УДК 004.056.55 **DOI:10.30837/rt.2021.4.207.06**

В.І. РУЖЕНЦЕВ, д-р техн. наук, О.І. ФЕДЮШИН, канд. техн. наук, С.А. КОХАН

АНАЛІЗ СТІЙКОСТІ ARX СХЕМ ШИФРУВАННЯ ДО ІНТЕГРАЛЬНОЇ АТАКИ ТА АТАКИ НЕЗДІЙСНЕНИХ ДИФЕРЕНЦІАЛІВ

Вступ

Серед малоресурсних криптоалгоритмів перспективними вважаються Addition- Rotation-XOR (ARX) схеми, тобто алгоритми, які використовують лише три види операцій: Addition – модульне додавання, Rotation – циклічний зсув та XOR додавання. У попередніх роботах [1, 2] було обрано найбільш відомі алгоритми цього класу, розроблено зменшені моделі (розмір блоку 16 бітів) цих алгоритмів та проаналізовано стійкість до атак диференційного, лінійного криптоаналізу, а також ступінь нелінійності цих ARX перетворень. Однак на основі аналізу літератури з описом атак на ARX алгоритми зроблено висновок про високу ефективність інтегральної атаки та атаки нездійсненних диференціалів проти цих алгоритмів. Отже метою цієї роботи є аналіз стійкості 16-бітних ARX перетворень до більш спеціалізованих та ефективних для цього виду перетворень інтегральної атаки та атаки нездійсненних диференціалів.

Для досягнення мети потрібно обрати та реалізувати методи аналізу стійкості до інтегральної атаки та атаки нездійсненних диференціалів і застосувати їх до обраних зменшених ARX алгоритмів.

Зменшені ARX моделі та початковий аналіз стійкості

В цій частині наведено опис використаних в роботі 16-бітних ARX моделей, більшість з яких вже були представлені та розглядались в роботах [1, 2], а також основні результати аналізу цих схем з наведених робіт.

Перша ARX-схема ChaCha – це quarter-round потокового алгоритму Chacha 2 [3] зі зменшеним розміром підблоків. 16-бітовий блок схеми складається з чотирьох 4-бітових підблоків (рис. 1).

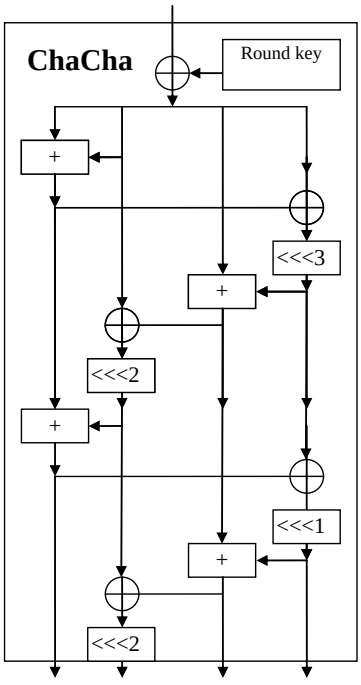


Рис. 1. ChaCha схема

Друга ARX-схема Speckey – це спрощена схема алгоритму Speck [4]. Спрощення полягає у відсутності двох операцій циклічного зсуву, які в оригінальному варіанті передували операціям модульного додавання. 16-бітовий блок схеми Speckey складається з двох 8-бітових підблоків (рис. 2).

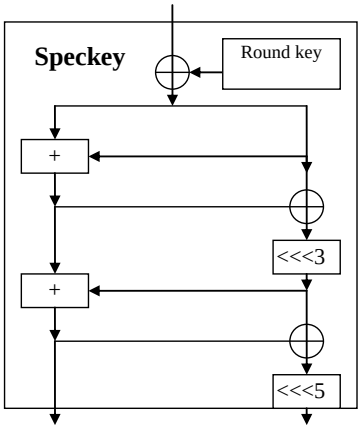


Рис. 2. Speckey схема

Зменшена модель циклу алгоритму Simon [4] представлена на рис. 3.

