УДК 004.056:061.68

Совин Я.Р., Чінка В.М., Хома В.В., Решетар Я.В.

ДОСЛІДЖЕННЯ КРИПТОГРАФІЧНОЇ ЯКОСТІ ВБУДОВАНОГО ГЕНЕРАТОРА ВИПАДКОВИХ ЧИСЕЛ У МІКРОКОНТРОЛЕРАХ STM32F4XX З ЯДРОМ ARM CORTEX-M4F

В статті проведений аналіз статистичної характеристики вбудованого ГІВЧ мікроконтролерів сімейства STM32F4xx з метою виявлення та усунення можливих недоліків, проведена оцінка продуктивність генератора та можливість його використання у вбудованих криптографічних аплікаціях.

Ключові слова: генератор випадкових чисел, криптографія, мікросхема, мікро контролер.

Вступ. Генератор випадкових чисел (ГВЧ) є важливим компонентом більшості криптографічних систем. Основними сферами застосування ГВЧ в криптографії є генерація ключів для симетричних і асиметричних криптоалгоритмів, вироблення випадкових повідомлень в протоколах автентифікації побудованих за схемою "запит-відповідь", формування бітів доповнення до потрібного розміру блоку, утворення векторів ініціалізації у блокових шифрах та масок для маскування проміжних результатів виконання криптоалгоритму з метою протидії атакам через сторонні канали [1, 2].

ГВЧ повинні задовольняти високі вимоги до їх характеристик, оскільки вразливість ГВЧ може спричинити компрометацію всієї криптосистеми, або значно ослабити її криптостійкість.

Ідеальний ГВЧ здатний генерувати випадкові послідовності чисел, які є статистично рівномірно розподілені, незалежні, непередбачувані та невідтворювані. Проте такий ГВЧ це лише математична абстракція, а ГВЧ, які використовуються в криптографії, лише певною мірою наближаються до неї і відповідно до цього умовно поділяються на три базові класи [2]:

- Генератори псевдовипадкових чисел (ГПВЧ). Побудовані на певному детермінованому алгоритмі, який ініціалізується зовнішньо згенерованим випадковим числом – так званим зародком (seed). Відповідно, при однакових значеннях зародку ГПВЧ завжди генерують однакові послідовності. Такі генератори є прості та дешеві в реалізації (особливо апаратній). Для забезпечення високого рівня захищеності ГПВЧ повинні періодично оновлювати значення зародку.
- Криптографічно захищені ГПВЧ (КЗГПВЧ). Базуються на ГПВЧ, але алгоритм, що служить для утворення випадкових чисел, робить неможливим в обчислювальному сенсі передбачення наступного значення, навіть якщо відомі сам алгоритм і попередні вихідні дані. З цією метою можуть використовуватися, наприклад, алгоритми симетричного шифрування.
- Генератори істинно випадкових чисел (ГІВЧ). ГІВЧ використовують або певний фізичний випадковий процес – теплові чи дробові шуми, фазовий джиттер або певні випадкові явища – дії користувача, системний час, вміст ОЗП, сигнал від мікрофонного входу і т.п. Враховуючи, що ГІВЧ використовують для генерації зародків у ГПВЧ та КЗГПВЧ, то можна стверджувати, що вони відіграють фундаментальну роль в забезпеченні захищеності всієї криптосистеми.

Попри велику кількість запропонованих у науковій літературі ГІВЧ питання побудови ефективного (з точки зору потрібних ресурсів, швидкодії та споживаної потужності) і надійного ГВЧ для вбудованих систем є актуальним з ряду причин.

У складі інтегральних схем (мікроконтролерів чи ПЛІС) доступні фізичні джерела випадкових процесів для ГІВЧ мають аналогову природу. Проте аналогові джерела шуму важко інтегрувати в цифрові обчислювальні засоби, оскільки це спричиняє збільшення розміру кристалу та споживаної потужності. Проблемою також є те, що всередині мікросхеми на аналогові кола ГІВЧ впливають близько розташовані цифрові кола та кола

живлення, які генерують періодичні завади значно вищого рівня ніж сам шум. Перенесення аналогової частини ГІВЧ на нову платформу чи технологічний процес потребує значних витрат коштів та часу на редизайн мікросхеми.

