УДК 621.3.06 **DOI:10.30837/rt.2020.1.200.05**

М.Ю. РОДІНКО, Р.В. ОЛІЙНИКОВ, д-р техн. наук

ДОСЛІДЖЕННЯ ПРОДУКТИВНОСТІ МАЛОРЕСУРСНОГО БЛОКОВОГО ШИФРУ «КИПАРИС» НА РІЗНИХ ПЛАТФОРМАХ

Вступ

Особливе місце у сучасній криптографії займають малоресурсні шифри, призначені для використання у пристроях з обмеженою кількістю споживання енергії. Інтерес до малоресурсних примітивів проявив й Національний Інститут Стандартів та Технологій США, який організував конкурс на розробку малоресурсного алгоритму AEAD [1, 2].

Так само як і в традиційних, у малоресурсних блокових шифрах найбільш поширеними типами високорівневої конструкції є SPN-структура та мережа Фейстеля. В деяких шифрах застосовуються специфічні різновиди цих конструкцій такі, наприклад, як ARX-подібна SPN-структура у блоковому шифрі SPARX [3] або один з варіантів узагальненої мережі Фейстеля (англ. Generalized Feistel Network, GFN) у шифрах CLEFIA [4] та TWINE [5].

Серед блокових шифрів, заснованих на SPN-структурі, окремо можна виділити AES-подібні алгоритми, до яких можна віднести блокові шифри KLEIN [6] та Midori [7]. Причиною, з якої автори спираються на AES при розробці нових алгоритмів, є високий рівень вивченості шифру та його стійкість, що доведена математично та перевірена часом. До малоресурсних SPN-шифрів також відносяться PRESENT [8], PRINCE [9] та ін.

Багато малоресурсних блокових шифрів засновано і на мережі Фейстеля. В якості циклової функції мережі Фейстеля у малоресурсних алгоритмах все частіше застосовується не чергування S-блоків та лінійного перетворення, а так зване ARX-перетворення (англ. Addition, Rotation, XOR), що складається з операцій модульного додавання, циклічного зсуву та додавання за модулем 2. Операції, як правило, виконуються не над байтами, а над цілим напівблоком, розмір якого найчастіше складає 32 або 64 біти, що суттєво підвищує швидкодію перетворень при використанні шифру на процесорах з архітектурою аналогічної розрядності. До таких шифрів відносяться SPECK [10], TEA [11], XTEA [12], а також нещодавно розроблений блоковий шифр «Кипарис» [13]. На відміну від інших алгоритмів, «Кипарис» підтримує довжину блока та ключа 256 та 512 біт, завдяки чому залишиться стійким у постквантовий період.

Блоковий шифр «Кипарис» оперує блоками даних розміром l біт, із використанням ключа шифрування довжиною k біт, $l, k \in \{256, 512\}$, $l = k$. Операції циклової функції виконуються над s-бітними словами, $s \in \{32, 64\}$. Як вже було зазначено, «Кипарис» представляє собою мережу Фейстеля з ARX-перетворенням у якості циклової функції, що містить вісім додавань за модулем 2^s, вісім додавань за модулем 2 та вісім циклічних зсувів.

Метою роботи є дослідження продуктивності (а саме, такого показника як швидкість зашифрування в режимі простої заміни, у Мбіт/с) блокового шифру «Кипарис» та порівняння з продуктивністю інших відомих блокових шифрів на платформах Windows, Linux та Android.

1. Методика вимірювання швидкодії блокових шифрів

Для вимірювання швидкодії блокових шифрів використовувалась методика та програмний код, написаний мовою програмування C++, що застосовувались для дослідження продуктивності блокового шифру «Калина» [14]. З метою отримання точних та достовірних результатів методика передбачає багатократне (наприклад, восьмикратне) зашифрування блоку пам’яті фіксованого розміру (наприклад, 1 ГБ) у режимі простої заміни. Блок пам’яті складається з N блоків даних, де N залежить від розміру вхідного блока алгоритму шифру-

вання. Кожен з *N* блоків представляється у вигляді масиву 64-бітових беззнакових цілих чисел (розмірність масиву залежить від розміру блока, яким оперує шифр).

