КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 4 (12), 2021
ISSN 2663 - 4023
DOI 10.28925/2663-4023.2021.12.151162
УДК 004.056.5:004.7

Гнатюк Сергій Олександрович
д.т.н., доцент, заступник декана факультету кібербезпеки, комп'ютерної та програмної інженерії
Національний авіаційний університет, Київ, Україна
ORCID ID: 0000-0003-4992-0564
s.gnatyuk@nau.edu.ua

Бурмак Юлія Анатоліївна
здобувач НДЛ протидії кіберзагрозам в авіаційній галузі
Національний авіаційний університет, Київ, Україна
ORCID ID: 0000-0002-5410-6260
julburmac@gmail.com

Бердибаєв Рат Шиндалійович
керівник науково-технічного центру проблем інформаційної безпеки імені Турганбека Омара
Алматинський університет енергетики та звязку, Алмати, Казахстан
ORCID ID: 0000-0002-8341-9645
r.berdybaev@aues.kz

Александер Марек Богуслав
д.т.н., професор
Державна вища технічна школа у Новому Сончі, Польща
ORCID ID: 0000-0003-2619-1063
aleksandermarek4@gmail.com

Оспанова Дінара Манапівна
PhD докторант
Казахський гуманітарно-юридичний інноваційний університет, Семей, Казахстан
ORCID ID: 0000-0002-2206-7367
odm-1778@mail.ru

МЕТОД ПОБУДОВИ ГЕНЕРАТОРІВ ПСЕВДОВИПАДКОВИХ ПОСЛІДОВНОСТЕЙ ДЛЯ КРИПТОГРАФІЧНИХ ЗАСТОСУВАНЬ У 5G МЕРЕЖАХ

Анотація. Сьогодні генератори псевдовипадкових чисел використовуються у різних системах і застосунках, у тому числі як генератори ключів у потокових шифрах. Впровадження новітніх інформаційно-комунікаційних технологій (зокрема, 5G мереж) підсилює вимоги до забезпечення конфіденційності критичних даних і змушує розробляти нові методи та засоби криптографічного захисту. Існуючі генератори, як і інші криптографічні алгоритми, не задовольняють вимогам за швидкістю обробки та стійкістю до відомих видів атак. З огляду на це, у цій статті розроблено метод побудови генераторів псевдовипадкових послідовностей, який дозволив будувати ефективні генератори для криптографічних застосувань. На основі цього методу розроблено і реалізовано у вигляді програмних застосунків генератори псевдовипадкових послідовностей, які будуть корисними для криптографічних застосувань в сучасних 5G мережах. Розроблені генератори псевдовипадкових послідовностей пройшли комплексне статистичне тестування за методикою NIST STS (показали результати, не гірші за результати відомих генераторів псевдовипадкових послідовностей, що використовуються на практиці для розв'язання аналогічних задач). Крім того, вони є більш швидкими у порівнянні з аналогами, які сьогодні використовується у 5G мережах (наприклад, з алгоритмами SNOW та Trivium). У подальших роботах планується дослідити стійкість розроблених генераторів псевдовипадкових послідовностей до різних типів криптоаналітичних атак, а також провести моделювання роботи розроблених програмних генераторів псевдовипадкових послідовностей з використанням обладнання базових станцій сучасних 5G мереж.

Ключові слова: захист інформації, псевдовипадкова послідовність, гамма, криптографічні застосування, 5G мережі, генератор, криптографічна стійкість, швидкість, NIST STS.

ISSN 2663 - 4023

ВСТУП

Генератори псевдовипадкових чисел можуть використовуватися як генератори ключів у потокових шифрах [1]. Метою використання генераторів є формування нескінченного ключового слова, використовуючи відносно малу довжину самого ключа. Генератор псевдовипадкових чисел створює послідовність бітів, схожу на випадкову (за статистичними параметрами). Насправді, такі послідовності обчислюються за певними правилами і не є випадковими, тому вони можуть бути абсолютно точно відтворені як на стороні що передає, так і на приймаючій стороні. Послідовність ключових символів, що використовується при шифруванні, має бути не лише досить довгою. Якщо генератор ключів при кожному включенні створює одну і ту ж послідовність бітів, то зламати таку систему буде можливо. Отже, вихід генератора потоку ключів має бути функцією ключа. У цьому випадку розшифрувати і прочитати повідомлення можна буде лише з використанням того ж ключа, який використовувався при шифруванні.

