доступ к хорошо защищенным информационно-телекоммуникационным системам является весьма актуальной проблемой в области информационной безопасности [1]. Для решения этой проблемы предложено использование механизмов отрицаемого шифрования, в случае получения несанкционированного доступа к информации позволяют ее распорядителям как отрицать факт существования данных, так и обеспечить их конфиденциальность и защитить распорядителей информации от применения грубой силы со стороны злоумышленников для получения ключевой информации. В статье изложены подход к применению существующих алгоритмов отрицаемого шифрования для защиты информации в больших массивах данных.

Цель. Основная цель исследования заключается в проверке гипотезы о возможности использования алгоритмов отрицаемого шифрования для работы с большими массивами данных, поскольку все алгоритмы данного направления являются асимметричными и не адаптированы для работы с «big data».

Метод. Проверка гипотезы осуществляется путем введения дополнительных блоков обработки данных в выходной алгоритм отрицаемого шифрования с открытым ключом на основе расширенной криптографической схемы Рабина [2], структура и особенности которого наиболее подходят для проверки выдвинутой автором гипотезы.

Результаты. По результатам экспериментов авторами предложено прототип алгоритма отрицаемого шифрования, который реализует блочное шифрование данных, а также сохраняет особенности механизма двусмысленности выходного алгоритма шифрования. Кроме того, предложенные авторами изменения обеспечивают увеличение производительности работы предложенного алгоритма, при реализации определенных вычислений, по сравнению с существующими подходами [3–6].

Выводы. Авторами решена задача применения существующих алгоритмов отрицаемого шифрования для защиты информации в больших массивах данных, на примере алгоритма отрицаемого шифрования с открытым ключом на основе расширенной криптографической схемы Рабина. Предложенный подход к построению гибридного алгоритма с механизмом оспаривания демонстрирует не только сохранение основных свойств базового алгоритма, но и возможность блочного шифрования данных любого размера с хорошими показателями производительности. То есть предложенный алгоритм не только позволяет решить задачу обеспечения конфиденциальности данных, в случае несанкционированного доступа к ним, но и делает его пригодным для практического применения.

КЛЮЧЕВЫЕ СЛОВА: блочное шифрование, грубая сила, отрицаемое шифрования, информационно-телекоммуникационная система, двусмысленность, несанкционированный доступ, обработка данных, псевдовероятность, публичные данные, распознавание, расширенная криптографическая схема Рабина, секретные данные, статические данные.

UDC 004.056.55

DENIABLE ENCRYPTION BASED ON HYBRID CRYPTOGRAPHIC SYSTEMS USING

Galchenko A. V. – Information Security Professional at Enterprise State «Scientific and Manufactured Complex «Iskra», Zaporizhzhya, Ukraine.

Choporov S. V. – PhD, Associate Professor of the Software Engineering Department, Zaporizhzhya National University, Zaporizhzhya, Ukraine.

ABSTRACT

Context. Unauthorized access to well-protected information and telecommunication systems is a topical problem in the information security field [1]. For this problem solving, it is proposed to use the deniable encryption mechanisms, which allows its managers to object to the existence of data, ensure their confidentiality and protect the information managers from brute force using by the intruders to obtain key information, during the unauthorized access to information. The article outlines the approach to the use of the existing deniable encryption algorithms for the protection of large data arrays.

Objective. The main purpose of this researching is a hypothesis test that it's possibile to use deniable encryption algorithms for large data arrays protection, because all algorithms in this direction are asymmetric and not adapted to work with "big data".

DOI 10.15588/1607-3274-2019-1-17

Method. The test of hypothesis is carried out by additional data processing units using in the output deniable encryption algorithm with public key, which based on the extended cryptographic scheme of Rabin [2] and whose structure and features are most suitable for verifying this hypothesis put forward by the author.

Results. According to the experiments result, the authors proposed a prototype of deniable encryption algorithm with block encryption of data implemented, which also preserves the deniability mechanism features from the original deniable encryption algorithm. Besides, all changes in algorithm which were proposed by the authors provide the productivity increasing compared with existing approaches [3–6].