Використання зовнішніх джерел випадкових процесів є небажаним з точки зору захищеності від сторонніх впливів і складності організації інтерфейсу з ними.

З огляду на це у вбудованих системах, особливо на базі мікроконтролерів (МК), досить складно реалізувати високоякісний та захищений ГІВЧ, а тому, в основному, використовуються криптографічно слабкі ГПВЧ: різні варіанти лінійного конгруентного методу чи регістрів зсуву зі зворотними зв'язками.

Аналіз останніх досліджень і публікацій. В загальному випадку ГІВЧ складається з фізичного джерела випадкового сигналу, дискретизатора та блоку детермінованої постобробки. Джерело випадкового сигналу генерує неперервний аналоговий сигнал (шум), який бінарно оцифровується (наприклад, компаратором). У багатьох випадках отримана випадкова послідовність піддається алгоритмічній постобробці, з метою маскування потенційних статистичних дефектів, що виникають внаслідок обмеженої смуги пропускання, технологічного розкиду параметрів, температурного дрейфу, дій зловмисника і т.д. Цілком очевидно, що алгоритмічна постобробка приводить до зменшення продуктивності.

Двома найпоширенішими методами побудови ГІВЧ є безпосереднє підсилення і дискретизація шумового сигналу електронних компонентів (резисторів, діодів, стабілітронів) та використання частотної нестабільності несинхронізованих генераторів.

У роботі [3] описано типовий ГІВЧ, що використовує комбінацію аналогових і цифрових компонентів. Він складається з двох стабілітронів, які є джерелом білого шуму. Шумовий сигнал підсилюється диференційним підсилювачем та дискретизується з допомогою компаратора та тригера. Оскільки кола підсилення, які повинні підсилити рівень шуму до цифрових логічних рівнів, споживають достатньо багато потужності, мають аналоговий характер та вносять спотворення в сигнал інтеграція подібних ГІВЧ у МК або ПЛІС є проблематичною.

ГІВЧ на основі цифрових генераторів використовують два незалежні генератори (без стабілізації частоти), відмінні внутрішні шуми яких спричиняють джиттер фази t_J (короткочасні зміщення фронтів в часі), що і є джерелом випадковості [2]. Вихід високочастотного генератора (ВЧГ) дискретизується за зростаючим фронтом сигналу низькочастотного генератора (НЧГ) з допомогою D-тригера (рис. 1. а).

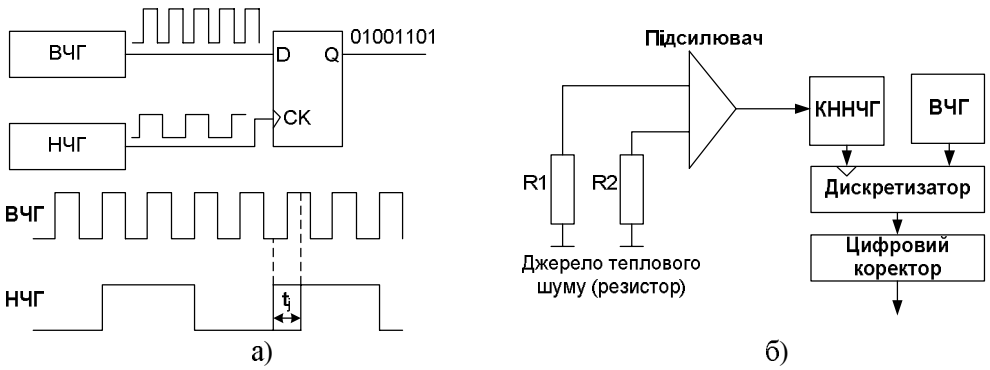

Рис. 1. ГІВЧ на основі незалежних цифрових генераторів (а) та його вдосконалений варіант (б)

Недоліком такого методу є необхідність накопичення джиттеру фази впродовж тривалого часу (оскільки джиттер цифрових генераторів є достатньо малий), щоб отримати якісні випадкові дані, а це обмежує продуктивність ГІВЧ на рівні 1 Мбіт/секунда, що може бути недостатнім для високопродуктивних криптосистем.

