Введення параметра довіри до джерел даних розширює можливості моделі в умовах невизначеності та потенційного інформаційного впливу, що робить її придатною для застосування в сучасних кіберфізичних системах.

1. Roman R., Lopez J. Security in the Internet of Things: Current status and future challenges // Computer Networks. 2021.
2. Information security, cybersecurity and privacy protection — Guidance on managing information security risks : ISO/IEC 27005:2022. Geneva, 2022.
3. Zadeh L. Fuzzy sets // Information and Control. 1965. Vol. 8 / 3. P. 338–353.

Модифікація шифру Present

УДК:004.056.55

Володимир Лужецький[1], Тетяна Кирилащук[2]

Вінницький національний технічний університет,
[1]lva.kzi2002@gmail.com, [2]kgt0998@gmail.com

Алгоритм PRESENT є легковаговим блоковим шифром [1, 2], що базується на SP-мережі та складається з операцій підстановки та перестановки бітів. Операції підстановки реалізуються з використанням 16-ти однакових S-блоків, що складаються з логічних елементів, а перестановки бітів реалізуються фізичним розташуванням зв'язків. Крім того, для розгортання ключа використовується ще один S-блок. Оскільки S-блоки є необоротними, то для розшифрування використовуються інші S-блоки. Разом із перевагами алгоритму PRESENT існують і певні недоліки, пов'язані зі структурою S-блоків та кількістю раундів шифрування. У стандартній реалізації в кожному раунді використовуються однакові S-блоки, що значно спрощує апаратну реалізацію алгоритму, однак створює повторювану криптографічну структуру. Така регулярність може негативно впливати на стійкість алгоритму до окремих видів криптоаналізу, зокрема лінійного та диференціального. Ще одним недоліком є необхідність використання окремих інверсних S-блоків для процесу розшифрування. Для досягнення достатнього рівня нелінійності та дифузії даних використовується 31 раунд перетворень щоб компенсувати використання однакових S-блоків у всіх раундах.

З метою підвищення ефективності алгоритму шифрування у доповіді пропонується використовувати різні S-блоки, побудовані на основі латинських квадратів та перестановки змінних для S-блоків. Використання різних нелінійних перетворень у різних раундах дозволяє ускладнити криптоаналітичні залежності між ними та потенційно забезпечити необхідну криптографічну стійкість при меншій кількості раундів. Такий підхід може сприяти пришвидшенню процесу шифрування, зменшенню затримок обробки даних та підвищенню продуктивності алгоритму в ресурсно-обмежених системах, зокрема в IoT-пристроях та вбудованих інформаційних системах.

Пропонується для реалізації S-блоків використовувати латинські квадрати 4-го порядку. Оскільки, існує 576 таких латинських квадратів [3], то є можливість вибрати з них 16, які забезпечать побудову оборотних S-блоків. Як

наслідок, для розшифрування будуть використовуватись ті ж самі S-блоки, що і для зашифрування. Схему S-блоку на основі двох однакових латинських квадратів наведено на рис.1.

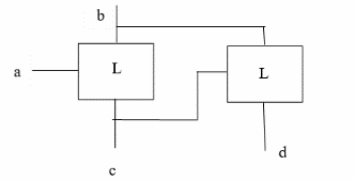

Рис.1 Схема S-блоку на основі латинських квадратів

Результат підстановки визначається як:

$$S'(x) = L(L(a,b),c),$$

де L - латинський квадрат, $a \parallel b$ - 4 вхідні біти, $c \parallel d$ – результат підстановки.

Проведені дослідження дозволили обрати 16 латинських квадратів 4x4, які є оборотними та забезпечують максимальну нелінійність S-блоків.