Conclusions. Authors have solved a problem of using the existing deniable encryption algorithms, for the large data arrays security, for example for the deniable encryption algorithm with open key, which based on the Rabin extended cryptographic scheme. The proposed approach of the hybrid algorithm constructing with deniable mechanism demonstrates not only the preservation of the basic algorithm properties, but also good performance of the any size data block encrypting ability. Proposed algorithm allows to solve not only the problem of ensuring data confidentiality during the unauthorized access to them, but also makes its suitable for practical using.

KEYWORDS: block encryption, brute force, deniable encryption, information and telecommunication system, denability, unauthorized access, data processing, pseudo-likelihood, public data, recognition, extended cryptographic schema Rabin, secret data, static data.

REFERENCES

1. Analiticheskij centr kompanii InfoWatch [Elektronnij resurs]. Moscow, InfoWatch, 2017–2018. Rezhim dostupu: www.infowatch.ru/analytics/digest/19546.
2. Moldovyan N. A., Vajchikauskas M. A. Rasshirenie kriptosxemy Rabina: algoritm otricaemogo shifrovaniya po otkrytomu klyuchu, *Nauchno-texnicheskij centr oboronnogo kompleksa «Kompas», Voprosy zashhity informacii,* 2014, No. 2, pp. 12–16.
3. Goldwasser S., Micali S. Probabilistic encrypton *Journal of Computer and System Sciences*, 1984, Vol. 28, pp. 270–299.
4. Ibrahim H. Receiver-Deniable Public-Key Encryption, *International Journal of Network Security,* 2009, Vol. 8, No. 2, pp. 159–165.
5. Moldovyan N. A., Birichevskij A. R., Mondikova Ya. A. Otricaemoe shifrovanie na osnove blochnyx shifrov, *Informacionno-upravlyayushhie sistemy*, 2014, No. 5, pp. 80–86.
6. Moldovyan A. A., Tatchina Ya. A. Sposoby psevdoveroyatnostnogo blochnogo shifrovaniya, Intellektual'nye texnologii na transporte, 2018, No. 1, pp. 25–30.
7. Canetti R., Dwork C., Naor M., Ostrovsky R. Deniable Encryption, Advances in Cryptology, CRYPTO, 1997, Procedings, pp. 90–104.
8. Bo Meng, Jiang Qing Wang A Receiver Deniable Encryption Scheme, *Proceedings of the 2009 International Symposium on Information Processing (ISIP'09)*, 2009, pp. 254–257.
9. Gal'chenko A. V. Perspektyvy vykorystannja zaperechuvanogo shyfruvannja v galuzi aviaperevezen', *AVIA 2017: HIII Mizhnarodna naukovo-tehnichna konferencija, Kyi'v, 19–21 kvitnja 2017 r, tezy dopovidej*. Kyi'v, NAU, 2017, pp. 24–28.
10. Gal'chenko A. V. Zahyst personal'nyh danyh z vykorystannjam algorytmiv neodnoznachnogo shyfruvannja, *Visnyk ZNU: Matematychne modeljuvannja ta prykladna mehanika*, 2017, No. 2, pp. 19–32.
11. Moldovyan N. A., Mixteev M. S., Nguen K. T. Protokol otricaemogo shifrovaniya po otkrytomu klyuchu, vklyuchayushhij proceduru autentifikacii pol'zovatelej, Voprosy zashhity informacii, 2016, No. 3, pp. 9–15.
12. Moldovyan N. A., Bashirov Z. S., Solnyshkin Zh. A. Protokol potochnogo otricaemogo shifrovaniya s razdelyaemym klyuchom, *Voprosy zashhity informacii,* 2015, No. 3, pp. 27–31.
13. Shnajer B. Prikladnaya kriptografiya: Protokoly, algoritmy, isxodnye teksty na yazyke Si. Moscow, Triumf, 2012, 816 p.

DOI 10.15588/1607-3274-2019-1-17