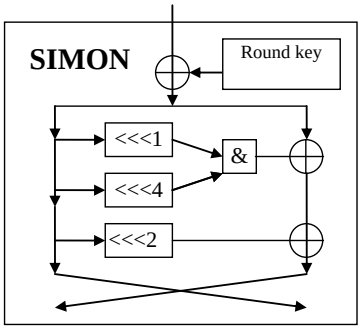
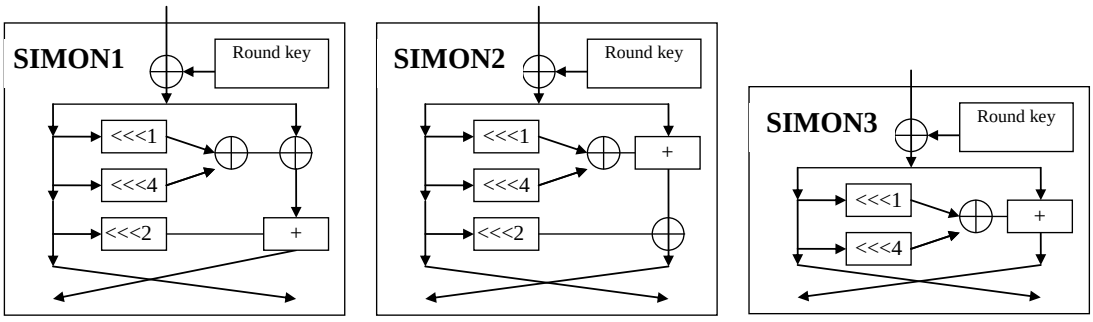


Рис. 3. Схема циклової функції шифру Simon

В роботі також розглядаються модифікації Simon1, Simon2, Simon3, які представлено на рис. 4 та які замість операції AND та деяких операцій XOR використовують модульне додавання.



а *б* *в*

Рис. 4. Модифікації циклової функції шифру Simon

Наступна ARX-схема – це зменшена схема алгоритму Chaskey [5]. 16-бітовий блок схеми Chaskey складається з чотирьох 4-бітових підблоків (рис. 5).

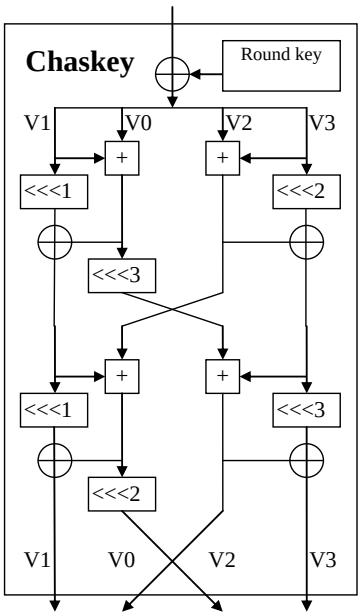


Рис. 5. Схема циклової функції схеми Chaskey

Ще одна ARX-схема – це зменшена схема S-блоку з алгоритма Sparkle, який в [6] названий Alzette. 16-бітовий блок зменшеної схеми Alzette складається з двох 8-бітових підблоків (рис. 6).

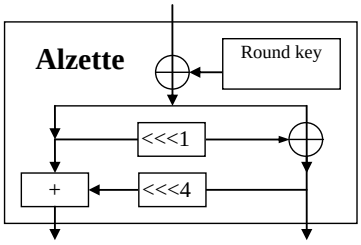


Рис. 6. Циклова функція схеми Alzette

У табл. 1 представлено кількість і формат операцій для розглянутих вище схем.

Таблиця 1

Кількість та формат операцій в одному циклі шифруючих перетворень (без додавання ключа)

Шифруючі схеми	Модульне додавання (Addition)	Циклічний зсув (Rotation)	XOR
Speckey	2*8 bit	2*8 bit	2*8 bit
ChaCha	4*4 bit	4*4 bit	4*4 bit
Simon	1*8 bit	3*8 bit	1*8 bit +1 AND
Simon1	1*8 bit	3*8 bit	2*8 bit
Simon2	1*8 bit	3*8 bit	2*8 bit
Simon3	1*8 bit	2*8 bit	1*8 bit
Chaskey	4*4 bit	4*4 bit	4*4 bit
Alzette	1*8 bit	2*8 bit	1*8 bit

