УДК 004.056.55 DOI: 10.26565/2519-2310-2022-1-03

ДОСЛІДЖЕННЯ ЗАСТОСУВАННЯ АЛГОРИТМІВ МАЛОРЕСУРСНОЇ КРИПТОГРАФІЇ У ДЕЦЕНТРАЛІЗОВАНИХ СЕРЕДОВИЩАХ

Євгеній Деменко, Олексій Нарєжній

Харківський національний університет імені В.Н. Каразіна, Харків, 61022, Україна
xa11868404@student.karazin.ua, o.nariezhnii@karazin.ua

Надійшла:червень 2022. Прийнята: липень 2022.

Анотація: *Метою даного матеріалу є ознайомлення з областю досліджень застосування малоресурсних алгоритмів криптографії для систем Інтернету речей (IoT) та можливості прямого впровадження в децентралізованих системах. За останні кілька років Інтернет речей став однією з найважливіших технологій століття. Зараз людство досягло високого рівня розвитку технологій, що дозволяє налаштовувати взаємодію між пристроями та створювати безперебійний зв'язок між людьми, процесами та речами. З появою 5G технології, IoT стали центром розвитку майже для всіх сучасних галузей. Пристрої в цій архітектурі значно менші та мають низьке енергоспоживання. Звичайні алгоритми шифрування, як правило, дорогі в обчислювальному плані через їхню складність і вимагають багато раундів, однак це може поставити під загрозу бажану цілісність. Криптографія з низьким ресурсом — це компроміс між вартістю впровадження, швидкістю, безпекою, продуктивністю та енергоспоживанням на пристроях IoT. Мотивація полегшеної криптографії полягає в тому, щоб використовувати менше пам'яті, менше обчислювальних ресурсів і менше енергоспоживання, щоб забезпечити рішення безпеки, яке може працювати на пристроях з обмеженими ресурсами. Блокові шифри мають фіксовану довжину бітів і різні кроки перетворення, які визначаються симетричним ключем. Блокові шифри дуже універсальні, що дуже корисно з точки зору IoT. Ще одна перевага полягає в тому, що цей процес має майже ідентичні методи шифрування та дешифрування. Тому його можна реалізувати з меншими ресурсами.*

Ключові слова: *малоресурсна криптографія; Інтернет речей; блокові шифри; IoT.*

1. Вступ

Сьогодні широкосмуговий Інтернет є загальнодоступним, а вартість підключення постійно знижується. Як наслідок, все більше гаджетів та різних датчиків підключаються до всесвітньої мережі Інтернет [1-2]. Всі ці тенденції створюють сприятливе підґрунтя для розвитку Інтернету речей (IoT). Однак, навколо Інтернету речей існує багато складнощів [3]. Перш за все це зв'язано зі складністю, як елементної бази, так і спеціальних алгоритмів обробки даних, що реалізуються в IoT пристроях. Процес обміну такою великою кількістю даних починається з самих пристроїв, які повинні безпечно взаємодіяти з платформою [3].

Пристрої, з яких складається система *Інтернет Речей* - це будь-який фізичний об'єкт, який можна унікально ідентифікувати (за допомогою URI або унікального ідентифікатора ресурсу) і який може надсилати/отримувати дані шляхом підключення до мережі [1]. Прикладами є транспортні засоби, промислові контролери, *RFID*-мітки, сенсорні вузли, смарт-карти, побутова техніка, тощо [2]. Вони можуть бути з'єднані між собою, з центральним сервером/мережею серверів або через хмарні сервіси.

Сутність Інтернету речей - зв'язок та обмін інформацією [1]. Однак, всі ці дані не генеруються лише для того, щоб їх десь зберігати і «забути про їх існування». Основна ціль їх використання, це автоматизація. IoT практично «стирає» розрив між цифровим та фізичним світом, однак має і зворотний бік процесу - компрометація IoT пристроїв, може мати небезпечні наслідки в «реальному» світі.

Як правило пристрій Інтернету речей, містять один або декілька датчиків, які використовуються для збору даних та підтримки мережевих інтерфейсів. Тип і номенклатура даних, що збирають ці датчики, залежать від конкретного пристрою і їх функціонального завдання. Прицьому, всі накопичені дані, та дані телеметрії (технологічна інформація) інтенсивно

циркулють в межах відповідної мережі пристроїв, що обумовлює актуальність питань забезпечення безпеки обміну інформацією.