Додатково застосовується перестановка змінних:

$$\pi_1 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 1 & 2 & 3 \end{pmatrix};\ \pi_2 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 1 & 4 & 2 \end{pmatrix};\ \pi_3 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 1 & 3 \end{pmatrix};\ \pi_4 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 1 & 2 \end{pmatrix};$$

$$\pi_5 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 3 & 1 & 2 \end{pmatrix};\ \pi_6 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \end{pmatrix};\ \pi_7 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 2 & 1 \end{pmatrix};\ \pi_8 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 3 & 2 & 1 \end{pmatrix}.$$

де π - перестановка індексів. Це означає, що перед застосуванням S-блоку змінюється порядок бітів, що дозволяє підвищити нелінійність та ускладнити алгебраїчну структуру. Ще одна модифікація шифру PRESENT полягає в тому, що накладання ключа після перестановки виконується не на основі операції XOR, а як операція, що описується латинським квадратом. Тобто, замість суматорів за модулем 2 використовуються логічні схеми, що реалізують латинські квадрати.

Таким чином, модифікація полягає не в зміні всієї структури PRESENT, а саме в заміні набору з 16 однакових S-блоків на набір з 16 різних S-блоків, сформованих на основі різних оборотних латинських квадратів. Такий підхід дозволяє зберегти легковагову структуру шифру, але водночас підвищити його криптографічну стійкість. Апаратна складність реалізації S-блоку на основі оборотних латинських квадратів складає 64 умовні одиниці, тоді як складність реалізації S-блоку шифру PRESENT дорівнює 95 умовних одиниць.

Отже, встановлено, що використання латинських квадратів у поєднані з перестановками змінних дозволяє досягти балансу між підвищенням криптостійкості та збереженням прийнятного рівня апаратної складності, що є важливим для ресурсно-обмежених середовищ, таких як IoT-пристрої та вбудовані системи.

1. Bogdanov A., Knudsen L., Leander G. та ін. PRESENT: An Ultra-Lightweight Block Cipher // Cryptographic Hardware and Embedded Systems – CHES 2007. Berlin : Springer, 2007. С. 450–466.
2. Wang Y., Ha Y. Compact FPGA implementation of PRESENT cipher with optimized S-box // IEEE Transactions on Very Large Scale Integration Systems. – 2011. – Vol. 19, №10. – P. 1864–1873.
3. A. Donald Keedwell and József Dénes. Latin Squares and their Applications. Elsevier, 2015. 439 с. URL: https://doi.org/10.1016/c2014-0-03412-0.

Розробка архітектури програмного застосунку для децентралізованої торгівлі електроенергією з використанням смарт-контрактів в блокчейн

УДК 004.4

Ганна Неласа[1], Вахтанг Чіхладзе[2],
Андрій Ублінських[3], Олег Неласий[4]

Інститут проблем моделювання в енергетиці ім. Г.Є. Пухова,
[1]annanelasa@gmail.com,
Національний технічний університет України «КПІ імені Ігоря Сікорського», [2]the.vaho1337@gmail.com,
Cytric, [3]andreo.ublin24@gmail.com,
Національний університет «Запорізька політехніка», [4]oleg.nelasy@gmail.com

В роботі представлено архітектуру програмного застосунку для децентралізованої торгівлі електроенергією з використанням смарт-контрактів в блокчейн [1]. На сьогодні смартконтракти набули широкого застосування та стали невід'ємною складовою блокчейн і фінансової індустрії.

Основна ідея полягає в токенізації 15-хвилинних порцій електроенергії, розділенні вартості електроенергії та прибутку на різні токени, та введенні фінансового посередника між виробниками та споживачами, який балансує момент розрахунків між ними, тобто сплачує виробнику за поставлений обсяг електроенергією відразу, а оплату від споживачів отримує пізніше при використанні відповідних порцій. Фінансовий посередник може отримувати дохід за свої послуги, оскільки протокол Pendle [2] довів створення ринку дохідності при розділенні базового активу та майбутнього доходу.

Відповідно в схемі використовуються токени:

- PT (PrincipalToken) - токен, який надає право користування електроенергією в заданий проміжок часу, «тіло» активу, тобто сума, що буде повернута після настання дати погашення. PT не приносить дохід, а лише гарантує повернення активу.
- YT (YieldToken) - токен, який містить у собі майбутній прибуток виробника-продавця.
- SY (Standardized Yield Token) виступає як «обгортка», що представляє повну вартість активу разом із майбутньою дохідністю.