В роботах [1, 2] схеми використовували одну операцію XOR додавання з ключем на початку перетворень і, потім, деяку кількість циклових перетворень. Аналіз показників криптографічної стійкості зменшених моделей (16 бітний блок та ключ) в [1, 2] продемонстрував, що майже для усіх з них можливо отримати показники випадкової підстановки при використанні певної кількості циклів. Виключенням стала схема алгоритму Simon – не дозволяє отримати ці показники навіть при великій кількості циклів. Схеми, що оперують

4-бітними та 8-бітними підблоками показували схожі результати. Так, схеми з 4-бітними блоками потребували 60 – 72 ARX операцій, а схеми з 8-бітними блоками потребували 32-36 ARX операцій для досягнення диференційних та лінійних показників випадкових підстановок. Підсумкові таблиці з мінімальною кількістю циклів та кількістю елементарних операцій для забезпечення показників випадкової підстановки з робіт [1, 2] представлено у табл. 2, 3.

Таблиця 2

Кількість 8-бітних операцій для забезпечення стійкості проти диференційних, лінійних та алгебраїчних атак

Шифруючі схеми	Мінімальна кількість циклів	Кількість 8-бітних операцій			
		Addition	Rotation	Xor	Всього
Speckey	6	12	12	12	36
Simon2	6	6	18	12	36
Simon3	8	8	16	8	32
Alzette	9	9	18	9	36

Таблиця 3

Кількість 4-бітних операцій для забезпечення стійкості проти диференційних, лінійних та алгебраїчних атак

Шифруючі схеми	Мінімальна кількість циклів	Кількість 4-бітних операцій			
		Addition	Rotation	Xor	Всього
ChaCha	6	24	24	24	72
Chaskey	5	20	20	20	60

Аналіз стійкості до інтегральної атаки

Інтегральний криптоаналіз є одним з найбільш ефективних видів нападу на найпоширеніший у світі шифр Rijndael і його варіант – AES зі зменшеною кількістю циклів. Аналіз робіт, присвячених криптоаналізу ARX-алгоритмів, показав, що і для цього класу шифрів атака є однієї з найбільш успішних.

Інтегральною атака названа тому, що в атаці розглядається проходження через перетворення шифру суми станів. Тут під різними станами розуміються деякі проміжні значення блоків даних у процесі їхнього зашифрування. Подібно тому, як у диференційній атаці виконується "транспортування" різності через перетворення шифру, у даній атаці через цикли шифру проводиться значення суми деякої кількості станів.

Якщо є можливість з високою ймовірністю визначити значення деяких бітів суми станів після r циклів шифрування, то це означає, що може бути організована інтегральна атака на (r+1)-цикловий шифр.

Для проведення аналізу стійкості ARX схем до інтегральної атаки був використаний метод [7]. У табл. 4 наведено алгоритм для тестування стійкості.

Таблиця 4

Алгоритм пошуку R-циклових інтегралів

	Вхідні дані: 16 бітне R-циклове шифруюче перетворення E. Num_key – кількість випадково обраних ключів.
1	Перебір всіх 16 варіантів позиції пасивного біта у вхідному блоці
1.1	Перебір Num_key варіантів ключа k.
1.1.1	Формування 2^{15} 16 бітних вхідних станів x
1.1.2	Зашифрування кожного з 2^{15} 16 бітних вхідних станів: $E_k(x)$
1.2	Перевірка наявності збалансованих бітів (XOR сума на всій множині криптограм дорівнює 0) в криптограмах для поточного варіанта позиції пасивного біта на вході
	Вихідні дані: Знайдені збалансовані біти в криптограмах – R-циклові інтеграли.

Цей алгоритм було використано для пошуку інтегралів для всіх 16-бітних ARX схем.

Кількість ключів Num_key в роботі [7] запропоновано 10, але в експериментах використовували 32.

Одним з найбільш цікавих висновків аналізу стало те, що жодна з розглянутих зменшених моделей (16-бітний блок та ключ) з лише однією операцією XOR додавання з ключем (саме такий варіант перетворень розглядався в роботах [1, 2]) не забезпечує відсутності інтегралів при будь-якій кількості циклів.