Можна сформувати три основні вимоги, якими повинні задовольняти криптографічно стійкі генератори псевдовипадкових послідовностей або гамми:

1) Період гамми має бути значним для шифрування повідомлень різної довжини.

2) Гамма має бути важко передбаченою. Це означає, що якщо відомі тип генератора і шматок гамми, то неможливо передбачити наступний за цим шматком біт гамми або передуючий цьому шматку біт гамми.

3) Генерування гамми не має бути пов'язане з великими технічними і організаційними труднощами.

Для генерування дійсно випадкової послідовності за допомогою комп'ютера необхідно використовувати його апаратні засоби. Ці засоби можуть фіксувати наступні явища: шум від напівпровідникових приладів; біти оцифрованого звуку з мікрофону; інтервали між перериванням зовнішніх або внутрішніх пристроїв; інтервали між натисканням клавіш; температура повітря на апаратних складових. Також існують генератори випадкових чисел у вигляді плат або зовнішніх пристроїв. Такі генератори використовуються в сучасних криптосистемах для військових. Вони підключаються до комп’ютерів за допомогою портів вводу-виводу. Основні джерела для них слугують: білий Гаусівський шум; запис радіоефіру; виміри теплових флуктуацій тощо.

Незважаючи на те, що для проектування генераторів псевдовипадкових наборів необхідне комплексе тестування на випадковість, вони в багатьох випадках знаходять використання в комп'ютерних програмах прикладного характеру. Також можуть бути реалізовані для будь-яких типів комп'ютерних систем. Особливої актуальності ці питання набувають із впровадженням нових підвищених вимог конфіденційності (приватності) в сучасних мережах 5G (рис. 1) [2].

Security Elements		WiFi 6 (WPA3)	5G	4G
Terminal identity identification	RADIUS	Terminal - authentication server (AAA)	Terminal - 3GPP AAA	
	Identity identification	Enterprise-level certificate, username/password	Identify authentication information in SIM cards	
Authentication	Security authentication modes	EAP-TLS (certificate) EAP-PEAP (username/password) EAP-AKA (SIM cards on mobile phones)	3GPP and non-3GPP: 5G AKA and EAP-AKA'	3GPP: EPS AKA Non-3GPP: EAP-AKA and EAP-AKA'
Authentication flexibility	Authentication modes in different scenarios	Third-party system authentication such as Portal authentication and WeChat authentication	None	None
NAS security algorithm negotiation	Encryption algorithm	AES	Snow3G, AES, and ZuC	Snow3G, AES, and ZuC
	Algorithm key	256 bits (128 bits for WPA2)	256 bits	128 bits

Рис.1. Порівняння процедур забезпечення конфіденційності в різних мережах

ISSN 2663 - 4023

Зважаючи на зазначене, *метою цієї роботи* є розробка й дослідження ефективного методу генерування псевдовипадкових послідовностей для криптографічних застосувань у сучасних 5G мережах.

ОСНОВНА ЧАСТИНА ДОСЛІДЖЕННЯ

Теоретичне обґрунтування розробленого методу

Прототипом для методу побудови генераторів псевдовипадкових послідовностей було обрано відомий генератор Trivium [3]. Порівняно з прототипом було змінено:

1. Введено параметри n, t, e, k, при фіксації яких формується нова структура генератора псевдовипадкових послідовностей (змінюється розрядність операцій). Всі операції виконуються не над бітами, а над векторами певного розміру (байтами).
2. Для підвищення показників не лінійності введено використання операції підстановки $S(x)$. Слід зауважити, що для кожного нового генератора псевдовипадкових послідовностей можна задати свою унікальну операцію підстановки $S(x)$.
3. Для генерації псевдовипадкових послідовностей використовується вектор внутрішнього стану генератора E_i, ключовий вектор для генерації послідовності K та індекс поточної ітерація генерування i.
4. У функції генерування F_{gen} змінено етап ініціалізація змінних, введено операцію динамічного циклічного зсуву та операцію підстановки.
5. У функції генерування F_{gen} запропоновано використання незалежних функцій F_A, F_B, F_C і F_D, що залежать від значень вектору внутрішнього попереднього етапу генерації, ключового вектора K та індексу поточної ітерація генерування i. Виходом функцій F_A, F_B, F_C і F_D будуть дані необхідного розміру (розмір даних входу і виходу будуть різними). Слід зауважити, що для кожного нового генератора псевдовипадкових послідовностей можна задати свої унікальні функції F_A, F_B, F_C і F_D. По суті дані функції це є окремі байт орієнтовані генератори послідовностей (не обов'язково криптостійкі), які можуть працювати паралельно, тому для кращої оптимізації швидкість генерації послідовностей в цих функціях має бути приблизно однаковою.
6. Змінений фінальний крок формування послідовності m, змінена послідовність операція, введено використання операції підстановки.