Існуючі в даний час платформи IoT використовують переважно централізовану модель, згідно з якою вони виступають в якості «брокерів» або концентраторів для управління обміном даними між пристроями IoT [3]. Однак, багато досліджень свідчать, що IoT повинен використовувати насамперед децентралізовану модель для забезпечення безпечного обміну даними. При цьому ключовими проблемами реалізації традиційної криптографії в пристроях IoT вважаються наступні [4]:

- низький рівень наявної обчислювальної потужності (*або відсутність батареї у випадку пасивних RFID-міток*);
- обмеженість ресурсів наявної пам'яті IoT пристроїв ;
- невелика фізична площа для реалізації збірки;
- низький заряд батареї (або навпаки її відсутність);
- реакція в реальному часі.

Малоресурсна або ж *легка криптографія* є компромісом між такими категоріями, як вартість реалізації, швидкість, безпека, продуктивність та енергоспоживання на пристроях з обмеженими ресурсами. При цьому, мотивація для використання малоресурсної криптографії полягає у використанні меншого обсягу пам'яті, менших обчислювальних ресурсів та меншого енергоспоживання заради забезпечення безпеки [5].

2. Класифікація та застосування малоресурсних криптографічних примітивів

За останнє десятиліття було запропоновано низку малоресурсних криптопримітивів, які мають переваги у продуктивності порівняно з стандартними криптографічними стандартами. Ці примітиви відрізняються від звичайних алгоритмів припущеннями, що малоресурсні примітиви не призначені для широкого кола застосувань і можуть накладати обмеження на потужність зловмисника.

Малоресурсна криптографія - це розділ криптографії, метою якого є розробка алгоритмів для використання в пристроях, які не здатні забезпечити більшість існуючих кодів і мають достатні ресурси (пам'ять, потужність, розмір) для роботи [5]. Хорошо відомі чотири типи малоресурсних криптографічних примітивів, які доступні для використання:

- Малоресурсні блокові шифри (LWBC);
- Малоресурсні потокові шифри (LWSC);
- Малоресурсні хеш-функції (LWHF);
- Криптографію еліптичних кривих (ECC).

Основними факторами, за якими можна проаналізувати малоресурсні криптографічні примітиви є: розмір блоку, розмір ключа, структура та кількість раундів. ECC є ще одним із варіантів малоресурсної криптографії, причому, будучи асиметричним шифром, він має можливість забезпечувати автентифікацію та неспростування.

Властивості малоресурсної криптографії обговорювалися в *ISO/IEC 29192* в *ISO/IEC JTC 1/SC 27. ISO/IEC 29192* є новим проектом зі стандартизації малоресурсної криптографії, і проект знаходиться в процесі стандартизації. У стандарті ISO/IEC 29192 властивості малоресурсності описуються на основі цільових платформ.

Дотричаючись завдань проектування, малоресурсні алгоритми використовують зазвичай менші розміри блоків - 32, 48 або 64 біт, ніж звичайний шифр, який має більший розмір блоків - 64 або 128 біт [6]. Малоресурсні алгоритми застосовують менші розміри ключів, (менше 96 біт). Найменший розмір ключа, за даними *NIST*, становить 112 біт [6]. У стандарті *ISO/IEC 29192* [6] детально описані властивості малоресурсності, що

встановлюються на цільових платформах. По-перше, легкість апаратних засобів оцінюється за розміром мікросхеми та їх енергоспоживання і, по-друге, за обсягом потрібної пам'яті.

Поряд з продуктивністю та вартістю, безпека є невід'ємним показником для будь-якого алгоритму малоресурсної криптографії. Властивість стійкості до атак будь-якого алгоритму малоресурсної криптографії може бути виміряна за допомогою криптоаналізу. Сенс криптоаналізу заключається в пошуці слабких місць алгоритму та розробку методів дешифрування [7]. Існує чотири основних типи атак на блоковий шифр [8]: – диференціальний криптоаналіз; – лінійний криптоаналіз; – інтегральний криптоаналіз; – алгебраїчні атаки. Ці атаки базуються на використанні «відомого відкритого тексту», «тільки шифрованого тексту», «обраного шифрованого тексту», «обраного відкритого тексту», а також атаки «людина посередині», атаки «грубою силою» та атак «побічного каналу» [8]. Крім того криптографію розділяють на дві основні напряму: симетричні та асиметричні шифри. Відповідно, у табл. 1 наведено порівняння, яке дозволяє продемонструвати різницю між асиметричною та симетричною криптографією [9].

