УДК 004.8 DOi 10.30837/ bi.2020.1(94).11

І. В. Кириченко[1], Г. Ю. Терещенко[2], І. В. Груздо[3]

[1] к.т.н., ст.викладач кафедри програмної інженерії,
Харківський національний університет радіоелектроніки, Україна,
iryna.kyrychenko@nure.ua, ORCID iD: 0000-0002-7686-6439

[2] асистент кафедри програмної інженерії,
Харківський національний університет радіоелектроніки, Україна,
hlib.tereshchenko@nure.ua, ORCID iD: 0000-0001-8731-2135

[3] к.т.н., доцент кафедри програмної інженерії, Харківський національний університет радіоелектроніки,
Україна, irina.gruzdo@nure.ua, ORCID iD: 0000-0002-4399-2367

ЗАСТОСУВАННЯ СИМЕТРИЧНИХ АЛГОРИТМІВ В БЛОКЧЕЙНІ

Актуальність цієї роботи зумовлена тим, що методи та технології віддалених мережевих атак регулярно вдосконалюються, а існуючі алгоритми та системи шифрування не завжди повністю захищають конфіденційну інформацію. Але розвиток технологій блокчейнів, які мають високу криптографічну стабільність, також не стоїть на місці. Ці обставини роблять розробку та впровадження безпечної системи передачі даних із використанням криптографічних підписів із відкритим та приватним ключами дуже актуальними.

БЛОКЧЕЙН, СИММЕТРИЧНІ АЛГОРИТМИ, BLOWFISH, DES

Актуальность данной работы обусловлена тем, что методы и технологии удаленных сетевых атак регулярно совершенствуются, а существующие алгоритмы и системы шифрования не всегда полностью защищают конфиденциальную информацию. Но и развитие технологий блокчейн, обладающих высокой криптографической стабильностью, также не стоит на месте. Эти обстоятельства делают разработку и внедрение системы безопасной передачи данных с использованием криптографических подписей с открытыми и закрытыми ключами очень актуальными.

БЛОКЧЕЙН, СИММЕТРИЧЕСКИЕ АЛГОРИТМЫ, BLOWFISH, DES

The relevance of this work is due to the fact that the methods and technologies of remote network attacks are regularly improved, and existing encryption algorithms and systems do not always fully protect confidential information. But the development of blockchain technologies, which have high cryptographic stability, also do not stand still. These circumstances make the development and implementation of a secure data transmission system using cryptographic signatures with public and private keys very relevant.

BLOCKCHAIN, SYMMETRIC ALGORYTHMS, BLOWFISH, DES

Вступ

Технологія зберігання інформації блокчейн стала досить актуальна в останні роки. Неможливо заперечувати її значимість в суспільстві, в якому збільшується потреба у підтвердженні правдивості і охорони збереженої інформації. Численні фахівці переконані в тому, що ця технологія здатна застосовуватися в багатьох галузях. Технології, засновані на застосуванні блокчейна можуть зробити переворот в концепції урядового управління, економічних послуг та індустрії. Надаються величезні можливості, проте основне питання полягає в їх реалізації.

Блокчейн – це розподілена база даних, в якій можна зберігати будь-які дані або транзакції. У блокчейні зберігається інформація усієї мережі, що складається з персональних комп'ютерів, і в результаті виходить не тільки децентралізований, а й розподілений простір. Отже, ні компанія, ні людина, ні будь-яка інша довірена сторона не є власником цієї мережі. Всі люди можуть користуватися системою і тим самим підтримувати її функціонування, тому одній людині дуже складно зламати або повністю знищити мережу. Користувачі мережі використовують свої персональні комп'ютери для зберігання пучків даних інших користувачів, які називаються блоками («blocks») в хронологічному ланцюзі («chain»), звідси і назва «блокчейн» дослівно «ланцюг з блоків».

Смарт-контракти – головний рушій технології блокчейн, адже тільки з системою, як блокчейн, в якій не можна підробити, змінити та видалити дані, смарт контракти можуть прирівнюватися до офіційних документів. Наразі, у блокчейні використовується лише один алгоритм шифрування у смарт контрактах, бо вважається, що для його взлому необхідно більше декількох сот років на сучасному етапі. Але, ніхто не змушую нас використовувати лише один алгоритм, що як використовувати різні алгоритми, або суміш алгоритмів шифрування, які зможуть дати навіть більшу криптостійкість та швидкість виконання алгоритму [1].

