УДК 004.056:061.68

Я. Р. Совин, Ю. М. Наконечний, М. Ю. Стахів
Національний університет "Львівська політехніка",
кафедра захисту інформації

ОСОБЛИВОСТІ КРИПТОГРАФІЧНИХ АКСЕЛЕРАТОРІВ У МІКРОКОНТРОЛЕРАХ ЗАГАЛЬНОГО ПРИЗНАЧЕННЯ

Проаналізовано криптоакселератори у 8/16/32-бітових мікроконтролерах загального призначення з погляду швидкодії та функціональних можливостей.

Ключові слова – криптоакселератор, мікроконтролер, шифрування, хеш-функція.

The article analyzes the cryptographic accelerator in 8/16/32-bit general purpose microcontrollers in terms of performance and functionality.

Key words: cryptographic accelerator, microcontroller, encryption, hash.

Вступ

У зв'язку з технічним прогресом проблема захисту інформації, яка обробляється електронними засобами, є доволі актуальною. Серед усього спектра методів захисту даних від небажаного доступу і збереження інформацією своїх основних властивостей особливе місце посідають криптографічні методи. При цьому намагаються досягти конфіденційності та цілісності інформації, що забезпечується такими криптографічними алгоритмами, як шифрування та хешування. До того ж для шифрування потрібні секретні ключі, які виробляють генератори випадкових чисел (ГВЧ).

Сьогодні криптоалгоритми широко використовують в обчислювальних та інформаційно-вимірювальних системах і мережах. Важливою і невід'ємною складовою інформаційних технологій є також вбудовані системи (ВС), де ціна та витрати енергії виходять на перший план, а обчислювальна потужність сконцентрована у недорогих центральних процесорах у складі мікроконтролерів (МК) загального призначення. Такі системи з обмеженими ресурсами знаходять широке застосування під час побудови безпровідних сенсорних мереж, індустріальних, споживчих, медичних, автомобільних та кіберфізичних систем, IoT-пристроїв, інтелектуальних карт та токенів, систем охоронно-пожежної сигналізації, безпеки і контролю доступу, систем промислово-побутової автоматизації і моніторингу, wearable-електроніки (фітнес-трекерів, розумних годинників, окулярів) тощо. Тож потреба у захисті інформації загалом та в криптографічних методах зокрема гостро відчувається у цих системах. Основною проблемою вбудованих систем є те, що надзвичайно важко у готовому пристрої одночасно оптимізувати рівень безпеки, ціну та продуктивність – три основні умови для успішного впровадження проекту.

Історично криптоалгоритми розроблялися для імплементації в інформаційних системах, побудованих на високопродуктивних мікропроцесорах загального призначення. Як результат, існуючі криптоалгоритми доволі погано пристосовані до застосування у вбудованих системах, переважна більшість з яких ґрунтується на 8/16/32-бітових процесорах з малими обчислювальними ресурсами. Операції шифрування і хешування за традиційної програмної реалізації на МК загального призначення є доволі повільними і потребують значних витрат постійної і оперативної пам'яті. Генерація випадкових чисел на мікроконтролері, який є детермінованою системою, теж викликає відчутні складнощі – в результаті відомі ГВЧ є або повільними, або не достатньо випадковими. Відповідно пошук нових алгоритмів та способів реалізації, які б добре працювали на цих платформах, є важливим і актуальним завданням, з яким пов'язаний такий напрямок, як легковагова (lightweight) або малоресурсна криптографія.

Для подолання цих проблем у мікроконтролери загального призначення почали включати спеціалізовані модулі – т. зв. криптоакселератори (криптографічні прискорювачі), які дають змогу значно прискорити (у десятки, сотні і навіть тисячі разів) виконання певних криптоалгоритмів. Криптоакселератори (КА) працюють окремо від ядра і в такий спосіб ядро процесора може фактично не брати участі у криптографічних обчисленнях, зберігаючи свої ресурси для виконання інших завдань. Криптоакселератори на відміну від стандартних інтерфейсів (SPI, I^2C, UART, USB і т.д.) мають специфічні реалізації залежно від моделі і виробника МК, що ускладнює оптимальний вибір для конкретного завдання.