Таблиця 1 – Порівняння методів криптографії

Параметр	**Особливості різновидів реалізації**	
	Криптографія з симетричним ключем	Криптографія з асиметричним ключем
Ключ	Один загальний приватний ключ	Унікальна пара приватного та публічного ключів. Генерація відкритих ключів залежить від криптографічних алгоритмів, заснованих на односторонніх математичних функціях.
Кількість ключів	Експоненційно пропорційні кількості користувачів	Лінійно пропорційні кількості користувачів
Швидкість та складність	Це прості алгоритми, завдяки цьому процес шифрування може бути здійснений швидко.	Це набагато складніший процес, ніж шифрування з симетричним ключем, і він відбувається повільніше через те, що для використання різних ключів потрібно більше часу.
Апаратна складність	Використовує алгоритми що потребують відносно недорогого апаратного забезпечення.	Більш складна реалізація апаратного забезпечення, яка обчислює важкі алгоритми які потребують більш потужне апаратне забезпечення.
Використання	Здебільшого використовується, для передачі великих обсягів даних.	Використовується в невеликих транзакціях, в першу чергу для автентифікації та встановлення безпечного каналу зв'язку перед фактичною передачею даних.
Алгоритми	RSA, DSA, ECC	Stream cipher: Trivium, Chacha, WG-8, Espresso, Grain 128. Block Ciphers: AES, DES, 3DES, Blowfish, Twofish, Curupira, PRESENT, KATAN. TEA, Humming Bird, RECTANGLE, SIMON

В межах даного матеріалу увага зосередженна, перш за все, на криптографії з симетричним ключем, яка може має можливість широко застосовуватися на пристроях, що піддаються жорстким ресурсним обмеженням [5]. В свою чергу, асиметричні шифри набагато вимогливі до обчислювальних ресурсів, ніж їх симетричні альтернативи.

Криптографія з симетричним ключем складається з основних функцій, таких як блокові або потокові шифри, а також методів застосування основної функції до пакету, яку носять назву режимом роботи блокового шифру для автентифікації чи шифрування [9]. Зусилля щодо криптографічній стандартизації малоресурсних примітивів розглядають як програмний, так і апаратний аспекти безпеки, котрі, зазвичай, мають, різні метрики. Програмні метрики включають цикли, пам'ять і цикл на байт, тоді як апаратні метрики враховують пропускну здатність, площу, співвідношення по всій площі. Тому важко отримати пряме порівняння між цими двома показниками [6].

Симетричне шифрування використовує один і той же ключ, як для шифрування, так і для розшифрування даних. Цей метод шифрування є безпечним і відносно швидким. Його основним недоліком є спільне використання ключа двома сторонами, що спілкуються. Зловмисник може розшифрувати дані, якщо має доступ до ключа. Алгоритми з симетричним ключем забезпечують конфіденційність і цілісність даних, але не гарантують автентифікацію [9]. Цей тип шифрування використовує три типи алгоритмів, заснованих на хешуванні, потоковому та блоковому шифрах.

3. Малоресурсні блокові шифри

Симетрична шифрування допомагає при проектуванні однієї і тієї ж схеми для шифрування і дешифрування з мінімальними витратами.

Блокові шифри - різновид симетричних шифрів, в яких обробляється відразу весь блок. Блокові шифри використовуються для побудови хеш-функцій та кодів автентифікації повідомлень (MAC) [10]. Полегшені блокові шифри базуються на двох типах структур: Мережа підстановки-перестановки (SPN) та Фейстеля.

Мережа Фейстеля (FN) - це багатораундовий шифр, який ділить вхідний блок на дві частини і працює тільки над половиною (дифузія) в кожному раунді шифруванні або дешифруванні. Між раундами ліворуч і праворуч половини блоку міняються місцями. Структура Фейстеля використовує свою кругову функцію лише на половині стану [10].