Відомі декілька спроб зменшити час накопичення джиттеру. Наприклад, запропонований фірмою Intel ГІВЧ (рис. 1. б) в якості додаткового джерела ентропії використовує два диференційно включених резистори, чий тепловий шум підсилюється і здійснює модуляцію частоти керованого напругою низькочастотного генератора (КННЧГ), сигнал якого служить для дискретизації виходу незалежного високочастотного генератора (ВЧГ). Вихідна послідовність піддається постобробці з використанням коректора фон Неймана та алгоритму хешування SHA-1 [4]. Проте така архітектура не є повністю цифрова, що ускладнює її реалізацію.

Інший підхід до вдосконалення схеми незалежних генераторів полягає в їх використанні для тактування лінійного регістра зсуву зі зворотними зв'язками та скінченного автомата різної розрядності, виходи яких об'єднуються за допомогою операції XOR [5].

Описані ГІВЧ орієнтовані на FPGA, ASIC, SoC, тобто обчислювальні засоби з програмованою внутрішньою структурою і не придатні для реалізації в готовому МК з жорсткою архітектурою.

Враховуючи цю проблему деякі виробники останнім часом стали включати до складу своїх МК апаратні ГІВЧ. Зокрема у високопродуктивних МК сімейства STM32F4xx (фірма STMicroelectronics) побудованих на базі універсального ядра ARM Cortex-M4F з'явився вбудований модуль ГІВЧ, що у поєднанні з апаратною підтримкою основних криптографічних операцій (шифрування – AES/DES/TDES, гешування – MD5/SHA-1/HMAC) робить їх зручною платформою для криптоаплікацій.

При цьому на перший план виходить не менш важлива задача – тестування якості вбудованого генератора, виявлення та усунення недоліків.

Мета статті. Метою даної роботи є дослідити статистичні характеристики вбудованого ГІВЧ мікроконтролерів сімейства STM32F4xx з метою виявлення та усунення можливих недоліків, оцінити продуктивність генератора та можливість використання у вбудованих криптографічних аплікаціях.

Архітектурні особливості вбудованих ГІВЧ мікроконтролерів сімейства STM32F4xx. Модуль ГІВЧ в мікроконтролерах сімейства STM32F4xx побудований на джерелі аналогового шуму і забезпечує генерацію випадкових 32-бітних чисел. Також в ньому передбачені спеціальні кола, які здійснюють онлайн-контроль роботи ГІВЧ та сигналізують про можливі збої - такі як генерація постійних значень або постійної послідовності значень.

Структурна схема апаратного ГІВЧ МК сімейства STM32F4xx представлена на рис. 2 [6].

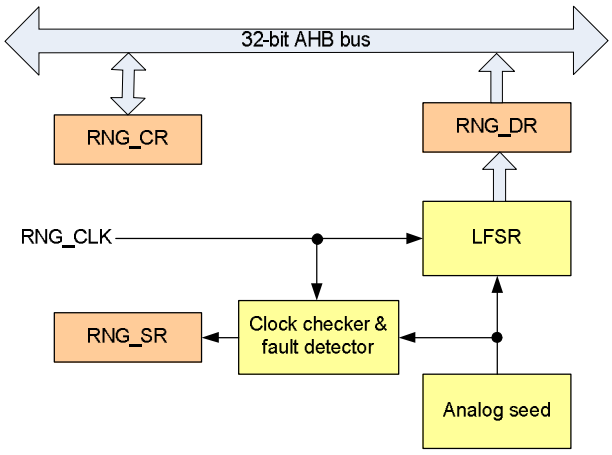

Рис. 2. Структурна схема модуля ГІВЧ в МК сімейства STM32F4xx

Аналогові кола генерують зародок (Analog seed), що поступає на лінійний регістр зсуву зі зворотними зв'язками (LFSR). Аналогові кола побудовані на незалежних генераторах, чиї виходи об'єднуються операцією XOR. Для тактування LFSR використовується окремий