При зміні/фіксації параметрів n, t, e, k визначенні функцій F_A, F_B, F_C і F_D та $S(x)$ можна будувати різноманітні генератори псевдовипадкових послідовностей.

Опис методу побудови генераторів псевдовипадкових послідовностей

Нехай $n, t \in Z_+$, тоді для генерації псевдовипадкової послідовності M, $M \in V_N$, $V_N \in \{0,1\}^N$, довжиною $N = n \cdot t$ біт, потрібно сформувати t послідовностей довжиною n біт кожна:

$$M = (m_1, m_2, ..., m_{t-1}, m_t),\ m_i \in V_n,\ i = \overline{1,t}.$$

Процес генерації кожного m_i, $m_i \in V_n$, $i = \overline{1,t}$, відбувається наступним чином:

ISSN 2663 - 4023

$$m_i, E_i = F_{gen}\left(E_{i-1}, K, i\right),\ i = \overline{1,t},$$

де E_i – вектор внутрішнього стану генератора після генерації i-го m_i, $E_i \in V_e$, $e \in Z_+$, $E_0 = IV$, IV – вектор ініціалізації, $IV \in V_e$, K – ключовий вектор для генерації послідовності, $K \in V_k$, $k \in Z_+$, F_{gen} - функція генерації послідовності m_i.

Функція $F_{gen}\left(E, K, i\right)$ виконується в два етапи:

1) ініціалізація змінних;
2) формування послідовності.

Етап 1 функції $F_{gen}\left(E, K, i\right)$. На початку виконується обробка вектора внутрішнього стану E, $E \in V_e$:

$$E = S\left(E\right) <<< i,$$

де $x <<< y$ – операція правого побітового циклічного зсуву аргументу x на y − біт, $S\left(x\right)$ − деяка операція підстановки.

Далі вектор внутрішнього стану генератора E і ключовий вектор K розкладаються на 4 частини:

$$E = \left(E_a, E_b, E_c, E_d\right),\ E \in V_e,\ e = a + b + c + d,$$
$$E_a \in V_a,\ E_b \in V_b,\ E_c \in V_c,\ E_d \in V_d,\ a, b, c, d \in Z_+,$$
$$K = \left(K_a, K_b, K_c, K_d\right),\ K \in V_k,\ k = a' + b' + c' + d',$$
$$K_a \in V_{a'},\ K_b \in V_{b'},\ K_c \in V_{c'},\ K_d \in V_{d'},\ a', b', c', d' \in Z_+.$$

Вектори E_a, E_b, E_c, E_d і K_a, K_b, K_c, K_d будуть використовуватись в наступному етапі функції $F_{gen}\left(E, K, i\right)$.

Етап 2 функції $F_{gen}\left(E, K, i\right)$. На даному етапі виконується формування послідовності m, $m \in V_n$. Для цього використовуються чотири додаткових функції $F_A\left(E_a, K_a, i\right)$, $F_B\left(E_b, K_b, i\right)$, $F_C\left(E_c, K_c, i\right)$ і $F_D\left(E_d, K_d, i\right)$, функції F_A, F_B, F_C і F_D − деякі функції, що на вхід приймають значення певного вектора внутрішнього стану і ключового вектора, а на вихід передається послідовність довжини n біт (ці функції можуть бути побудовані на основі нелінійних регістрів зсуву, блокових і потокових шифрів, геш-функцій тощо).