Аналізуючи криптоакселератори у найпоширеніших родинах мікроконтролерів, які використовуються у ВС, можна оцінити швидкодію та виграш у продуктивності порівняно з програмними реалізаціями криптоалгоритму, зробити висновок про зручність і гнучкість роботи з ними, а отже, про можливість і доцільність їх використання для конкретної аплікації.

Аналіз останніх досліджень і публікацій

Навіть високопродуктивні мікропроцесори загального призначення (Intel, AMD) з високими тактовими частотами, великими обсягами оперативної та кеш-пам'яті, потужною системою команд і підтримкою багатопоточності зіткнулися з проблемою недостатньої продуктивності під час реалізації криптоалгоритмів. Для вирішення цієї проблеми виробники почали переміщати криптографічну обробку даних в апаратні блоки своєї продукції – криптоакселератори. Прискорена апаратна криптографічна обробка замість програмного виконання цих самих алгоритмів дає змогу істотно розвантажити центральний процесор.

Прикладом такого підходу є представлені у табл. 1 розширення системи команд x86 командами виконання криптоалгоритму AES – Advanced Encryption Standard New Instructions (AES-NI) з метою прискорення додатків, що використовують AES-шифрування [1].

Таблиця 1

Інструкції AES-NI

Інструкція	Призначення
AESENC	Виконати один раунд зашифрування AES
AESENCLAST	Виконати останній раунд зашифрування AES
AESDEC	Виконати один раунд розшифрування AES
AESDECLAST	Виконати останній раунд розшифрування AES
AESKEYGENASSIST	Посприяти генерації раундового ключа AES
AESIMC	Формування ключів для режиму Equivalent Inverse Cipher

За допомогою цих команд можна пришвидшити виконання алгоритму AES приблизно у 7–10 разів, зокрема за даними [2], для мікропроцесора Pentium 4 швидкодія зростає з 28,0 тактів/байт (без AES-NI) до 3,5 тактів/байт (з AES-NI).

Універсальніший підхід, який дасть змогу апаратно пришвидшувати фактично будь-який криптоалгоритм, – це нещодавно анонсований компанією Intel намір вбудовувати ПЛІС у свої процесори, що дасть змогу для багатьох завдань у 10–20 разів підвищити швидкість обчислень.

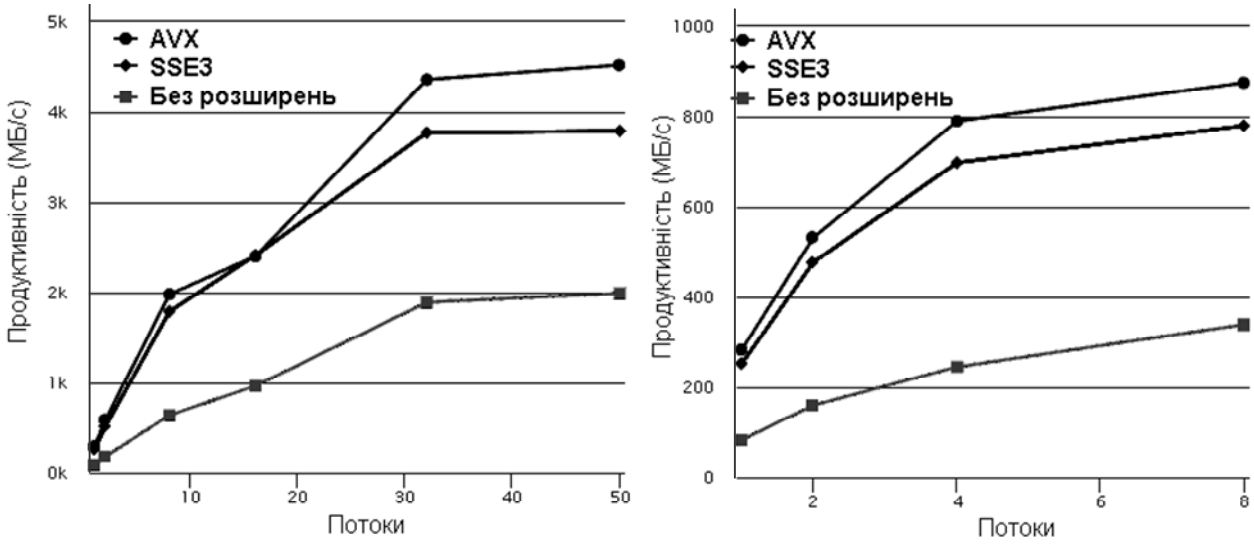
Ще один спосіб пришвидшення криптографічних операцій завдяки паралельним обчисленням – використання векторних інструкцій, що уможливлюють виконувати кілька операцій за один такт процесора. Можливість виконання векторних операцій в обчислювальних системах на платформах x86/x64 забезпечується спеціальними процесорними розширеннями SSE і AVX [3].