Для кожного з алгоритмів *N* задається константою, наприклад *number_of_blocks_in_memory_128* для шифру з 128-бітовим розміром блока. Для ініціалізації блоку пам'яті певного розміру використовується відповідна функція, наприклад *InitMemoryEncryptionBlock128()*.

Для отримання поточного значення числа тактів процесору призначена функція *DetermineTime()*. Для отримання числа тактів в операційній системі Linux використовувалася функція *gettimeofday()* з бібліотеки <sys/time.h>. Для операційної системи Windows була написана власна функція *Mygettimeofday()*.

На основі значень системного часу до початку та після закінчення виконання програмного блоку, що реалізує зашифрування, було обчислено швидкість зашифрування у Мбіт/с.

Вимірювання швидкодії блокових шифрів здійснювалося на наступних платформах:

– Процесор Intel Core i7-7500U з тактовою частотою 2,7-2,9 GHz, операційна система Windows 10 x32;

– Процесор Intel Core i7-7500U з тактовою частотою 2,7-2,9 GHz, операційна система Windows 10 x64;

– Процесор Intel Core i5-4670U з тактовою частотою 3,4 GHz, операційна система Linux (64-bits);

– Процесор Mediatek MT6582 з тактовою частотою 1,3 GHz, 4 ядра ARM Cortex-A7, операційна система Android 4.2.2 Jelly Bean (32-bits);

– Процесор Exynos 7880 з тактовою частотою 1,9 GHz, 8 ядер ARM Cortex-A53, операційна система Android 8.0.0 (64-bits).

У зв'язку із тим, що процесори, використовувані у мобільних пристроях, мають набагато нижчу продуктивність, для вимірювання швидкодії алгоритмів на ОС Android замість 8 ГБ пам'яті шифрувалося 8 МБ.

Окрім блокових шифрів «Кипарис-256» та «Кипарис-512» для отримання оцінок щодо швидкості зашифрування було обрано наступні блокові шифри:

- AES-256 [15];
- SPECK-64/128 [10];
- SPECK-128/128 [10];
- SPARX-128/128 [3];
- ДСТУ ГОСТ 28147: 2009 [16].

AES та ГОСТ-28147-89 було обрано як найбільш відомі та перевірені часом алгоритми, а SPECK – з міркувань того, що цей шифр, подібно до шифру «Кипарис», заснований на мережі Фейстеля з ARX-перетворенням у якості циклової функції. SPARX був обраний для порівняння як перший (разом із LAX) доказово стійкий малоресурсний блоковий шифр.

Обчислення швидкодії блокового шифру AES-256 здійснювалось для оптимізованої реалізації з використанням таблиць передобчислень, представленої в [14]. Реалізацію шифру ДСТУ ГОСТ 28147:2009 було обрано з того ж джерела [14].

Також були використані реалізації блокових шифрів SPECK-64/128 та SPARX-128/128 з бібліотеки FELICS [17], що містить оптимізовані реалізації найбільш відомих малоресурсних алгоритмів. Зазначимо, що не всі реалізації з цієї бібліотеки видаються достатньо оптимізованими з точки зору швидкодії (принаймні ті, що орієнтовані на застосування на процесорах загального призначення). Так, наприклад, внесення незначних змін у програмний код, що реалізує блоковий шифр SPECK-64/128 (заміна викликів функцій у процедурі зашифрування простою підстановкою коду, який вони містять), дозволило підвищити швидкодію у декілька

разів. Для SPECK-128/128 було обрано реалізацію, запропоновану авторами [18], перевагою якої є дуже компактний програмний код (функція зашифрування містить менше десяти рядків коду).

2. Результати вимірювання швидкодії блокових шифрів

Результати вимірювання швидкодії шифрів на різних платформах наведені в табл. 1 – 3.