Тоді, процес генерації послідовності m, $m \in V_n$ та нового значення вектора внутрішнього стану E, $E \in V_e$ в функції $F_{gen}\left(E, K, i\right)$ буде таким:

Крок 1. Сформувати додаткові вектори A, B, C і D, та отримати нові значення векторів E_a, E_b, E_c і E_d:

$$A, E_a = F_A\left(E_a, K_a, i\right),\ A \in V_n,\ E_a \in V_a,$$
$$B, E_b = F_B\left(E_b, K_b, i\right),\ B \in V_n,\ E_b \in V_b,$$
$$C, E_c = F_C\left(E_c, K_c, i\right),\ C \in V_n,\ E_c \in V_c,$$
$$D, E_d = F_D\left(E_d, K_d, i\right),\ D \in V_n,\ E_d \in V_d.$$

Крок 2. Розрахувати нове значення вектору внутрішнього стану E, $E \in V_e$:

$$E = \left(E_b, E_d, E_a, E_c\right).$$

Крок 3. Сформувати послідовності m, $m \in V_n$:

$$AB = A <<< B,\ AB \in V_n,$$
$$CD = C <<< D,\ CD \in V_n,$$
$$BC = B + CD,\ BC \in V_n,$$
$$AD = AB \oplus D,\ AD \in V_n,$$
$$m = \overline{AD} \oplus S\left(BC\right),\ m \in V_n,$$

де $\oplus$ і $+$ відповідають відповідно операціям додавання за модулем 2 і 2^n, $S\left(x\right)$– операція підстановки.

Виходом функції $F_{gen}\left(E, K, i\right)$ будуть вектори m, $m \in V_n$ і E, $E \in V_e$:

$$\left(m, E\right) = F_{gen}\left(E, K, i\right).$$

Отже, у даному підрозділі запропоновано новий метод побудови генераторів псевдовипадкових послідовностей, який за рахунок обробки вектора внутрішнього стану та ключового вектору операціями підстановки, циклічного зсуву, складання за модулем 2 і 2^n та 4-ма нелінійними функціями, дозволить будувати ефективні генератори псевдовипадкових послідовностей (при фіксації параметрів n, t, e, k визначенні функцій F_A, F_B, F_C і F_D та $S(x)$.

Розробка генераторів на базі запропонованого методу

На базі запропонованого методу (описаного у попередньому підрозділі статті) було розроблено три генератори: 5Gen-1, 5Gen-2 та 5Gen-3.

5Gen-1 був спроектований з такими параметрами $n = 128$, $a = 128$, $b = 100$, $c = 111$, $d = 173$, $e = a + b + c + d = 512$, $a' = 128$, $b' = 128$, $c' = 128$, $d' = 128$, $k = a' + b' + c' + d' = 512$. F_A, F_B, F_C і F_D – функції, що побудовані на основі нелінійних регістрів зсуву. У якості операції $S(x)$ – використовується операція виду: $S(x) = \left(s_0\left(x_{31}\right), ..., s_0\left(x_0\right)\right)$, де $x_j \in V_{16}$, $j = \overline{0,31}$, S_0 – підстановка на множині V_{16}.

Підстановка S_0 побудована з параметрами, що наведені у табл. 1.

Таблиця 1

Параметри для побудови таблиці замін S_0 алгоритму 5Gen-1

M	*C*	*V*
{ 903, 3206, 640C, C818, 9031, 2063, 40C6, 818C, 319, 632, C64, 18C8, 3190, 6320, C640, 8C81 }	6D71	19CF

5Gen-2 був спроектований з такими параметрами $n = 256$, $a = 138$, $b = 120$, $c = 116$, $d = 138$, $e = a + b + c + d = 512$, $a' = 128$, $b' = 128$, $c' = 128$, $d' = 128$, $k = a' + b' + c' + d' = 512$. F_A, F_B, F_C і F_D – функції, що побудовані на основі

нелінійних регістрів зсуву. У якості операції $S(x)$ – використовується операція виду: $S(x) = \left(s_7\left(x_{63}\right), ..., s_0\left(x_0\right)\right)$, де $x_j \in V_8$, $j = \overline{0,63}$, s_b – підстановка на множині V_8, $b = \overline{0,7}$ (почергово використовуються 8 різних таблиць замін).