Криптостійкі алгоритми, прийняті в якості національних або світових стандартів, є загальнодоступними. Їх криптостойкость базується на нерозв'язних за прийнятний час математичних задачах. Але реалізація криптоалгоритмів з урахуванням високої швидкодії, відсутності помилок і гарантованого виконання

вимог математичних перетворень – непросте завдання, яким займаються кваліфіковані розробники.

В разі, якщо електронний підпис використовується в критичних додатках (наприклад, для виконання юридично значимих дій), реалізація криптоалгоритмів в обов'язковому порядку проходить процес сертифікації на відповідність вимогам безпеки. Додатково, засоби криптографічного захисту інформації (ЗКЗІ) можуть мати саме різне уявлення: від програмних бібліотек до високопродуктивних спеціалізованих залозок (Hardware Security Module, HSM).

Саме через складність реалізації і регулювання даного виду продукції існує ринок рішень з криптографічного захисту інформації, на якому грають різні гравці. З метою сумісності різних реалізацій, а також спрощення їх вбудовування в прикладне програмне забезпечення, були розроблені кілька стандартів, які стосуються різних аспектів роботи з ЗКЗІ і безпосередньо електронним підписом [2].

2. Експеримент

Кожна система безпеки повинна забезпечувати пакет функцій безпеки, які можуть забезпечити секретність системи. Ці функції зазвичай називають цілями системи безпеки. Ці цілі можна перерахувати за наступними п'ятьма основними категоріями:

– аутентифікація: це означає, що перед надсиланням та отриманням даних за допомогою системи слід підтвердити особу одержувача та відправника;

– секретність або конфіденційність: зазвичай ця функція (функція) – це те, як більшість людей ідентифікують захищену систему, що означає, що лише аутентифіковані люди здатні інтерпретувати зміст повідомлення (дати) і ніхто інший;

– цілісність: цілісність означає, що вміст повідомлених даних гарантується таким, що не містить будь-якого типу змін між кінцевими точками (відправник і одержувач), основна форма цілісності - це контрольна сума пакетів у пакетах IPv4;

– не передача: ця функція передбачає, що ні відправник, ні одержувач не можуть помилково заперечувати те, що вони надіслали певне повідомлення;

– надійність та доступність сервісу: оскільки захищені системи зазвичай піддаються нападу зловмисників, це може вплинути на їх доступність та тип послуг для користувачів, такі системи повинні забезпечувати спосіб надання своїм користувачам якості послуг, які вони очікують [3].

Таблиця 1 містить показники швидкості для деяких найпоширеніших криптографічних алгоритмів. Усі були закодовані в C#, складені з Microsoft Visual Studio (оптимізація всієї програми, оптимізація для швидкості, генерування коду P4) та працювали на процесорі Intel Core I5 2.1 ГГц під Windows 10, підпрограми використовувались для багатопоточного додавання та віднімання. Властивості SSE2 використовувались для багатопоточного множення.

З таблиці 1 можна помітити, що не всі режими були випробувані для всіх алгоритмів. Тим не менш, ці результати добре мають вказувати про те, як повинні виглядати представлені результати порівняння.

Таблиця 1

Результати порівняння

Algorithm	MB/ Seconds	Time spent (milliseconds)	Processed megabytes (2 ^ 20 bytes)
Blowfish	64,386	3,976	256
AES (128-bit key)	61,010	4,196	256
AES (192-bit key)	53,145	4,817	256
AES (256-bit key)	48,229	5,308	256
AES (128) CTR	57,710	4,436	256
AES (128) OFB	52,925	4,837	256
AES (128) CFB	47,601	5,378	256
AES (128) CBC	55,447	4,617	256
DES	21,340	5,998	128
(3DES)DES-XEX3	20,783	6,159	128
(3DES)DES-EDE3	9,848	6,499	64

Також показано, що Blowfish та AES мають найкращі показники серед інших. І обидва, як відомо, мають краще шифрування (тобто сильніші проти атак даних), ніж інші два [4].

На рисунках 1 і 2 наведені результати експериментів, які проводилися на двох різних машинах: I-5 2,66 МГц і I-5 2,4 ГГц.