SSE (Streaming SIMD Extensions) – це набір SIMD-інструкцій, розроблений Intel і вперше представлений у процесорах серії Pentium III. SSE додає в архітектуру процесора вісім (шістнадцять для x86-64) 128 бітових регістрів *XMM0-XMM7* (*XMM0-XMM15*), кожен з яких може містити чотири 32-бітові значення, що обробляються паралельно за допомогою SSE-інструкцій.

AVX (Advanced Vector Extensions) – розширення системи команд x86 для мікропроцесорів Intel і AMD як подальший розвиток і вдосконалення SSE. Ширина векторних регістрів SIMD

збільшена зі 128 до 256 біт (регістри *YMM0-YMM15*), а для роботи з *YMM*-регістрами додані нові 256-бітові AVX-інструкції.

У [4] показано, як можна застосувати ці розширення для ефективної реалізації алгоритму блокового симетричного шифрування (БСШ) (ГОСТ 28147-89) на серверних і користувацьких ПК, що дало збільшення продуктивності у 2–2,5 раза.



а *б*

Рис. 1. Шифрування ГОСТ 28147-89 у режимі гамування для Intel Xeon E5-2680 (а) та Intel Core i7 (б) [4]

Також як криптоакселератори можуть використовуватись графічні процесори GPU (Graphics Processing Units). Перевагами використання графічних процесорів для реалізації криптографічних алгоритмів є: більша кількість ядер (від кількох сотень до кількох тисяч), ефективніше використання площі кристала та вища у кілька разів пропускна здатність. Особливо широке застосування GPU отримали для обчислень функції хешування у протоколі криптовалюти BitCoin.

У мікропроцесорах фірми Intel також присутній ще один криптоакселератор: модуль Intel Secure Key – це умовна назва для нових інструкцій *RDRAND* та *RDSEED* і вбудованого в процесор апаратного генератора випадкових чисел, який їх реалізує [5]. Intel називає його “цифровий генератор випадкових чисел” (Digital Random Number Generator, DRNG).

DRNG можна розбити на три логічні рівні:

1. Джерело ентропії (ES), яке продукує випадкові біти з недетермінованого апаратного процесу з використанням теплового шуму у напівпровідниках і передає їх блоку підготовки.

2. Блок підготовки, що на основі двох 256-бітних значень від ES генерує одне 256-бітне за алгоритмом AES-CBC-MAC. Цим досягається маскування потенційних статистичних дефектів. Згенероване 256-бітне значення використовується як зародок на наступному рівні для ініціалізації генератора псевдовипадкових чисел DRBG.

3. Deterministic Random Bit Generator (DRBG). Генерує випадкові дані великого об’єму з високою швидкодією (до 6 Гбіт/сек), використовуючи стандартний алгоритм CTR_DRBG на основі AES. Дані поступають у буфер, з якого зчитуються інструкціями *RDRAND*.

4. Enhanced Nondeterministic Random Number Generator. Розширений недетермінований ГВЧ призначений для того, щоб зробити доступним згенеровані в блоці підготовки зародки для використання в інших програмних засобах. Дані поступають у буфер, з якого зчитуються інструкціями *RDSEED*.