Підстановка s_i, $i = \overline{0,7}$ побудована з параметрами, що наведені у табл. 2-9:

Таблиця 2

Параметри для побудови таблиці замін s_0 алгоритму 5Gen-2

M	*C*	*V*
{ 29, 52, A4, 49, 92, 25, 4a, 94 }	07	D8

Таблиця 3

Параметри для побудови таблиці замін s_1 алгоритму 5Gen-2

M	*C*	*V*
{ 70, E0, C1, 83, 7, E, 1C, 38 }	A2	44

Таблиця 4

Параметри для побудови таблиці замін s_2 алгоритму 5Gen-2

M	*C*	*V*
{ 3E, 7C, F8, F1, E3, C7, 8F, 1F }	72	63

Таблиця 5

Параметри для побудови таблиці замін s_3 алгоритму 5Gen-2

M	*C*	*V*
{ E3, C7, 8F, 1F, 3E, 7C, F8, F1 }	43	9B

Таблиця 6

Параметри для побудови таблиці замін s_4 алгоритму 5Gen-2

M	*C*	*V*
{ E5, CB, 97, 2F, 5E, BC, 79, F2 }	A0	8C

Таблиця 7

Параметри для побудови таблиці замін s_5 алгоритму 5Gen-2

M	*C*	*V*
{ AB, 57, AE, 5D, BA, 75, EA, D5 }	7B	C6

Таблиця 8

Параметри для побудови таблиці замін s_6 алгоритму 5Gen-2

M	*C*	*V*
{ 91, 23, 46, 8C, 19, 32, 64, C8 }	ED	B0

Таблиця 9

Параметри для побудови таблиці замін S_7 алгоритму 5Gen-2

M	*C*	*V*
{ F8, F1, E3, C7, 8F, 1F, 3E, 7C}	18	75

5Gen-3 був спроектований з такими параметрами $n=128$, $a=128$, $b=128$, $c=128$, $d=128$, $e=a+b+c+d=512$, $a'=128$, $b'=128$, $c'=128$, $d'=128$, $k=a'+b'+c'+d'=512$. F_A, F_B, F_C і F_D – функції, що побудовані на основі AES-128. У якості операції $S(x)$ – використовується операція виду: $S(x)=\left(S_0\left(x_{63}\right),\ldots,S_0\left(x_0\right)\right)$, де $x_j \in V_8$, $j=\overline{0,63}$, S_0 – підстановка на множині V_8.

Підстановка S_0 побудована з параметрами, що наведені у табл. 10.

Таблиця 10

Параметри для побудови таблиці замін S_0 алгоритму 5Gen-3

M	*C*	*V*
{ 20, 40, 80, 1, 2, 4, 8, 10}	59	6B

2.3. Практичне дослідження розробленого методу

Дослідження статистичних параметрів генераторів

Статистичні характеристики розроблених генераторів досліджувались за методикою NIST STS [5], а результати порівнювались із результатами генератора BBS, функціями гешування SHA-256, SHA-512, потоковими шифрами SNOW і Trivium [3,4] (використовуються у 5G мережах). Для цього дослідження, на основі розроблених хеш-функцій і функцій SHA-256 та SHA-512, були побудовані генератори для створення файлів необхідної довжини для статистичних тестів NIST STS.