Таблиця 1

Порівняння продуктивності малоресурсних блокових шифрів на платформі Windows 10 x32, процесор Intel Core i7-7500U

Блоковий шифр	Розмір блока, біт	Довжина ключа, біт	Швидкість зашифрування, Мбіт/с	Реалізація
«Кипарис-256»	256	256	3472,04	Власна
«Кипарис-512»	512	512	1555,54	Власна
AES-256	128	256	1441,85	[14]
SPECK	64	128	3059,16	[17]
SPECK	128	128	748,96	[18]
SPARX	128	128	661,83	[17]
ДСТУ ГОСТ 28147:2009	64	128	603,4	[14]

Шифри SPECK-128/128, SPARX-128/128 та ДСТУ ГОСТ 28147:2009 показали близький результат у межах 600-750 Мбіт/с. Далі йдуть блокові шифри AES-256 та «Кипарис-512» зі швидкістю порядку 1,5 Гбіт/с. Реалізація блокового шифру SPECK-64/128 [119] забезпечує швидкість шифрування порядку 3 Гбіт/с.

Значна різниця у швидкості зашифрування між реалізаціями шифрів SPECK-64/128 та SPECK-128/128 пояснюється тим, що SPECK-64/128 оперує 32-бітовим блоком, а SPECK-128/128 – 64-бітовим блоком, тому на 32-бітовій платформі SPECK-64/128 значно виграє у швидкодії. Теж саме стосується і шифрів «Кипарис-256» та «Кипарис-512».

Найкращий результат на 32-бітовій платформі Windows 10 показав блоковий шифр «Кипарис-512», його швидкодія склала майже 3,5 Гбіт/с.

Таблиця 2

Порівняння продуктивності малоресурсних блокових шифрів на платформі Windows 10 x64, процесор Intel Core i7-7500U

Блоковий шифр	Розмір блока, біт	Довжина ключа, біт	Швидкість зашифрування, Мбіт/с	Реалізація
«Кипарис-256»	256	256	3502,46	Власна
«Кипарис-512»	512	512	4942,77	Власна
AES-256	128	256	1653,79	[14]
SPECK	64	128	3038,03	[17]
SPECK	128	128	4786,81	[18]
SPARX	128	128	936,373	[17]
ДСТУ ГОСТ 28147:2009	64	128	526,583	[14]

На 64-бітовій платформі Windows 10 найкращий результат очікувано показали блокові шифри «Кипарис-512» (порядку 5 Гбіт/с) та SPECK-128/128 (порядку 4,8 Гбіт/с), які обробляють 64-бітові блоки даних. Крім того, перевагою блоковий шифру «Кипарис-512» є надвисокий рівень стійкості шифру, що дозволить йому застосовуватися у постквантовий період.

Результати для інших алгоритмів значно не змінилися у порівнянні з результатами, отриманими на 32-бітовій платформі, лише швидкість зашифрування алгоритму SPARX-128/128 помітно зросла з 749 до 937 Мбіт/с.

Таблиця 3

Порівняння продуктивності малоресурсних блокових шифрів на платформі Linux (64 bit), процесор Intel Core i5-4670U

Блоковий шифр	Розмір блока, біт	Довжина ключа, біт	Швидкість зашифрування, Мбіт/с	Реалізація
«Кипарис-256»	256	256	8418,3	Власна
«Кипарис-512»	512	512	5356,9	Власна
AES-256	128	256	1920,75	[14]
SPECK	64	128	3179,49	[17]
SPECK	128	128	5276,39	[18]
SPARX	128	128	1049,53	[17]
ДСТУ ГОСТ 28147:2009	64	128	640,469	[14]

Згідно з результатами, представленими у табл. 3, співвідношення між швидкостями досліджуваних алгоритмів на 64-бітовій платформі Linux приблизно таке саме, як і на Windows 10 x64. Головною відмінністю є значне покращення результату для шифру «Кипарис-256», швидкість якого перевершила 8 Гбіт/с, що може бути пов'язано з використанням нової версії компілятора gcc version 5.4.0, який виконує певну оптимізацію. Далі йдуть шифри «Кипарис-512» та SPECK-128/128 з приблизно однаковим результатом у більш ніж 5 Гбіт/с.

У табл. 4, 5 представлено результати вимірювання швидкодії обраних алгоритмів на мобільній платформі Android.