Таким чином, головною перевагою структури Фейстеля є використання одного і того ж програмного коду для процесу шифрування та дешифрування. Це зумовлює низьке використання пам'яті. Вона може бути реалізована на апаратних засобах з низькою середньою потужністю. Фейстелівська структура не підходить для конструкцій з малою затримкою. SPN є більш швидким, але без розкладу ключів. Відсутність ключового розкладу робить вразливим до атак. При однаковій величині запасу стійкості та однакових витратах енергії, структура SPN є більш придатною, оскільки вона вимагає меншого раунду виконання. За аналогічних умов SPN матиме менші енерговитрати.

PRESENT та *CLEFIA* - єдині два алгоритми, що затверджені стандартом ISO/IEC 29192 [11,12].

AES є класичним прикладом алгоритму на основі SPN, працює на 128-бітному блоці з 128, 192 та 256-бітними варіантами ключів [13]. Мінімальна вимога еквівалентів воріт (GE), зафіксована для AES, становить близько 2400 GE *(на 23% менше, ніж звичайна)* [13-14], що все ще є важким для деяких невеликих додатків у реальному часі. Це показує порівняно ефективну продуктивність при забезпеченні додатковими ресурсами.

Основними параметрами для оцінювання малоресурсного блочного шифру є розмір ключа, розмір блоку, тип структури та кількість раундів [5]. Малоресурсний шифр повинен відповідати чотирьом вимогам:

- Мінімальна площа кремнію або обсяг пам'яті;
- Низьке енергоспоживання;

- Менша кількість еквівалентів воріт (*GE*) для ефективної апаратної реалізації;
- Достатній рівень безпеки.

RFID-мітки можуть мати близько 1000-10000 GE, з яких можуть бути доступні 300-2100 GE для аспектів безпеки [15]. Для впровадження відповідних рішень у сфері забезпечення інформаційної безпеки (ІБ), загальна кількість доступних GE становитиме приблизно 2000-3000. При цьому, блокові шифри мають бути обмежені меншою кількістю GE для того, щоб відповідати малоресурсним додаткам.

AES, PRESENT та CLEFIA – це три шифри є обов'язкові для вибору. Слід підкреслити, що AES є найбільш широко використовуваним шифром, оскільки був встановлений в якості стандарту для шифрування в 2002 році. Він використовується багатьма пристроями IoT, незважаючи на те, що він не є малоресурсним шифром. PRESENT та CLEFIA - це два шифри, які також були стандартизовані, але як малоресурсні шифри. Характеристики цих шифри можна використовувати в якості «опорних» при оцінці властивостей інших шифрів. Наприклад, відомо [12], що для шифру CLEFIA автори змінили розмір S-box з 4-х біт до 8-ми біт, так щоб досягти кращих результатів при виконанні на програмному забезпеченні. В роботі [16] оптимізовано шифри-фіналісти конкурсу AES, же основний акцент зроблено на зменшенни обсягів займаної пам'яті за рахунок зменшення розміру коду з використанням функцій заміни макросів та інших повторень коду.

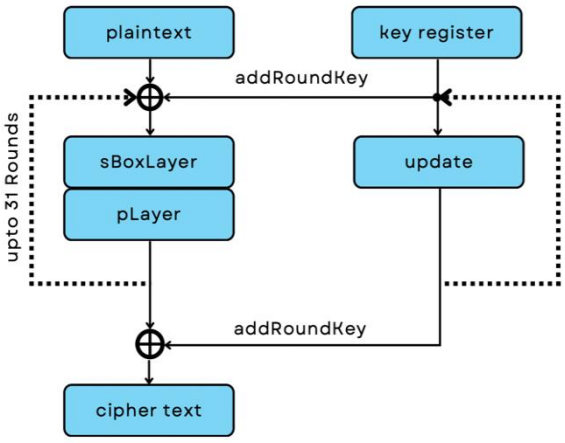

Рис. 1 – Алгоритмічний опис *PRESENT*

Використання AES призводе до високих GE [13], що робить їх нездійсненними для невеликих додатків, що працюють в реальному часі. Альтернативим рішенням для модифікації існуючого блокового шифру і створення ефективної апаратної моделі, є структура «*PRESENT*» [11]. Відповідний алгоритм (*див. рис. 1*) впроваджує малоресурсний блоковий шифр, та є більш меншим, ніж алгоритм AES. Розмір блоку процедур - 64, розмір ключа - 80 або 128, розмір S-box - 4. Один блок даних шифрується (розшифровується) за 31 раунд. Вихідні дані додатково розбиваються на блоки по 8 біт, де 4 біти цього значення складають стовпець, а ще чотири - рядок. Таким чином, вхідне значення замінюється на значення в S-Box.