Рис.2. Статистичний портрет алгоритму 5Gen-1 за методикою NIST STS

ISSN 2663 - 4023

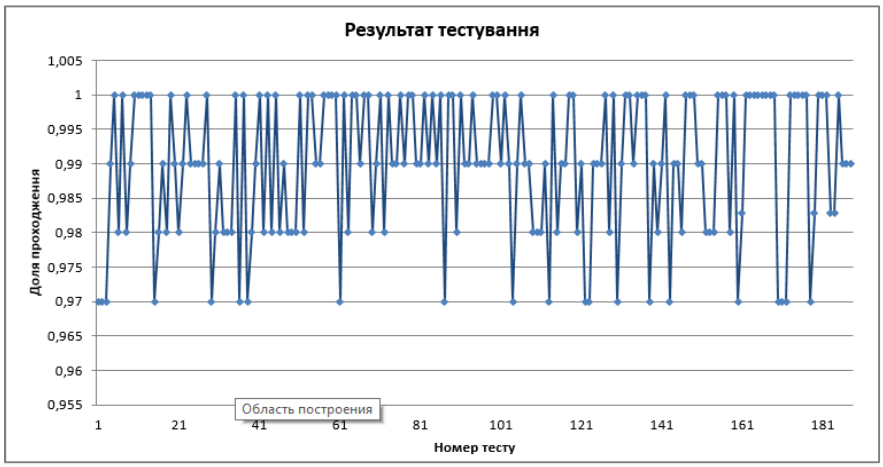

Рис.3. Статистичний портрет алгоритму 5Gen-2 за методикою NIST STS

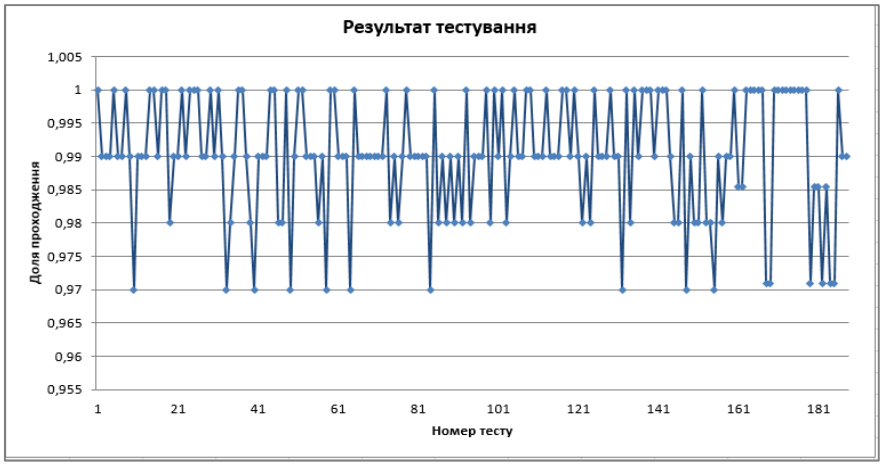

Рис.4. Статистичний портрет алгоритму 5Gen-3 за методикою NIST STS

Для здійснення тестувань були обрані такі параметри:

1. Довжина послідовності, що тестується n = 10^6 біт;
2. Кількість послідовностей, що тестується m = 100;
3. Рівень значущості α = 0,01.
4. Кількість тестів q = 188

Таким чином, обсяг вибірки, що тестується, склав $N = 10^6 \times 100 = 10^8$ біт, кількість тестів (q) для різних довжин q = 188, таким чином, статистичний портрет генератора містить 18800 значень імовірності P. В ідеальному випадку при m = 100 і α = 0,01 у ході тестування може бути відкинута тільки одна послідовність зі ста, тобто коефіцієнт проходження кожного тесту має складати 99%. Але це занадто жорстке правило. Тому застосовується правило на основі довірчого інтервалу. Нижня межа дорівнює 0,96015. Для кожного алгоритму генерувалось 10 файлів із послідовностями розміром 100 Мбіт, які і досліджувались за методикою NIST STS [6].

Як видно з результатів (рис.2-4, табл. 11), розроблені алгоритми пройшли комплексний контроль за методикою NIST STS та показали не гірші, а в деяких випадках і кращі, результати ніж відомі алгоритми [7].

Таблиця 11

Результати дослідження за методикою NIST STS

Генератор ПВП	Кількість тестів, у яких тестування пройшло	
	99% послід.	96% послід.
BBS	133.4 (70.96%)	188 (100%)
SHA-256	132.2 (70.32%)	187.9 (99.94%)
SHA-512	134.3 (71.44%)	188 (100%)
SNOW	134.8 (71.70%)	188 (100%)
Trivium	130.1 (69.20%)	187.6 (99.78%)
5Gen-1	134.1 (71,32%)	188 (100%)
5Gen-2	136.4 (72,55%)	187.8 (99.89%)
5Gen-3	137.3 (73,03%)	188 (100%)

Дослідження швидкісних параметрів генераторів

Для дослідження розроблених генераторів псевдовипадкових послідовностей 5Gen-1, 5Gen-2, 5Gen-3 – дані алгоритми були програмно реалізовані на мові програмування C++. Результати порівнювались із генератором псевдовипадкових послідовностей SNOW. Для дослідження були випадковим чином обрані кілька файлів різного розміру (файли розміром 1 МБ, 10 МБ, 100 МБ), кожен з яких зашифровувався досліджуваними алгоритмами, при цьому замірявся час обробки. Кожен файл оброблявся 10 разів кожним алгоритмом. Дослідження проводилися в однакових умовах на Intel Core i3-3220 3.3GHz. Усереднені результати наведено у табл. 12.