Таблиця 4

Порівняння продуктивності малоресурсних блокових шифрів на платформі Android 4.2.2 (32-bits), процесор Mediatek MT6582

Блоковий шифр	Розмір блока, біт	Довжина ключа, біт	Швидкість зашифрування, Мбіт/с	Реалізація
«Кипарис-256»	256	256	592	Власна
«Кипарис-512»	512	512	467	Власна
AES-256	128	256	109	[14]
SPECK	64	128	599	[17]
SPECK	128	128	205	[18]
SPARX	128	128	71	[17]

Таблиця 5

Порівняння продуктивності малоресурсних блокових шифрів на платформі Android 8.0.0 (64-bits), процесор Exynos 7880

Блоковий шифр	Розмір блока, біт	Довжина ключа, біт	Швидкість зашифрування, Мбіт/с	Реалізація
«Кипарис-256»	256	256	1263	Власна
«Кипарис-512»	512	512	999	Власна
AES-256	128	256	183	[14]
SPECK	64	128	639	[17]
SPECK	128	128	422	[18]
SPARX	128	128	145	[17]

На мобільних платформах «Кипарис» також продемонстрував високі результати. Так, наприклад, на процесорі Exynos 7880 «Кипарис-256» та «Кипарис-512» мають майже 1,3 Гбіт/с та 1 Гбіт/с відповідно, в той час, коли найближчий конкурент SPECK-64/128 має лише 640 Мбіт/с.

На рис. 1, 2 у графічному вигляді подано результати, представлені в табл. 1 – 5.

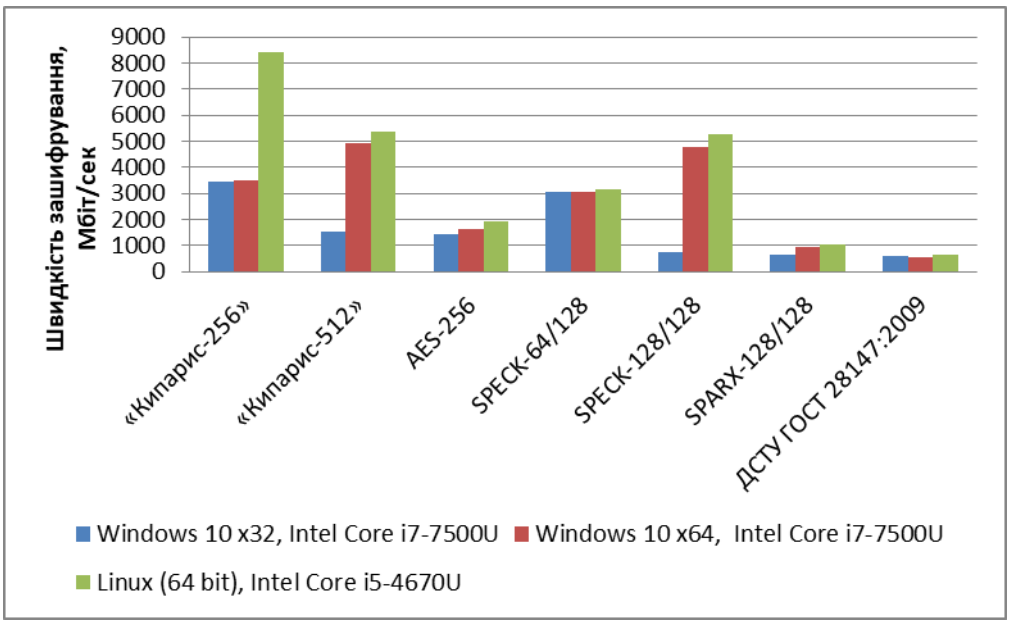

Рис. 1. Порівняння швидкодії шифру «Кипарис» з відомими блоковими шифрами на платформах загального призначення