У вбудованих системах криптоалгоритми довго реалізувалися тільки програмно, і лише недавно масово почали з’являтися інтегровані у мікроконтролери криптоакселератори, які будуть розглянуті у роботі. Використання криптоакселераторів дає такі переваги: вищу швидкодію і

енергоефективність, розвантаження центрального процесора, економію пам'яті, більшу стійкість до диференційних атак на енергоспоживання (Differential Power Analysis).

Щоб мати базу для порівняння з апаратними прискорювачами, проаналізуємо програмні реалізації криптоалгоритмів у ВС.

Традиційно у ВС домінують симетричні блокові шифри, з яких найчастіше використовується алгоритм AES, з розміром блока 128 біт та довжинами ключа 128-, 192- або 256-біт.

Оцінку швидкодії та вимоги до пам'яті найпоширеніших БСШ для різних мікроконтролерних архітектур розглядають численні дослідження, результати деяких з них подані у табл. 2. У цих роботах розглядаються впливи як архітектури, так і різних шляхів оптимізації на рівні алгоритму і компілятора, на продуктивність та обсяг необхідної пам'яті.

Таблиця 2

Параметри програмних реалізацій БСШ

Криптоалгоритм	Процесор	Зашифрування, тактів/блок	Розшифрування, тактів/блок	Розмір ROM (Flash), байт
DES [6]	AVR (8 біт)	8633	8154	4314
AES-128 [6]	AVR (8 біт)	6637	7429	2606
AES-128 [7]	AVR (8 біт)	4009	6073	3100
AES-128 [8]	C8051F326/7 (8 біт)	11053	34634	3981
AES-192 [8]	C8051F326/7 (8 біт)	12955	41590	3989
AES-256 [8]	C8051F326/7 (8 біт)	14857	48609	3997
AES-128 [9]	i8051 (8 біт)	3905	5876	2079
AES-256 [9]	i8051 (8 біт)	5372	8132	2412
AES-128 [7]	MSP430 (16 біт)	5432	8802	2536
AES-128 [10]	MSP430 (16 біт)	3564	4277	5160
DES [11]	MSP430 (16 біт)	2700	2700	5734
AES-128 [11]	MSP430 (16 біт)	7900	7900	1587
AES-128 [12]	PIC24 (16 біт)	2808	4490	3018
TDES [12]	PIC24 (16 біт)	13557	13557	7500
AES-128 [13]	ARM7TDMI (32 біти)	639	638	5966
AES-128 [14]	ARM Cortex-M3 (32 біти)	1388	1697	1898
AES-256 [14]	ARM Cortex-M3 (32 біти)	1956	2401	1898
AES-128 [15]	ARM Cortex-M3 (32 біти)	3509	5014	1496

Стосовно програмних реалізацій хеш-функцій відзначимо, що використання 8-бітових мікроконтролерів вимагає додаткових операцій завантаження і збереження даних у пам'яті. Це пов'язано з тим, що стан хеш-функції не поміщається у регістрах разом з іншими потрібними даними. Застосування 32-бітових МК дає змогу доволі істотно скоротити час обчислення хеш-функції за рахунок зменшення кількості операцій завантаження і збереження даних.

Для 8-бітових мікроконтролерів AVR у [16] подано результати програмної реалізації функцій хешування SHA-1 та SHA-256. Використання SHA-1 потребує 177 тактів/байт, 122 байти оперативної пам'яті та 1352 байти постійної пам'яті. Для SHA-256 потрібно 335 тактів/байт, 158 байтів оперативної пам'яті та 2720 байтів постійної.

Інші результати реалізації SHA-256 для мікроконтролерів ATtiny45 наведено у [17]. Для функції хешування потрібно 532 тактів/байт 143 оперативної та 1090 байтів постійної пам'яті.