Вихід з S-Box подається в блок перестановок, де біти переставляються місцями. Таким чином реалізовано 31 раунд обчислень.

Процес планування ключа буде оновлювати ключ для кожного раунду. Розшифрування виконується у зворотному порядку з інверсією S-Box.

В свою чергу, корпорація SONY представила CLEFIA, що стандартизована NIST у 2007 році. Вона базується на структурі Фейстеля і використовує 128-бітний блок з можливістю вибору ключа 128, 192, 256 біт через 18, 22, 26 раундів відповідно [12,17]. Дана реалізація демонструє високу продуктивність і стійкість до різних атак при порівняно високій вартості, оскільки найкомпактніша версія вимагає 2488 GE (тільки шифрування) для 128-бітового ключа [18]. CLEFIA має відповідні властивості [18], що робить його більш стійким до різних атак, але в той же час, це вимагає більшого об'єму пам'яті та обмежує його використання в надмалих додатках.

3.1 Порівняння методів малоресурсної криптографії

FELICS – система бенчмаркінгу з відкритим вихідним кодом, призначену для об'єктивного та послідовного оцінювання програмних реалізацій малоресурсних криптографічних примітивів для вбудованих пристроїв [19]. Фреймворк є довільно гнучким завдяки своїй модульній структурі, що дозволяє легко інтегрувати нові метрики, цільові пристрої та сценарії оцінювання. Вона складається з двох модулів, які в даний час можуть оцінювати продуктивність малоресурсних блокових і потокових шифрів на трьох широко використовуваних мікроконтролерах: 8-бітному AVR, 16-бітному MSP та 32-бітному ARM. FELICS має відносно простий користувальницький інтерфейс і призначений для використання розробниками шифрів для порівняння нових примітивів із існуючими. При цьому, отримані метрики є досить детальними та допомагають розробникам у виборі найкращого рішення, такого, що відповідає вимогам конкретного застосування.

Слід відмітити, що FELICS має реалізацію PRESENT, однак, вона не є вдалою [19]. Тому, в межах проведеного аналізу, було обрано 32-бітну реалізацію [11]. Ця реалізація була згодом оптимізована за допомогою 3-ох різних методів. В першу чергу, 4-розрядні S-box були замінені на 8-розрядні S-box для покращення продуктивності програмного забезпечення. Потім були розгорнуті всі перимутації через їх надмірну вартість для програмних реалізацій. Крім того, цикли були також повністю розгорнуті, щоб усунути всі залежності і підштовхнути шифр-код до найшвидшого рівня продуктивності. В кінці, звернення до пам'яті було зведено до мінімуму за рахунок утримання стану шифру в регістрах процесора протягом більшої частини часу виконання шифру.

Еталонна реалізація була отримана з сайту CLEFIA та адаптована до фреймворку FELICS [12, 17]. Еталонний алгоритм обчислює константи, які використовуються в планувальнику ключів, що призводить до невиправдано високого часу виконання. Тому була розроблена альтернативна версія 32-бітної реалізації, яка має попередньо обчислені значення всіх констант, що зберігаються в таблиці. Решта сім реалізацій, що були розроблені, експлуатують використання T-box. У той час як в цих реалізаціях застосовуються стандартні T-box для 8-бітового орієнтованого еталонного алгоритму та його оптимізованої 32-бітової орієнтованої версії відповідно, всі інші реалізації використовують скорочені T-box. Також, варто відмітити, що потокові шифри широко досліджуються в криптографічному середовищі через більш швидке виконання, але вони є вразливими до атак у порівнянні з блоковими шифрами.

В табл. 2 наведено стислі відомості щодо результатів порівняння деяких параметрів AES, PRESENT, CLEFIA та DES. Всі розглянуті шифри були реалізовані на модулі FELICS, для умов використання 8-розрядних мікроконтролерів AVR (сімейство 8-розрядних RISC мікроконтролерів, 100 kHz) [20].