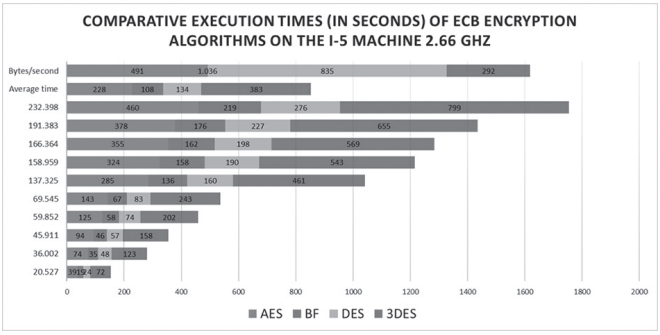

Рис. 1. Порівняльні часи виконання (у секундах) алгоритмів шифрування в режимі ECB на машині I-5 2,66 МГц

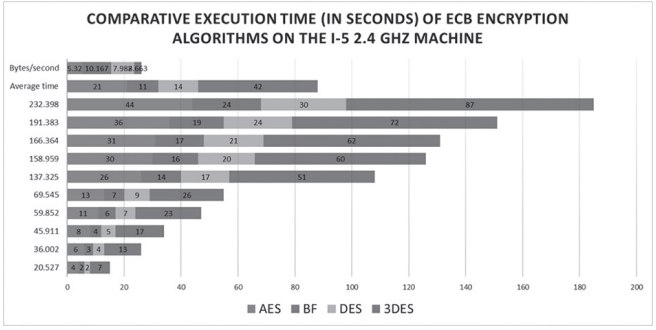

Рис. 2. Порівняльний час виконання (в секундах) алгоритмів шифрування в режимі ECB на машині I-5 2,4 ГГц

З результатів легко помітити, що Blowfish має перевагу перед іншими алгоритмами з точки зору пропускної здатності. Результати показали, що Blowfish має дуже хороші показники порівняно з іншими алгоритмами. Також було показано, що AES має кращі показники, ніж 3DES та DES. Дивовижно це також показує, що 3DES має майже 1/3 пропускної здатності DES, або іншими словами, для обробки однакового обсягу даних йому потрібно 3 рази більше часу, ніж DES.

Були проведені експерименти для порівняння продуктивності різних алгоритмів шифрування, реалізованих всередині .NET Core. Їх результати близькі до показаних на рисунках 3 та 4.

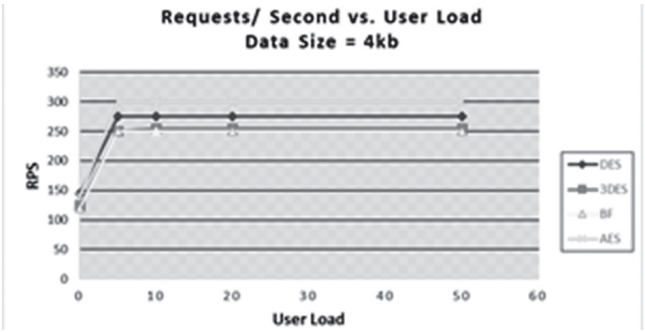

Рис. 3. Результати порівняння

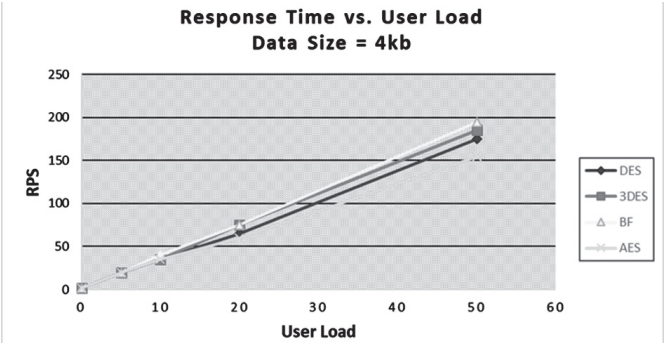

Рис. 4. Результати порівняння

У реалізації реалізовані керовані обгортки для DES, 3DES та Rijndael, доступні в System.Security.Cryptography, які охоплюють некеровані реалізації, доступні в CryptoAPI. Це DESCryptoServiceProvider, TripleDESCryptoServiceProvider і RijndaelManaged відповідно. У System.Security.Cryptography, яка була використана в тестах, є лише чиста керована реалізація Rijndael[5].

У таблиці 2 показані параметри алгоритмів, що використовуються в цьому експерименті.