Для всіх варіантів схеми Simon (чотири схеми, що представлені на рис. 3, 4) потрібно використовувати щонайменше три операції XOR додавання з ключем, щоб, використовуючи вказану у табл. 5 кількість циклів, забезпечити відсутність інтегралів.

Таблиця 5

Кількість 8-бітних операцій для забезпечення відсутності інтегралів

Схема	Мінімальна кількість XOR додавань з ключем	Мінімальна кількість циклів	Кількість ARX операцій			
			Addition	Rotation	Xor	Всього
Speckey	2	5	10	10	10	30
simon	3	10	10	30	20	60
Simon1	3	7	7	21	14	42
Simon2	3	7	7	21	14	42
Simon3	3	7	7	21	14	42
Alzette	2	9	9	18	9	36

Таблиця 6

Кількість 4-бітних операцій для забезпечення відсутності інтегралів

Схема	Мінімальна кількість XOR додавань з ключем	Мінімальна кількість циклів	Кількість ARX операцій			
			Addition	Rotation	Xor	Всього
ChaCha	2	10	40	40	40	120
Chaskey	2	6	24	24	24	72

Для інших схем (Speckey, Alzette, ChaCha, Chaskey) потрібно використовувати щонайменше дві операції XOR додавання з ключем для можливості забезпечення відсутності інтегралів при належній кількості циклів (див. табл. 5, 6).

Важливо також те, що подальше збільшення кількості операцій XOR додавання з ключем не змінює мінімальної кількості циклів для забезпечення стійкості для усіх схем, що розглядаються.

З табл. 5 бачимо, що кращій результат стійкості до інтегральної атаки серед схем, що оперують 8-бітовими блоками, демонструє схема Speckey. Ця схема потребує 30 операцій для забезпечення відсутності інтегралів. Найгірший результат демонструє Simon – 60 операцій. Серед схем, що оперують 4-бітовими блоками (див. табл. 6), краща – Chaskey – 72 операції, гірша ChaCha – 120 операцій. На відміну від результатів, що були представлені в роботах [1, 2], відмінності між кращими та гіршими схемами більш суттєві – майже в два рази.

Якщо порівнювати всі розглянуті схеми і рахувати, що одна 8-бітова операція еквівалентна двом 4-бітовим, то найуспішнішою схемою з точки зору стійкості до інтегральної атаки можна вважати Speckey.

Аналіз стійкості до атаки нездійснених диференціалів

Криптоаналітичний метод називається атакою нездійснених диференціалів (НД), оскільки використовує диференціали спеціального виду – ті, котрі не можуть здійснитися, тобто мають нульову ймовірність. Атака нездійснених диференціалів на r-цикловий шифр стає можливою, коли є $(r\text{-}1)$-цикловий НД.

Атака є однією з найбільш ефективних для сучасних алгоритмів шифрування, у тому числі і для ARX алгоритмів.

Алгоритм, який було використано для пошуку НД, наведено у табл. 7.

Таблиця 7

Алгоритм пошуку НД

Вхідні дані: Шифруюче перетворення E. Пуста строка таблиці різності відповідного розміру.	
1	Перебір всіх варіантів вхідної різності d
1.1	У строку таблиці різності запусують всі «0»
1.2	Перебір Num_key варіантів ключа k
1.2.1	Перебір всіх варіантів вхідного значення x
1.2.1.1	Інкрементуємо ячійку з індексом $E_k(x)$ xor $E_k(x$ xor $d)$
1.3	Перевіряємо строку таблиці різності на наявність «0». Кожен такий «0» відповідає НД
	Вихідні дані: Знайдені НД.

Алгоритм з табл. 5 було використано для пошуку інтегралів для всіх 16-бітних ARX схем. Експериментально визначено кількість ключів Num_key = 32, яка потрібна для того, щоб при певній кількості циклів не існувало НД. Відомо, що при малій кількості ключів завжди існують НД для будь-якої кількості циклів.

Як і в попередньому розділі важливим для забезпечення стійкості виявилась кількість операцій XOR додавання з ключем. Знову лише одна операція XOR додавання з ключем (саме такий варіант перетворень розглядався в роботах [1, 2]) не забезпечує стійкості при будь-якій кількості циклів для всіх схем.