Як і очікувалось, стандартизовані реалізації показують доволі повільні результати, за винятком PRESENT [11], де найповільніший час виконання має реалізація з невеликим обсягом коду. Це свидчить про те, що більшість оптимізацій дозволяє покращити час виконання навіть тоді, коли основна увага приділяється зменшенню розміру коду. PRESENT є «недружнім» малоресурсним шифром при націленості на програмні реалізації і тому, навіть з урахуванням декількох удосконалень та доопрацювань, все ще залишається дуже «важким» для програмного забезпечення (особливо для умов мобільних платформ). Ще одним результатом є те, що для збалансованих шифрів час виконання близький до 1000 тактів майже для кожного шифру, і лише PRESENT дає далекі від цього значення. Він має біт-орієнтовані перестановки, які важко обчислювати в програмному забезпеченні [11]. В той же час, як AES та CLEFIA підтримують блоки в 128 біт [12,17].

Таблиця 2 – Результати порівняння

Алгоритм	Алгоритм проектування шаблону	Розмір вхідного блоку	Розмір ключа	Кількість раундів	Площа (GEs)	Пропускна здатність (Kbps)	*Особливості*
AES	SPN	128	128	1,032	5440	15.53	*Високий рівень безпеки, гнучкість.*
PRESENT	SPN	64	80	31	10579	201.53	*Менша кількість воріт, менше пам'яті. Доцільний для шифрування невеликих обсягів даних.*
CLEFIA	GFN	128	128	36	27738	360.44	*Висока продуктивність та стійкість до різних атак.*

CLEFIA - алгоритм шифрування, котрий має серед досліджуваних алгоритмів найбільшу довжину блоку з довжиною блоку 128 біт [12,17], в той час як в інших алгоритмах перевага віддається довжині блоку 64 біт. Це важливо для пристроїв, що обмінюються даними в мережі Інтернет з об'єктами малої ємності. Ефективніше шифрувати блоки невеликого розміру, а також ті, що застосовують архітектуру Фейстеля. З іншого боку, збільшення розміру ключа знижує енергоефективність.

Звичайно, чим більший розмір ключа, тим краще забезпечується безпека. Однак, в умовах роботи IoT, більш вдалими слід вважати ключі від 80 біт до 128 біт (принаймні поки). Вибір структур простим способом, який не потребує занадто багато енергії, підвищує ефективність. Енергоємні структури, такі як процеси редукції та змішані удари, що використовуються в алгоритмах CLEFIA підтверджує цю ситуацію.

PRESENT має певну перевагу над CLEFIA: - нелінійний S-box використовує 4-бітову структуру, що призводить до меншого GE і меншого енергоспоживання. Додаткові властивості S-box допомагають PRESENT досягти бажаного лавинного ефекту, а результати роботи [11] свідчать про те, що PRESENT має компактний S-box. Також в PRESENT є 16 S-box, які розділені на чотири групи. Деякі важливі відмінності цих S-box наведені нижче [11]:

1. Вхідний біт до S-box надходить з 4 чітко визначених S-box тієї ж групи.
2. Вхідні біти до групи з чотирьох S-box надходять з 16 різних S-box.
3. Чотири вихідні біти з певного S-box надходять у чотири чітко визначені S-box, кожен з яких належить до окремої групи S-box у наступному раунді.
4. Вихідні біти S-box у різних групах подаються до різних S-box.

Апаратна реалізація AES для IoT, з точки зору ІБ, може залучати деякі апаратні атаки. Тому важливо спостерігати за цими спробами і своєчасно знаходити потрібні рішення. В цілому, AES та CLEFIA є двома найбільш вдалими прикладами шифрів, які витрачають багато ресурсів (на розмір коду) для досягнення їх більш швидкої роботи.

4. Висновки

Зростання масштабів використання IoT, обумовлює потребу в більш широкому запровадженні механізмів (алгоритмів) малоресурсного шифрування.

Для пристроїв із певними ресурсними обмеженнями, наявні стандарти криптографічних алгоритмів можуть бути занадто складними та/або занадто енерговитратними. Крім того, кіберзлочинці можуть скористатися недоліками паролів, які відносно легко підбираються, якщо немає жорстко декларованих вимог до паролів, які створюються користувачами.

Для пристроїв з обмеженими ресурсами, зокрема пристроїв IoT, малоресурсна криптографія є ефективним напрямом забезпечення безпеки їх мережевої взаємодії.