Таблиця 12

Результати дослідження швидкісних характеристик генераторів ПВП

Функцій гешування	Файл 1, 1 МБ		Файл 2, 10 МБ		Файл 3, 100 МБ	
	t ,с	v ,МБ/с	t ,с	v ,МБ/с	t ,с	v ,МБ/с
SNOW	0.011	90.91	0.107	93.46	1.01	99.01
5Gen-1	0.009	111.11	0.091	109.89	0.88	113.64
5Gen-2	0.010	100.00	0.098	102.04	0.92	108.70
5Gen-3	0.014	71.43	0.112	89.29	0.99	101.01

Згідно з результатами дослідження швидкість алгоритму 5Gen є вищою за SNOW до 21% (за виключенням двох результатів алгоритму 5Gen-3).

ВИСНОВКИ

У цій роботі розроблено метод побудови генераторів псевдовипадкових послідовностей, який за рахунок обробки вектора внутрішнього стану та ключового вектору операціями підстановки, циклічного зсуву, складання за модулем 2 і 2^n та чотирма нелінійними функціями, дозволив будувати ефективні генератори псевдовипадкових послідовностей.

На основі цього методу розроблено і реалізовано програмно три генератори псевдовипадкових послідовностей 5Gen, які будуть корисними як для функцій гешування, так і для інших криптографічних застосувань (генерування ключів, потокові

шифри тощо) в сучасних 5G мережах. Розроблені генератори пройшли комплексне статистичне тестування за методикою NIST STS, крім того, вони є більш швидкими у порівнянні з аналогами (зокрема, до 21% швидші у порівнянні з алгоритмом Snow, який використовується у 5G мережах).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Hu, Z., Gnatyuk, S., Okhrimenko, T., Tynymbayev, S., & Iavich, M. (2020). High-Speed and secure PRNG for cryptographic applications. *International Journal of Computer Network and Information Security*, *12*(3), 1–10. https://doi.org/10.5815/ijcnis.2020.03.01
2. Security Comparison Between Wi-Fi 6 and 5G. https://forum.huawei.com/ enterprise/en/security-comparison-between-wi-fi-6-and-5g/thread/615836-869
3. De Cannière C., Preneel B. (2005). TRIVIUM – Specifications. *eSTREAM, ECRYPT Stream Cipher Project*, Report 2005/030. http://www.ecrypt.eu.org/stream
4. Ekdahl, P., Johansson, T. (2000). SNOW. A new stream cipher. *Proceedings of the First NESSIE Workshop*.
5. Bassham, L. E., Rukhin, A. L., Soto, J., Nechvatal, J. R., Smid, M. E., Barker, E. B., Leigh, S. D., Levenson, M., Vangel, M., Banks, D. L., Heckert, N. A., Dray, J. F., & Vo, S. (2010). *A statistical test suite for random and pseudorandom number generators for cryptographic applications*. National Institute of Standards and Technology. https://doi.org/10.6028/nist.sp.800-22r1a
6. Gnatyuk, S., Okhrimenko, T., Azarenko, O., Fesenko, A., & Berdibayev, R. (2020). Experimental study of secure PRNG for q-trits quantum cryptography protocols. У *2020 IEEE 11th international conference on dependable systems, services and technologies (DESSERT)*. IEEE. https://doi.org/10.1109/dessert50317.2020.9125007
7. Mcginthy, J. M., & Michaels, A. J. (2019). Further analysis of prng-based key derivation functions. *IEEE Access*, *7*, 95978–95986. https://doi.org/10.1109/access.2019.2928768

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 4 (12), 2021
ISSN 2663 - 4023

Sergiy O. Gnatyuk
DSc, Associate Professor,
Vice-Dean of the Faculty of Cybersecurity, Computer and Software Engineering
National Aviation University, Kyiv, Ukraine
ORCID ID: 0000-0003-4992-0564
s.gnatyuk@nau.edu.ua,