тактовий сигнал (RNG_CLK), який формується спеціальною схемою ФАПЧ, тому якість ГІВЧ не залежить від значення основної тактової частоти МК. Генерація одного 32-бітного значення потребує до 40 тактів сигналу RNG_CLK. Максимальне значення RNG_CLK становить 48 МГц.

Коли 32-бітне випадкове число сформоване – воно пересилається в регістр даних (RNG_DR) та встановлюється відповідний прапорець в регістрі статусу (RNG_SR).

Паралельно здійснюється моніторинг тактового сигналу RNG_CLK та зародку. Регістр статусу містить спеціальні прапорці, які сигналізують про атипову послідовність зародків (прапорець SECS) або про те, що тактова частота є занизькою (прапорець CECS).

За збій приймаються дві ситуації: коли згенеровано 64 і більше послідовних біт з однаковим значенням (0 або 1), або 32 послідовні пари 0 і 1 (0101010101…01).

При виявленні збою ГІВЧ слід перезапустити з допомогою відповідних бітів регістра управління (RNG_CR).

У технічній документації на МК сімейства STM32F4xx відсутня детальна кількісна оцінка якості вбудованого генератора згідно сучасних вимог та стандартів для ГВЧ, а лише зазначено, що він забезпечує у 85% успішне проходження тестів згідно стандарту FIPS 140-2 [7], без вказання результатів по кожному з тестів та об'єму вибірки.

Слід відзначити, що хоча рання версія стандарту FIPS 140-2 і передбачала виконання чотирьох простих тестів (Monobit test, Poker test, Runs test, Long runs test) над випадковими послідовностями довжиною 20000 біт, проте ця вимога в останній діючій версії стандарту відсутня, а сам набір тестів за сучасними міркам є надто обмеженим і примітивним.

Вибір та налаштування апаратних засобів для експериментальних досліджень. Для експериментальних досліджень якості і продуктивності вбудованого ГІВЧ мікроконтролерів сімейства STM32F4xx нами було обрано плату STM32F4DISCOVERY на якій встановлено МК STM32F407VG з 32-бітним ядром ARM Cortex-M4F. Максимальна тактова частота МК становить 168 МГц і формується з допомогою внутрішніх кіл ФАПЧ від зовнішнього кварцового резонатора на 8 МГц. Програмується плата через порт USB відладчиком ST-LINK/V2 розташованим на цій же платі.

Генерація випадкових послідовностей здійснювалася при таких параметрах:

- Тактова частота ядра МК – 168 МГц;
- Тактова частота ГІВЧ (RNG_CLK) – 48 МГц;
- Напруга живлення МК – 3.3 В.

Згенеровані послідовності передавалися в USB-порт ПК з допомогою інтерфейсної плати.

Генерація виконувалася відповідно до вимого стандарту FIPS 140-2, згідно якого перше вироблене 32-бітне випадкове число не використовується, а кожен наступний вироблений 32-бітний блок даних повинен порівнюватися з попереднім і якщо вони співпадають, то це трактується як збій в роботі ГІВЧ. Додатково в процесі формування випадкових послідовностей відстежувалася частота появи збоїв, які виявляються колами ГІВЧ (прапорці SECS та CECS). Ця інформація також передавалася до ПК.

Під час відладки програми та генерації послідовностей нами не було виявлено жодного з вищевказаних збоїв, що свідчить про надійну і стабільну роботу вбудованого ГІВЧ.