Спираючись на результаті проведеного аналізу, можна констатувати, що AES, PRESENT та CLEFIA, є найбільш експериментально дослідженими та широко адаптованими блоковими шифрами. Однак, з появою нових алгоритмів малоресурсної криптографії, що є об'єктивним процесом, з'являються і нові методики та різновиди атак.

В цілому, бажаний малоресурсний алгоритм повинен забезпечувати баланс між вартістю, продуктивністю та безпекою. При цьому слід мати на увазі, що оптимізувати всі три цілі одночасно, дуже важко.

В індустрії IoT не існує AES-подібного стандарту для малоресурсних алгоритмів. З цієї причини найближчим часом можна очікувати інтенсифікацію розробок нових алгоритмів шифрування для нових IoT. Безумовно, безпечна мережева взаємодія має велике значення в сфері IoT, однак, крім питані ІБ, ефективне застосування IoT, є не менш важливим аспектом. Тому при розробці малоресурсних криптографічних алгоритмів, параметри їх енергоспоживання будуть вкрай актуальні.

Слід зазначити, що S-box PRESENT реалізує дуже компактну реалізацію, що споживає лише 21 GE для одного 4-бітового S-box. Крим того, з точки зору вимог до обсягів використовуваної пам'яті, цей алгоритм (в порівнянні з іншими) найкращім чином підходить для вирішення питань забезпечення малоресурсного криптографічного захисту даних при здійсненні мережевої взаємодії IoT (принаймні у найближчій перспективі).

Список літератури

[1] Xia, F., Yang, L. T., Wang, L., & Vinel, A. (2012). Internet of Things. International Journal of Communication Systems, 25(9), 1101–1102. https://doi.org/10.1002/dac.2417

[2] Jia, X., Feng, Q., Fan, T., & Lei, Q. (2012). RFID technology and its applications in Internet of Things (IoT). 2012 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet). https://doi.org/10.1109/cecnet.2012.6201508

[3] Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. Future Generation Computer Systems, 29(7), 1645–1660. https://doi.org/10.1016/j.future.2013.01.010

[4] Suo, H., Wan, J., Zou, C. and Liu, J. (2012) Security in the Internet of Things: A Review. IEEE International Conference on Computer Science and Electronics Engineering, Hangzhou, 23-25 March 2012, 648-651. https://doi.org/10.1109/ICCSEE.2012.373

[5] McKay, K. A., Bassham, L., Turan, M. S., & Mouha, N. (2017). Report on lightweight cryptography. https://doi.org/10.6028/nist.ir.8114

[6] ISO/IEC 29192-2:2012. (2012). Information technology – Security techniques – Lightweight cryptography – Part 2: Block ciphers. Retrieved from https://www.iso.org/obp/ui#iso:std:iso-iec:29192:-2:ed-2:v1:en.

[7] Banik, S., Bogdanov, A., Isobe, T., Shibutani, K., Hiwatari, H., Akishita, T., & Regazzoni, F. (2015). Midori: A Block Cipher for Low Energy. Advances in Cryptology – ASIACRYPT 2015, 411–436. https://doi.org/10.1007/978-3-662-48800-3_17

[8] Eisenbarth, T., Gong, Z., Güneysu, T., Heyse, S., Indesteege, S., Kerckhof, S., Koeune, F., Nad, T., Plos, T., Regazzoni, F., Standaert, F.-X., & van Oldeneel tot Oldenzeel, L. (2012). Compact Implementation and Performance Evaluation of Block Ciphers in ATtiny Devices. Progress in Cryptology - AFRICACRYPT 2012, 172–187. https://doi.org/10.1007/978-3-642-31410-0_11

[9] Chandra, S., Paira, S., Alam, S. S., & Sanyal, G. (2014). A comparative survey of Symmetric and Asymmetric Key Cryptography. 2014 International Conference on Electronics, Communication and Computational Engineering (ICECCE). https://doi.org/10.1109/icecce.2014.7086640

[10] Diehl, W., Farahmand, F., Yalla, P., Kaps, J.-P., & Gaj, K. (2017). Comparison of hardware and software implementations of selected lightweight block ciphers. 2017 27th International Conference on Field Programmable Logic and Applications (FPL). https://doi.org/10.23919/fpl.2017.8056808

[11] Bogdanov, A., Knudsen, L. R., Leander, G., Paar, C., Poschmann, A., Robshaw, M. J. B., Seurin, Y., & Vikkelsoe, C. (2007). PRESENT: An Ultra-Lightweight Block Cipher. Cryptographic Hardware and Embedded Systems - CHES 2007, 450–466. https://doi.org/10.1007/978-3-540-74735-2_31

[12] Shirai, T., Shibutani, K., Akishita, T., Moriai, S., & Iwata, T. (2007). The 128-bit blockcipher CLEFIA. In International workshop on fast software encryption, 181-195. Springer, Berlin, Heidelberg.