Yuliia A. Burmak
PhD degree applicant in NAU Cybersecurity R&D Lab
National Aviation University, Kyiv, Ukraine
ORCID ID: 0000-0002-5410-6260
julburmac@gmail.com

Rat Sh. Berdibayev
Chair of Scientific and Technical Center of Information Security Problems n.a. Turganbek Omar
Almaty University of Power Energy and Telecommunication, Almaty, Kazakhstan
ORCID ID: 0000-0002-8341-9645
r.berdybaev@aues.kz

Marek B. Aleksander
DSc, Professor
State University of Applied Sciences in Nowy Sącz, Poland
ORCID ID: 0000-0003-2619-1063
aleksandermarek4@gmail.com

Dinara M. Ospanova
PhD Student
Kazakh Humanitarian Juridical Innovative University, Semey, Kazakhstan
ORCID ID: 0000-0002-2206-7367
odm-1778@mail.ru

METHOD FOR DEVELOPING PSEUDO-RANDOM NUMBER GENERATORS FOR CRYPTOGRAPHIC APPLICATIONS IN 5G NETWORKS

Abstract. Today, pseudo-random number generators are used in various systems and applications, including as key generators in stream ciphers. The implementation of the latest information and communication technologies (in particular, 5G networks) strengthens the requirements for ensuring the confidentiality of critical data and forces the development of new methods and means for cryptographic protection. Existing generators, like other cryptographic algorithms, do not meet the requirements for processing speed and security against known types of attacks. From this position, in the paper a method for constructing pseudo-random sequence generators was developed. It allows to build efficient generators for cryptographic applications. Based on this method, software generators of pseudo-random numbers have been developed and implemented. These will be useful for cryptographic applications in modern 5G networks. The developed pseudo-random number generators have passed complex statistical testing by the NIST STS technique (showed results not worse than the results of known pseudo-random sequence generators used in practice to solve similar problems). Besides, they are faster in comparison with analogues used today in 5G networks (for example, with algorithms SNOW and Trivium). In further works it is planned to investigate the security of the developed pseudo-random generators against different types of cryptanalytic attacks, as well as to simulate the work of the developed pseudo-random sequence generators using the base station equipment of modern 5G networks.

Keywords: information security, pseudo-random number, gamma, cryptographic applications, 5G network, generator, cryptographic security, speed, NIST STS.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Hu, Z., Gnatyuk, S., Okhrimenko, T., Tynymbayev, S., & Iavich, M. (2020). High-Speed and secure PRNG for cryptographic applications. *International Journal of Computer Network and Information Security*, *12*(3), 1–10. https://doi.org/10.5815/ijcnis.2020.03.01
2. Security Comparison Between Wi-Fi 6 and 5G. https://forum.huawei.com/ enterprise/en/security-comparison-between-wi-fi-6-and-5g/thread/615836-869
3. De Cannière C., Preneel B. (2005). TRIVIUM – Specifications. *eSTREAM, ECRYPT Stream Cipher Project*, Report 2005/030. http://www.ecrypt.eu.org/stream
4. Ekdahl, P., Johansson, T. (2000). SNOW. A new stream cipher. *Proceedings of the First NESSIE Workshop*.
5. Bassham, L. E., Rukhin, A. L., Soto, J., Nechvatal, J. R., Smid, M. E., Barker, E. B., Leigh, S. D., Levenson, M., Vangel, M., Banks, D. L., Heckert, N. A., Dray, J. F., & Vo, S. (2010). *A statistical test suite for random and pseudorandom number generators for cryptographic applications*. National Institute of Standards and Technology. https://doi.org/10.6028/nist.sp.800-22r1a
6. Gnatyuk, S., Okhrimenko, T., Azarenko, O., Fesenko, A., & Berdibayev, R. (2020). Experimental study of secure PRNG for q-trits quantum cryptography protocols. У *2020 IEEE 11th international conference on dependable systems, services and technologies (DESSERT)*. IEEE. https://doi.org/10.1109/dessert50317.2020.9125007
7. Mcginthy, J. M., & Michaels, A. J. (2019). Further analysis of prng-based key derivation functions. *IEEE Access*, *7*, 95978–95986. https://doi.org/10.1109/access.2019.2928768