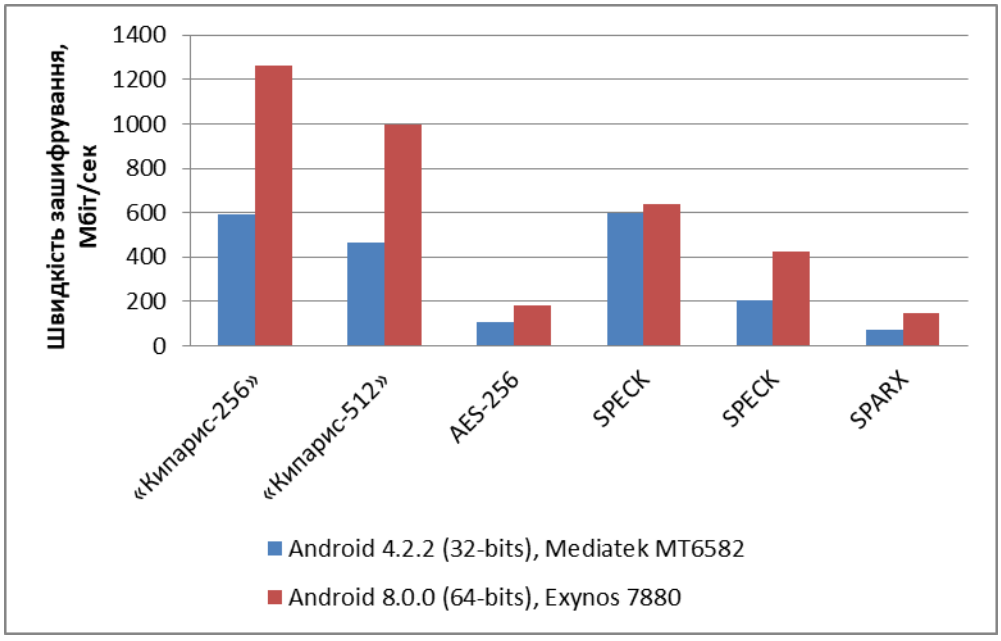

Рис. 2. Порівняння швидкодії шифру «Кипарис» з відомими блоковими шифрами на платформі Android

Висновки

1. Порівняння швидкодії блокового шифру «Кипарис» зі швидкодією відомих малоресурсних алгоритмів здійснювалося із використанням програмної реалізації, розробленої мовою програмування С++, що дозволяє отримати високу продуктивність (на базі нативного коду) за рахунок використання машинно-незалежної мови програмування.

2. Блоковий шифр «Кипарис» продемонстрував високу продуктивність на всіх досліджуваних програмно-апаратних платформах:

а) на платформі Windows 10 з 32-бітовою архітектурою найкращий результат показав шифр «Кипарис-256» (трохи менше 3,5 Гбіт/с), за ним слідує SPECK-128/128 (3 Гбіт/с), а шифр AES-256 відстає майже у 2,5 рази (1,5 Гбіт/с);

б) на платформі Windows 10 з 64-бітовою архітектурою блоковий найкращий результат показав шифр «Кипарис-512» (майже 5 Гбіт/с), якому незначно поступився за швидкістю шифр SPECK-128/128 (4,8 Гбіт/с); при цьому блоковий шифр «Кипарис-512» забезпечує надвисокий рівень стійкості;

в) на платформі Linux з 64-бітовою архітектурою блоковий шифр «Кипарис-256» показав надвисокий результат зі швидкодії (понад 8 Гбіт/с), далі з приблизно однаковим результатом слідують шифри «Кипарис-512» та SPECK-128/128 (понад 5 Гбіт/с);

г) на платформі Android 8.0.0 найкращими також були блокові шифри «Кипарис-256» та «Кипарис-512» (1,3 Гбіт/с та 1 Гбіт/с відповідно), за якими слідує SPECK-64/128 з результатом у 0,6 Гбіт/с.

3. Загалом, з точки зору продуктивності та зручності реалізації на різних програмно-апаратних платформах алгоритм «Кипарис» має наступні переваги:

а) два варіанти шифру («Кипарис-256» та «Кипарис-512») орієнтовані на 32-бітову та 64-бітову архітектури відповідно;

б) висока швидкодія перетворень незалежно від платформи, що використовується;

в) компактна реалізація незалежно від платформи, що використовується (сервер, робоча станція або мобільний пристрій);

г) мінімальний необхідний об’єм пам’яті для швидкодіючої реалізації, відсутність необхідності у таблицях передобчислень;

д) можливість організації ефективних захищених високошвидкісних каналів зв’язку між мобільними системами та серверами, у тому числі тими, що використовують апаратні прискорювачі.