[13] Federal Information Processing Standards Publication 197 Announcing the ADVANCED ENCRYPTION STANDARD (AES). (2001). Retrieved from https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf

[14] Moradi, A., Poschmann, A., Ling, S., Paar, C., & Wang, H. (2011). Pushing the Limits: A Very Compact and a Threshold Implementation of AES. Advances in Cryptology – EUROCRYPT 2011, 69–88. https://doi.org/10.1007/978-3-642-20465-4_6

[15] Juels, A., & Weis, S. A. (2005). Authenticating Pervasive Devices with Human Protocols. Advances in Cryptology – CRYPTO 2005, 293–308. https://doi.org/10.1007/11535218_18

[16] Grossschadl, J., Tillich, S., Rechberger, C., Hofmann, M., & Medwed, M. (2007). Energy Evaluation of Software Implementations of Block Ciphers under Memory Constraints. 2007 Design, Automation & Test in Europe Conference & Exhibition. https://doi.org/10.1109/date.2007.364443

[17] Akishita, T., & Hiwatari, H. (2012). Very Compact Hardware Implementations of the Blockcipher CLEFIA. Selected Areas in Cryptography, 278–292. https://doi.org/10.1007/978-3-642-28496-0_17

[18] Hatzivasilis, G., Fysarakis, K., Papaefstathiou, I., & Manifavas, C. (2017). A review of lightweight block ciphers. Journal of Cryptographic Engineering, 8(2), 141–184. https://doi.org/10.1007/s13389-017-0160-y

[19] Dinu, D., Biryukov, A., Großschädl, J., Khovratovich, D., Le Corre, Y., & Perrin, L. (2015). FELICS - Fair Evaluation of Lightweight Cryptographic Systems. Retrieved from https://csrc.nist.gov/csrc/media/events/lightweight-cryptography-workshop-2015/documents/papers/session7-dinu-paper.pdf

[20] Meiser, G., Eisenbarth, T., Lemke-Rust, K., & Paar, C. (2008). Efficient implementation of eSTREAM ciphers on 8-bit AVR microcontrollers. 2008 International Symposium on Industrial Embedded Systems. https://doi.org/10.1109/sies.2008.4577681

Received: on June 2022. Accepted: on July 2022.

Authors:
Eugene Demenko, CSD Student, V. N. Karazin Kharkiv National University, Kharkov, Ukraine.
E-mail: xa11868404@student.karazin.ua
Oleksii Nariezhnii, Ph.D., Associate Professor, V. N. Karazin Kharkiv National University, Kharkov, Ukraine.
ORCID ID https://orcid.org/0000-0003-4321-0510
E-mail: o.nariezhnii@karazin.ua

Research of application of low-resource cryptography algorithms in decentralized environments.

Abstract. The purpose of this material is to analysis of the application of low-resource cryptography algorithms for Internet of Things (IoT) systems and the possibility of their implementation in decentralized systems. Over the past few years, the Internet of Things has become one of the most important technologies of the century. Modern IT developments has reached a high level of technological development, which allows you to customize the interaction between IoT devices and provide connection between people. With the appearance of 5G technologies, the IoT has become the center of development, for almost to all modern industries. Devices in this architecture are significantly smaller and have low power consumption. Conventional encryption algorithms tend to be computationally expensive due to their complexity and require many processing rounds. Low-resource cryptography is a compromise between implementation cost, speed, security, performance, and power consumption on IoT devices. The motivation for lightweight cryptography is to use less memory, less computing resources, and less power consumption to provide a security solution that can run on resource-constrained devices. Block ciphers have a fixed length (of bits) and special transformation stages, which are determined by a symmetric key. Block ciphers are quite versatile, which is very useful from an IoT perspective. Another advantage is that block ciphers has nearly proportional encryption and decryption methods. Therefore, it can be implemented with fewer resources.

Keywords: low-resource cryptography; Internet of things; block ciphers; IoT.