Щонайменше дві операції XOR додавання з ключем достатньо для лише для двох схем: Chaskey та Speckey. Для всіх інших схем потрібно не менше трьох операцій XOR додавання з ключем для забезпечення відсутності нездійснених диференціалів (див. табл. 8, 9).

Таблиця 8

Кількість 8-бітних операцій для забезпечення відсутності нездійснених диференціалів

Схема	Мінімальна кількість XOR додавань з ключем	Мінімальна кількість циклів	Кількість ARX операцій			
			Addition	Rotation	Xor	Всього
Speckey	2	5	10	10	10	30
simon	3	9	9	27	18	54
Simon1	3	7	7	21	14	42
Simon2	3	6	6	18	12	36
Simon3	3	6	6	18	12	36
Alzette	3	9	9	18	9	36

Таблиця 9

Кількість 4-бітних операцій для забезпечення відсутності нездійснених диференціалів

Схема	Мінімальна кількість XOR додавань з ключем	Мінімальна кількість циклів	Кількість ARX операцій			
			Addition	Rotation	Xor	Total
ChaCha	3	9	36	36	36	108
Chaskey	2	5	20	20	20	60

Порівнюючи результати аналізу наявності інтегралів (табл. 5, 6) та нездійснених диференціалів (табл. 8, 9) для зменшених 16-бітних ARX схем, можна побачити, що для досягнення відсутності нездійснених диференціалів потрібно або стільки, або на 1 менше циклів, ніж для досягнення відсутності інтегралів. Але для деяких схем (Alzette, ChaCha) при цьому потрібно більше XOR додавань з ключем.

Підсумкові дані щодо кількості циклів, кількості XOR додавань з ключем та кількості елементарних ARX перетворень, які потрібні для того, щоб перетворення не відрізнялись від випадкової підстановки, наведено в табл. 10, 11 (дані отримані на основі табл. 2, 3, 5, 6, 8, 9).

Таблиця 10

Кількість 8-бітних операцій для забезпечення показників випадкової підстановки

Схема	Мінімальна кількість XOR додавань з ключем	Мінімальна кількість циклів	Кількість ARX операцій			
			Addition	Rotation	Xor	Всього
Speckey	2	6	12	12	12	36
Simon	3	10	10	30	20	60
Simon1	3	7	7	21	14	42
Simon2	3	7	7	21	14	42
Simon3	3	8	8	24	16	48
Alzette	3	9	9	18	9	36

Таблиця 11

Кількість 4-бітних операцій для забезпечення показників випадкової підстановки

Схема	Мінімальна кількість XOR додавань з ключем	Мінімальна кількість циклів	Кількість ARX операцій			
			Addition	Rotation	Xor	Total
ChaCha	3	10	40	40	40	120
Chaskey	2	6	24	24	24	72

На основі табл. 10 можна виділити найкращі 8-бітові схеми: Speckey та Alzette. При цьому Speckey потребує меншої кількості XOR додавань з ключем. Найгірші показники продемонструвала схема Simon, яка потребує майже вдвічі більшої кількості операцій (60 проти 36). Серед 4-бітних схем (табл. 11) кращою є Chaskey, яка потребує 72 4-бітних операції. Якщо рахувати, що одна 8 бітова операція еквівалентна двом 4-бітовим, то ця схема еквівалентна кращій 8-бітній схемі Speckey.

Висновки

1. Проаналізовано показники криптографічної стійкості зменшених моделей (16 бітний блок та ключ) відомих та поширених сьогодні ARX алгоритмів шифрування: Chacha, Speckey, Simon, Chaskey, Sparkle та їх модифікації до найбільш ефективних для цього класу алгоритмів атак: інтегральна атака та атака нездійснених диференціалів. Продемонстровано, що для більшості з них можливо отримати показники випадкової підстановки при використанні певної кількості циклів та певної кількості XOR додавань з ключем.