Список літератури:

1. Lightweight cryptography. Project overview. NIST: веб-сайт. URL: https://csrc.nist.gov/projects/lightweight-cryptography.

2. Submission Requirements and Evaluation Criteria for the Lightweight Cryptography Standardization Process. NIST: веб-сайт. URL: https://csrc.nist.gov/CSRC/media /Projects/Lightweight-Cryptography/documents/final-lwc-submission-requirements-august2018.pdf.

3. Daniel Dinu, et al. Design strategies for ARX with provable bounds: Sparx and LAX // International Conference on the Theory and Application of Cryptology and Information Security, Springer, Berlin, Heidelberg, 2016. P. 484-513.

4. Taizo Shirai, et al. The 128-bit blockcipher CLEFIA // International workshop on fast software encryption. Springer, Berlin, Heidelberg, 2007. P. 181-195.

5. Suzaki T., Minematsu K., Morioka S., et al. Twine: A lightweight, versatile block cipher // ECRYPT Workshop on Lightweight Cryptography, LC11, 2011, P. 146–169.

6. Gong Z., Nikova S., Law Y. W. KLEIN: a new family of lightweight block ciphers // International Workshop on Radio Frequency Identification: Security and Privacy Issues. Springer, Berlin, Heidelberg, 2011. P. 1-18.

7. Banik S., et al. Midori: A block cipher for low energy // Advances in Cryptology – ASIACRYPT 2015: Proceedings of 21st International Conference on the Theory and Application of Cryptology and Information Security, 2015, Auckland, New Zealand. Part II. Vol. 9453 of LNCS, Springer, Berlin, Heidelberg, 2015. P. 411-436.

8. A. Bogdanov, et al. PRESENT: An Ultra-Lightweight Block Cipher. Springer, Berlin, Heidelberg, 2007. P. 450-466.

9. Borghoff J., et al. PRINCE – A Low-Latency Block Cipher for Pervasive Computing Applications – Extended Abstract // Advances in Cryptology – ASIACRYPT 2012: Proceedings of 18th International Conference on the Theory and Application of Cryptology and Information Security, 2-6 Dec., 2012, Beijing, China, Vol. 7658 of LNCS. Springer, Berlin, Heidelberg, 2012. P. 208-225.

10. Beaulieu R., et al. The SIMON and SPECK lightweight block ciphers // Design Automation Conference (DAC), 2015 52nd ACM/EDAC/IEEE. IEEE, 2015. P. 1-6.

11. Wheeler D. J. and Needham R. M. TEA, a Tiny Encryption Algorithm // International Workshop on Fast Software Encryption. Springer, Heidelberg, 1995. P. 363–366.

12. Needham R. M., Wheeler D. J. TEA extensions // Technical report, the Computer Laboratory, University of Cambridge, 1997.

13. Родінко М.Ю., Олійников Р.В. Постквантовий малоресурсний симетричний блоковий шифр «Кипарис» // Радіотехніка. 2017. Вип. 189. С. 100-107.

14. Roman-Oliynykov/ciphers-speed: веб-сайт. URL: https://github.com/Roman-Oliynykov/ciphers-speed.

15. Pub, NIST FIPS. 197: Advanced encryption standard (AES), Federal information processing standards publication 197.441: 0311, 2001.

16. ДСТУ ГОСТ 28147: 2009. Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования (ГОСТ 28147-89).

17. FELICS. Cryptolux. URL: https://www.cryptolux.org/index.php/FELICS. 18. Ray Beaulieu et al. The SIMON and SPECK Families of Lightweight Block Ciphers, IACR, 19 June, 2013, URL: https://eprint.iacr.org/2013/404.

Харківський національний
університет імені В.Н. Каразіна

Надійшла до редколегії 09.01.2020