Результати тестування криптографічної якості ГІВЧ. Для перевірки якості ГВЧ застосовуються різні набори статистичних тестів, серед яких стандартом де-факто є набір тестів розроблений National Institute of Standards and Technology (NIST) [8]. У порівнянні з іншими відомими наборами тестів тести NIST використовують відкриті, детально специфіковані алгоритми, містять контрольні послідовності для перевірки правильності їх реалізації, а також забезпечують однозначну інтерпретацію результатів тестування.

Набір NIST складається з 15 окремих статистичних тестів, кожен з яких здійснює перевірку бінарної послідовності на одну з можливих ознак відхилення від випадковості. На підставі результатів тестів приймається (або відхиляється) гіпотеза про те, що дана послідовність є випадковою.

Результатом виконання кожного тесту є так зване $P\text{-}value$, яке лежить в діапазоні [0, 1]. Рівень значущості α задає імовірність того, що випадкова послідовність буде сприйнята як невипадкова. Якщо $P\text{-}value \geq \alpha$, то приймається, що послідовність, яка тестується, пройшла перевірку і є випадковою.

Тестування проводилося при рівні значущості $\alpha = 0.01$, який рекомендований в [8]. Враховуючи, що мінімальна кількість випадкових послідовностей m повинна бути обернено пропорційною до α ($m \geq 100$) в даній роботі обрано значення $m = 1000$. Оскільки деякі тести для отримання коректного результату вимагають, щоб розмір випадкової послідовності n був не менше 10^6 біт, нами було прийнято $n = 10^6$. Таким чином, сумарний об'єм вибірки становив 10^9 біт.

Для дослідження статистичних властивостей випадкових послідовностей згенерованих вбудованим ГІВЧ МК нами було створене програмне забезпечення в середовищі MatLab, що реалізує набір тестів NIST, а також виконує протоколювання та інтерпретацію результатів.

У середовищі IAR 6.30 Embedded Workbench for ARM було написано програму для МК STM32F407VG, яка здійснює генерацію вбудованим ГІВЧ 1000 послідовностей, розміром 10^6 біт кожна.

Особливістю тестів №14-15 (Random Excursions та Random Excursions Variant) є те, що їх результат ($P\text{-}value$) буде достовірним лише тоді, коли для даної послідовності певний параметр J, який розраховується за наведеним в [8] алгоритмом, буде мати значення не менше 500 ($J \geq 500$). Якщо $J < 500$, то це не є ознакою дефекту ГІВЧ, а лише вказує на те, що результати тесту можуть бути недійсними. Тому для коректного виконання тестів №14-15 окремо генерувалися і відбиралися послідовності з $J \geq 500$.

До кожної з 1000 послідовностей застосовувався набір тестів з параметрами вказаними в табл. 1 при рівні значущості рівному $\alpha = 0.01$.

Для оцінки якості вбудованого ГІВЧ мікроконтролерів STM32F4xx ми дотримувалися рекомендацій щодо інтерпретації результатів, описаних в [8]. Документ [8] передбачає дві стратегії для ухвалення рішення про те, чи даний ГВЧ пройшов тест на випадковість.

Стратегія №1. Ця стратегія для кожного тесту визначає частку послідовностей $P1$, що пройшли перевірку ($P\text{-}value \geq \alpha$) та порівнює її з нижньою межею довірчого інтервалу, рівною

$$P1_{THR} = (1-\alpha) - \sqrt{\frac{(1-\alpha)\alpha}{m}} = 0.980561.$$

Якщо хоча б для одного з 15 тестів частка $P1$ виходить за нижню межу довірчого інтервалу $\left(P1 < P1_{THR}\right)$, то приймається рішення, що ГІВЧ тест на випадковість не пройшов.