2. Одним з цікавих висновків аналізу стало те, що жодна з розглянутих зменшених моделей (16 бітний блок та ключ) з лише однією операцію XOR додавання з ключем (саме такий варіант перетворень розглядався в роботах [1, 2]) не забезпечує відсутності інтегралів та нездійснених диференціалів при будь-якій кількості циклів, що, на наш погляд, свідчить про більшу ефективність атак, що розглядаються в цій роботі, а також про важливість операцій додавання з ключем в загальній криптографічній стійкості ARX алгоритмів. Так, для всіх варіантів схеми Simon (чотири схеми, що представлені на рис. 3, 4), Alzette і ChaCha потрібно використовувати щонайменше три операції XOR додавання з ключем, щоб, використовуючи вказану у табл. 5, 6 кількість циклів, забезпечити відсутність інтегралів та нездійснених диференціалів. Для інших двох схем – Speckey і Chaskey – достатньо двох таких операцій.

3. Підсумкові дані аналізу наведено в табл. 10, 11 (дані отримані на основі табл. 2, 3, 5, 6, 8, 9). Ці таблиці відображають кількість циклів, кількість XOR додавань з ключем та кількість елементарних ARX перетворень, які потрібні для того, щоб ARX перетворення не відрізнялись від випадкової підстановки. На основі табл. 10, 11 можна виділити найкращі та найгірші ARX перетворення. Кращі 8-бітові схеми: Speckey та Alzette (потребують по 36 ARX операцій). При цьому Speckey потребує меншої кількості XOR додавань з ключем, тому має перевагу. Найгірші показники продемонструвала схема Simon, яка потребує значно більшої кількості операцій – 60. Серед 4-бітних схем (табл. 11) кращою є Chaskey, яка потребує 72 4-бітних операції. Якщо рахувати, що одна 8-бітова операція еквівалентна двом 4-бітовим, то ця схема еквівалентна кращій 8-бітній схемі Speckey.

Список літератури:

1. Руженцев В.І. Порівняльний аналіз ARX схем шифрування // Радіотехніка. 2020. Вип. 202. С. 79 – 86.

2. Victor Ruzhentsev. Comparative analysis of ARX transformations // Book of Abstracts 20th Central European Conference on Cryptology, June 24 – 26, 2020, Zagreb, Croatia. P. 42-43.

3. Daniel J. Bernstein. Chacha, a variant of Salsa20. SASC 2008 –the State of the Art in Stream Ciphers. See also https://cr.yp.to/chacha.html, 2008.

4. R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers. The SIMON and SPECK families of lightweight block ciphers // Cryptology ePrint Archive, Report 2013/404, 2013. http://eprint.iacr.org/2013/404.

5. Nicky Mouha, Bart Mennink, Anthony Van Herrewege, Dai Watanabe, Bart Preneel, and Ingrid Verbauwhede. Chaskey: An effcient MAC algorithm for 32-bit microcontrollers // Antoine Joux and Amr M. Youssef, editors, SAC 2014: 21st Annual International Workshop on Selected Areas in Cryptography, volume 8781 of Lecture Notes in Computer Science, pages 306–323. Springer, Heidelberg, August 2014. Doi:10.1007/978-3-319-13051-4_19

6. Lightweight cryptography project of the American National Institute of Standards and Technology. https://csrc.nist.gov/projects/lightweight-cryptography.

7. J. Ren, S. Chen. Cryptanalysis of Reduced-Round SPECK. IEEE Xplore. Vol. 7, 2019. P. 63045-63056. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8715440.

Надійшла до редколегії 05.11.2021

Відомості про авторів:

Руженцев Віктор Ігорович – д-р техн. наук, доцент, Харківський національний університет радіоелектроніки, професор кафедри безпеки інформаційних технологій; Україна; e-mail: viktor.ruzhentsev@nure.ua, ORCID: http://orcid.org/0000-0002-1007-6530

Федюшин Олександр Іванович – канд. техн. наук, доцент, Харківський національний університет радіоелектроніки, доцент кафедри безпеки інформаційних технологій; Україна; e-mail: oleksandr.fediushyn@nure.ua, ORCID: http://orcid.org/0000-0002-3600-405X

Кохан Сергій Анатолійович – Харківський національний університет радіоелектроніки, аспірант кафедри безпеки інформаційних технологій; Україна; e-mail: serhii.kokhan@nure.ua