Таблиця 2

Налаштування алгоритмів

Algorithm	Block size (Bit)	Key size (Bit)
DES	64	64
3DES	64	192
Rijndael	128	259
Blowfish	64	448

3DES та AES підтримують інші налаштування, але ці параметри представляють максимальні параметри безпеки, які вони можуть запропонувати. Більша довжина ключів означає, що потрібно докласти більше зусиль, щоб порушити захист зашифрованих даних.

Оскільки тест оцінювання призначений для оцінки результатів при використанні блок-шифру, через обмеження пам'яті на тестовій машині (1 Гб) тест розбиває блоки даних про завантаження на менші розміри. Дані навантаження поділяються на блоки даних і вони створюються за допомогою класу RandomNumberGenerator, доступного в просторі імен System.Security.Cryptography.

Експерименти проводяться з використанням Intel core i-5 64-бітового процесора з 8 ГБ ОЗУ. Програма моделювання складається за допомогою стандартних налаштувань у .NET 2019 Visual Studio для програм C# Windows. Експерименти були проведені кілька разів, щоб переконатися в тому, що результати є послідовними та справедливими для порівняння різних алгоритмів.

Для оцінки продуктивності порівняних алгоритмів необхідно визначити параметри, для яких алгоритми повинні бути протестовані.

Оскільки особливості безпеки кожного алгоритму як його сили проти криптографічних атак вже відомі та обговорюються. Вибраним фактором для визначення продуктивності є швидкість алгоритму для шифрування/дешифрування блоків даних різного розміру [6].

Розглядаючи різні розміри блоків даних (від 0,5 МБ до 20 МБ), алгоритми оцінювались з точки зору часу, необхідного для шифрування та дешифрування блоку даних. Усі реалізації були точними, щоб переконатися, що результати будуть відносно справедливими та точними.

Програма приймає три входи: алгоритм, режим шифрування та розмір блоку даних. Після успішного виконання відображаються дані, що генеруються, шифруються та розшифровуються. Зауважте, що більшість персонажів не можуть з'являтися, оскільки вони не мають представлення символів. Ще одне порівняння проводиться після успішного процесу шифрування/дешифрування, щоб переконатися, що всі дані обробляються правильним шляхом, порівнюючи згенеровані дані (оригінальні блоки даних) та дешифрований блок даних, що генерується в процесі.

Перший набір експериментів проводили в режимі ECB, результати показані на рисунку 5 Результати показують перевагу алгоритму Blowfish над іншими алгоритмами з точки зору часу обробки. Це також показує, що AES споживає більше ресурсів, коли розмір блоку даних порівняно великий. Результати, показані тут, відрізняються від результатів, отриманих раніше, оскільки розміри блоків даних значно більше, ніж ті, що використовуються в експерименті [7].

Тут також можна помітити, що 3DES вимагає завжди більше часу, ніж DES через свою

характеристику трифазного шифрування. Blowfish, хоча він має довгий ключ (448 біт), перевершує інші алгоритми шифрування. DES і 3DES, як відомо, мають черв'якові отвори у своєму захисному механізмі, Blowfish та AES, з іншого боку, поки що не мають.

Ці результати не мають нічого спільного з іншими навантаженнями на комп'ютер, оскільки кожен окремий експеримент проводився кілька разів, в результаті чого був досягнутий майже однаковий очікуваний результат. Впровадження DES, 3DES та AES в .NET вважається найкращим.

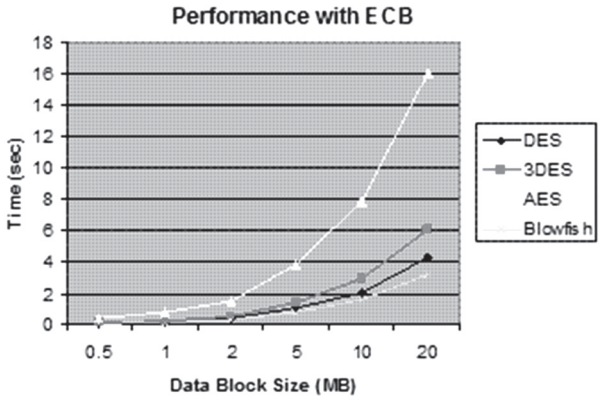

Рис. 5. Результати роботи в режимі ECB

Як очікується, CBC вимагає більше часу на обробку, ніж ECB, через його ключовий характер. Результати, показані на рисунку 6 також вказують на те, що додатковий час, що додається, не є важливим для багатьох застосувань, знаючи, що CBC значно кращий, ніж ECB з точки зору захисту. Різницю між двома режимами важко помітити неозброєним оком, результати показали, що середня різниця між ECB та CBC становить 0,059896 секунди, що порівняно невелика [8].