Тести Random Excursions та Random Excursions Variant передбачають обчислення $P\text{-}value$ для різних значень аргументу x, зокрема Random Excursions – для 8 значень $\left(x = -4,...,-1,\ 1,...,4\right)$ та Random Excursions Variant – для 18 значень $\left(x = -9,...,-1,\ 1,...,9\right)$. З метою зменшення обсягу статті у табл. 1 вказані лише мінімальні значення $P1$ та відповідні їм аргументи x отримані в тестах № 14-15.

Як слідує з даних в табл. 1 вбудований ГІВЧ пройшов перевірку на випадковість для всіх 15-ти тестів.

Стратегія №2. Дана стратегія базується на тому, що в якісного ГІВЧ розподіл $P\text{-}value$ для кожного тесту є рівномірним на інтервалі [0, 1]. Для перевірки цієї гіпотези використовується тест χ^2 значень $P\text{-}value$ розбитих на 10 підінтервалів $C1\text{-}C10$ з кроком 0.1. Якщо отримане в результаті перевірки гіпотези значення $P2 < 0.0001$, то приймається рішення, що ГІВЧ тест не пройшов. З метою зменшення обсягу статті у табл. 2 вказані лише мінімальні значення $P2$ та відповідні їм аргументи x отримані в тестах №14-15.

З наведених у табл. 2 результатів видно, що вбудований ГІВЧ пройшов перевірку для всіх 15 тестів.

Результати тестування ГІВЧ згідно стратегії №1 (при $m = 1000$ і $n = 10^6$) Таблиця. 1

№	Статистичний тест	Параметри тесту	Кількість послідовностей, що пройшли тест	*P1*	Висновок
1	Frequency	-	995	0.995	Тест пройдено
2	Block Frequency	*M=128*	994	0.994	Тест пройдено
3	Runs	-	989	0.989	Тест пройдено
4	Longest Run	*M=10000*	986	0.986	Тест пройдено
5	Rank	-	993	0.993	Тест пройдено
6	DFT	-	987	0.987	Тест пройдено
7	Non-Overlapping Template	*M=9, B=110100010*	990	0.990	Тест пройдено
8	Overlapping Template	*M=9*	991	0.991	Тест пройдено
9	Linear Complexity	*M=500*	988	0.988	Тест пройдено
10	Universal	*L=8, Q=2356*	990	0.990	Тест пройдено
11	Serial	*M=16,* $\nabla\psi_m^2$	992	0.992	Тест пройдено
		M=16, $\nabla^2\psi_m^2$	990	0.990	Тест пройдено
12	Approximate Entropy	*M=10*	990	0.990	Тест пройдено
13	Cumulative Sums	*Forward*	992	0.992	Тест пройдено
		Reverse	994	0.994	Тест пройдено
14	Random Excursions	*x =-3*	983	0.983	Тест пройдено
15	Random Excursions Variant	*x =-3*	984	0.984	Тест пройдено

Результати тестування ГІВЧ згідно стратегії №2 (при $m = 1000$ і $n = 10^6$) Таблиця 2

Статистичний тест	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	*P2*	Висновок
Frequency	90	102	111	95	97	93	87	103	116	106	0.556460	Тест пройдено
Block Frequency	93	103	101	101	95	84	125	93	99	106	0.310049	Тест пройдено
Runs	99	92	90	111	102	122	86	98	102	98	0.365253	Тест пройдено
Longest Run	101	86	102	112	86	98	109	106	100	100	0.676615	Тест пройдено
Rank	102	91	93	110	108	82	97	107	113	97	0.476911	Тест пройдено
DFT	136	102	89	90	89	100	114	92	91	97	0.018413	Тест пройдено
Non-Overlapping Template	87	102	96	107	81	122	114	100	106	85	0.080519	Тест пройдено
Overlapping Template	115	109	106	92	109	104	84	88	99	94	0.401199	Тест пройдено
Linear Complexity	110	101	83	93	92	92	116	103	106	104	0.452173	Тест пройдено
Universal	126	101	113	89	97	105	77	89	98	105	0.051942	Тест пройдено
Serial	100	103	96	107	101	88	90	103	114	98	0.809249	Тест пройдено
	113	107	87	95	94	100	107	92	107	98	0.725829	Тест пройдено
Approximate Entropy	97	94	88	113	94	100	111	107	102	94	0.735908	Тест пройдено
Cumulative Sums	94	95	88	97	104	100	103	112	117	90	0.562591	Тест пройдено
	90	107	106	97	94	89	108	103	106	100	0.867692	Тест пройдено
Random Excursions ($x = 4$)	80	114	118	86	103	100	88	121	98	92	0.037813	Тест пройдено
Random Excursions Variant ($x = 3$)	83	92	93	79	89	104	104	101	128	127	0.002863	Тест пройдено