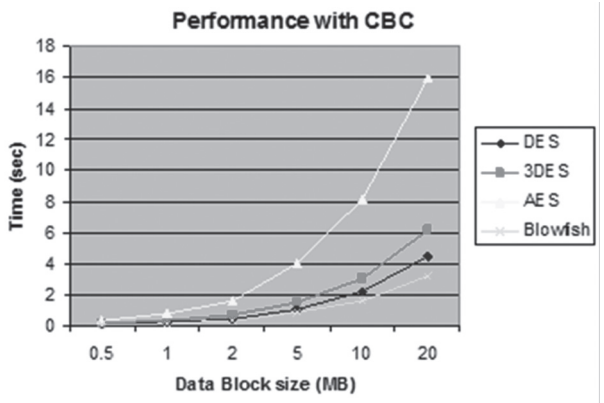

Рис. 6. Результати роботи в режимі CBC

AES показав низькі результати роботи порівняно з іншими алгоритмами, оскільки він вимагає більшої потужності для обробки. Використання режиму CBC додало додаткового часу на обробку, але в цілому це було відносно незначним, особливо для певного застосування, яке вимагає більш безпечного шифрування до відносно великих блоків даних.

На підставі усіх вищеотриманних даних, було складено таблицю 3 з усіма алгоритмами, які можуть використовуватися з технологією блокчейн та надано коротку характеристику кожного з них [9].

Представлені результати моделювання показали, що Blowfish має кращу продуктивність, ніж інші поширені алгоритми шифрування, що використовуються. Оскільки Blowfish досі не має жодних відомих слабких місць безпеки, що робить його відмінним кандидатом, щоб його розглядали як стандартний алгоритм шифрування[10].

Висновки

Нинішній час вимогливо ставиться до технічних рішень, особливо до питання приватності та анонімності даних, в тому числі коли справа стосується фінансової діяльності. Блокчейн технологія і справді є революційною та дає можливість не модифікувати старе ставлення, а подивитися на питання взаємовідносин користувачів та сторонніх інстанцій абсолютно під іншим кутом.

Раніше завдання полягало в тому, як знайти для власної системи найбільш надійних постачальників ресурсів, регуляторів та верифікаторів, тепер – яку форму блокчейн технології використовувати, адже його спільнота ефективно заміщає всі вище перелічені ролі в системі.

Для цього не потрібно будувати модель з довершеною технологією, бо під час реалізації на практиці все одно прийдеться чимось поступатися на перевагу пріоритетним характеристикам. Саме тому обирається синтез адаптивних підходів, що передбачає:

— відокремлення бізнес-стратегій та сфер, де блокчейн не може бути використана чи застосована тільки частково з огляду на певні ліміти технології;

— узагальнене та ретельне оформлення вимог користувачів вищезгаданих систем;

— будову нових моделей блокчейну, які в тих чи інакших умовах зможуть забезпечувати максимум вимог користувачів.

Незважаючи на вже реалізовані алгоритми збереження приватності, є щонайменше чотири різних аналізи, які розроблені для виявлення конференційної інформації в середовищі цієї криптосистеми. Цей аналіз було успішно здійснено внаслідок прозорості даних у блокчейні, а також питанню ліквідності та ідентифікації поведінки користувачів. Проте потрібно зробити так, щоб система водночас ефективно працювала і при цьому задовольняла широке коло вимог користувачів.