У табл. 3 подано результати тестування ГІВЧ згідно методики наведеної в старій версії стандарту FIPS 140-2, де $P1$ – частка послідовностей, що пройшли тест.

Результати тестування ГІВЧ згідно FIPS 140-2 (при $m = 50000$ і $n = 20000$) Таблиця 3

Monobit	Poker	Runs	Long Runs
$P1 = 0.999840$	$P1 = 0.999900$	$P1 = 0.995980$	$P1 = 0.999820$

Оцінка продуктивності вбудованого ГІВЧ. За результатами експериментів було встановлено, що ГІВЧ забезпечує достатньо високу продуктивність – не менше 34.88 Мбіт/секунду при будь-якому рівні оптимізації компілятора (Low, Medium, High). Враховуючи потенційно великий обсяг генерованих даних, як недолік можна відзначити відсутність у ГІВЧ режиму прямого доступу до пам'яті, що дозволив би розвантажити центральний процесор від операцій з пересилання даних між ГІВЧ і пам'яттю.

Висновки. Отже, результати досліджень вказують на те, що вбудовані ГІВЧ МК сімейства STM32F4xx задовольняють вимоги які ставляться до ГІВЧ в криптографічних аплікаціях. Висока продуктивність та стабільність роботи вбудованого ГІВЧ у поєднанні з наявністю апаратного шифрування і гешування робить МК сімейства STM32F4xx перспективною платформою для широкого спектру криптографічних пристроїв.

ЛІТЕРАТУРА

1. Secure Integrated Circuits and Systems // Ed. Ingrid M.R. Verbauwhede. – Springer-Verlag, 2010. – 246 p. – ISBN 978-0-387-71827-9.

2. Cryptographic Engineering // Ed. Koc C.-K. – New York: Springer Science+Business Media, 2009. – 522 p. – ISBN: 978-0-387-71816-3.

3. Killmann W., Schindler W. A Design for a Physical RNG with Robust Entropy Estimators // Proceedings of the 10th International Workshop on Cryptographic Hardware and Embedded Systems (CHES'08), 2008, Washington, USA, LNCS, Vol. 5154, pp. 146-163, Springer, Heidelberg (2008).

4. Jun, B., Kocher P. The Intel Random Number Generator. Cryptography Research, Inc., White Paper prepared for Intel Corporation, 1999, 8 p.

5. Tkacik T. A Hardware Random Number Generator // Proceedings of the 4th International Workshop on Cryptographic Hardware and Embedded Systems (CHES'02), 2002, Redwood Shores, USA, LNCS, Vol. 2523, pp. 450-453, Springer, Heidelberg (2002).

6. Reference manual. STM32F405xx, STM32F407xx, STM32F415xx and STM32F417xx advanced ARM-based 32-bit MCUs (RM0090) // STMicroelectronics, 2011, 1316 p.

7. FIPS PUB 140-2. Security Requirements for Cryptographic Modules // Federal Information Processing Standards Publication 140-2, 2001, 69 p.

8. NIST SP 800-22rev1a. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications // National Institute of Standards and Technology Special Publication 800-22rev1a, 2010, 131 p.

Надійшла: 21.08.2012р.

Рецензент: д.т.н., проф. Козловський В.В.