Нові підходи, що орієнтовані не тільки на збереження конфіденційності клієнтів, а й на поліпшення технології блокчейн взагалі, а саме: підвищення децентралізації мережі, швидкодію транзакцій, можливість мікро-операцій тощо. Важливо пам'ятати, що поняття приватності не статична та фіксована річ. Це

Таблиця 3

Порівняння криптографійних алгоритмів

Алгоритм	Ким створений	Рік	Розмір ключа	Розмір блоку	Раунд	Структура	Складність	Функції
DES	IBM	1975	64 бітів	64 бітів	16	Фейстель	Ні	Не достатньо сильний
DH	Вітфілд Діфф і Мартін Гелман	1976	Змінна	—	—	Фейстель	—	Хороша безпека та швидкість
E-DES	IBM	1977	1024 бітів	128 бітів	16	Фейстель	—	Хороша безпека та швидкість
RSA	Рівест Шамір Адлеман	1977	1024 - 4096	128 бітів	1	Алгоритм відкритого ключа	Ні	Відмінна безпека та низька швидкість
T-DES	IBM	1978	112 або 168	64 бітів	48	Фейстель	Так	Відмінна безпека та швидкість
ECC	Ніл Кобліц та Віктор Міллер	1985	Більше, ніж симетричні та змінні	Змінна	1	Алгоритм відкритого ключа	Так	Відмінна безпека та швидкість
EEE	Тахер Ельгамал	1985	1024 бітів	—	—	Алгоритм відкритого ключа	Так	Досить забезпечний і швидкісний
RC4	Рон Рівест	1987	Змінна	40-2048	256	Фейстель потік	Так	швидкий шифр
RC2	Рон Рівест	1987	8, 128, 64 за замовчуванням	64 бітів	16	Фейстель	—	Гарна та швидка безпека
BLOWFISH	Брюс Шнайер	1993	32-448	64 бітів	16	Фейстель	Так	Швидкий шифр в SSL
SEAL	Філіп Рогавей та Дон Коппер-Сміт	1994	160 бітів	32 бітів	2	Алгоритм відкритого ключа	Так	Не сильна безпека і швидка швидкість
DSA	NIST	1997	Змінна	—	—	Алгоритм відкритого ключа	Так	Хороша безпека та швидкість
RC6	Рон Рівест та ін	1998	128 біт до 256 біт	128 бітів	20	Фейстель	Так	Хороша безпека
AES	Джоан Дейман та Інсент Ріджмен	1998	128,192,256 бітів	128 бітів	10, 12, 14	Перестановка заміни	Так	Безпека відмінна. Це найкраще в забезпеченні безпеки та шифрування

ціль, за яку постійно ведеться боротьба між криптографами та зловмисниками. Абсолютної приватності не існує. Тому наша задача – адаптуватися до технологічного середовища, яке змінюється, та пропонувати нові алгоритми й підходи, що будуть забезпечувати надійність та ефективність роботи системи.

Список літератури:

[1] Асиметрична схема. URL: https://helpiks.org/4-30251.html (дата звернення: 06.03.2020).

[2] Використання хеш-функцій. URL: https://helpiks.org/4-30249.html (дата звернення: 15.03.2020).

[3] Про електронний цифровий підпис. // База даних «Законодавство Уркаїни» / ВР України. URL: https://zakon.rada.gov.ua/laws/show/852-15 (дата звернення: 15.03.2020).

[4] Ефект від впровадження wms. URL: https://stud.com.ua/172359/logistika/efekt_vprovadzhennya (дата звернення: 10.03.2020).

[5] *Єлізаров А. Б.* Аналіз алгоритмів електронно-цифрового підпису підчас передачі інформації. URL: http://www.rusnauka.com/22_AND_2016/Informatica/4_215303.doc.htm (дата звернення: 03.03.2020).

[6] *Кесавулу Р.* Анализ производительности криптографических алгоритмов в информационной безопасности. URL: https://www.ijert.org/performance-analysis-of-cryptographic-algorithms-in-the-information-security (дата звернення: 01.04.2020).

[7] *Клімушин П.* С. Механізми забезпечення довіри в національній системі електронних цифрових підписів. URL: http://www.kbuapa.kharkov.ua/e-book/tpdu/2013-2/doc/1/08.pdf (дата звернення: 05.04.2020).

[8] Комп'ютерна криптографія. URL: http://dspace.tneu.edu.ua/retrieve/ 49411/%D0%9B%D0%B5%D0%BA%D1%86%D1%96%D1%97_%D0%9A%D0%9A.pdf (дата звернення: 05.04.2020).

[9] Криптоаналіз блочних та потокових шифрів. URL: https://infopedia.su/8xb3f.html (дата звернення: 15.04.2020).

[10] *Лапіна М. А.* Правове регулювання відносин у сфері електронного документообігу. URL: http://yport.inf.ua/pravovoe-regulirovanie-otnosheniy-sfere.html (дата звернення: 15.04.2020).

Надійшла до редколегії 03.04.2020