В [11] для мікроконтролерів MSP430 під час реалізації алгоритму SHA-256 досягнуто такі результати: за оптимізації за швидкодією хешування одного блока займає 34100 тактів на блок (533 тактів/байт), вимагаючи 4080 байтів пам'яті програм, а за оптимізації за розміром коду – 44300 тактів (692 тактів/байт) та 2251 байт ПЗП.

Мета роботи – проаналізувати криптоакселератори у найпоширеніших 8/16/32-бітових родинах мікроконтролерів з погляду швидкодії та гнучкості роботи, що дасть змогу обґрунтовано вибирати оптимальне рішення під час розроблення механізмів захисту у вбудованих системах.

Криптоакселератори у 8-бітових мікроконтролерах

AVR. Усі мікроконтролери AVR родини XMega фірми Atmel оснащені криптоакселераторами блокових симетричних шифрів DES і AES [18].

Зокрема у системі команд мікроконтролерів XMega передбачена інструкція *DES K*, яка відповідає одному (*K*-му) з 16-ти раундів алгоритму DES. Як вхідні дані для команди використовується 64-бітовий блок даних, розташований у регістрах загального призначення (РЗП) *R0-R7* та 64-бітовий ключ, розміщений у РЗП *R8-R15*. Прапорець *H* регістру стану *SREG* задає тип операції: *H* = 0 – зашифрування, *H* = 1 – розшифрування (рис. 2, *а*). Режими роботи DES, які підтримуються, – лише ECB.

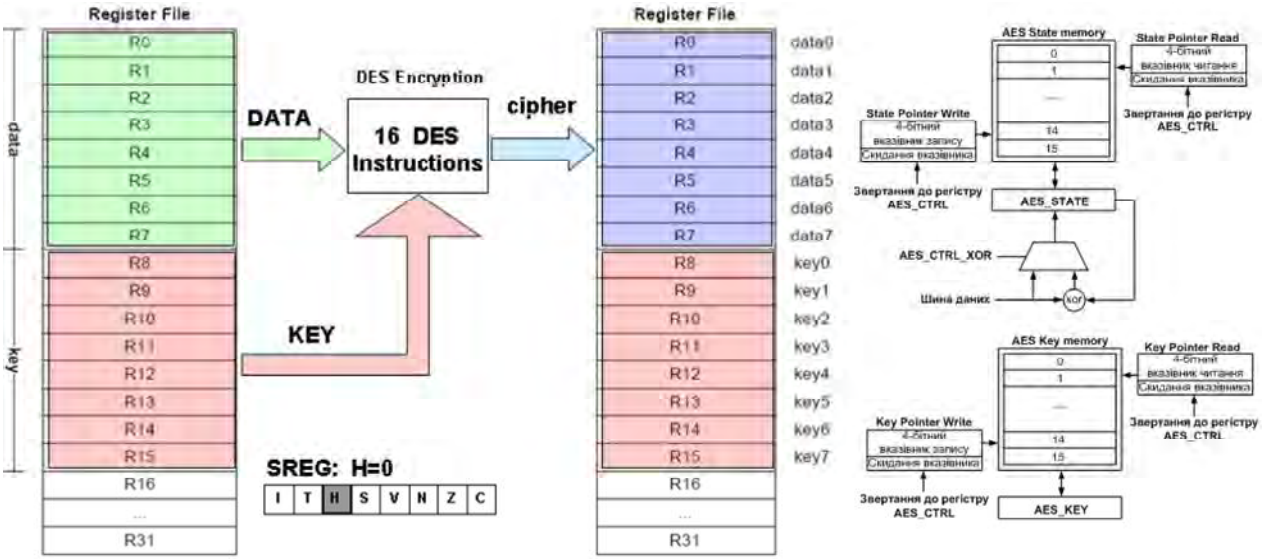


Рис. 2. Виконання операції шифрування алгоритмом DES (а) та структура криптомодуля AES (б) у мікроконтролерах родини XMega

Крім підтримки алгоритму DES, на рівні системи команд у мікроконтролерах родини XMega реалізована апаратна підтримка алгоритму AES за допомогою криптомодуля AES (рис. 2, *б*).

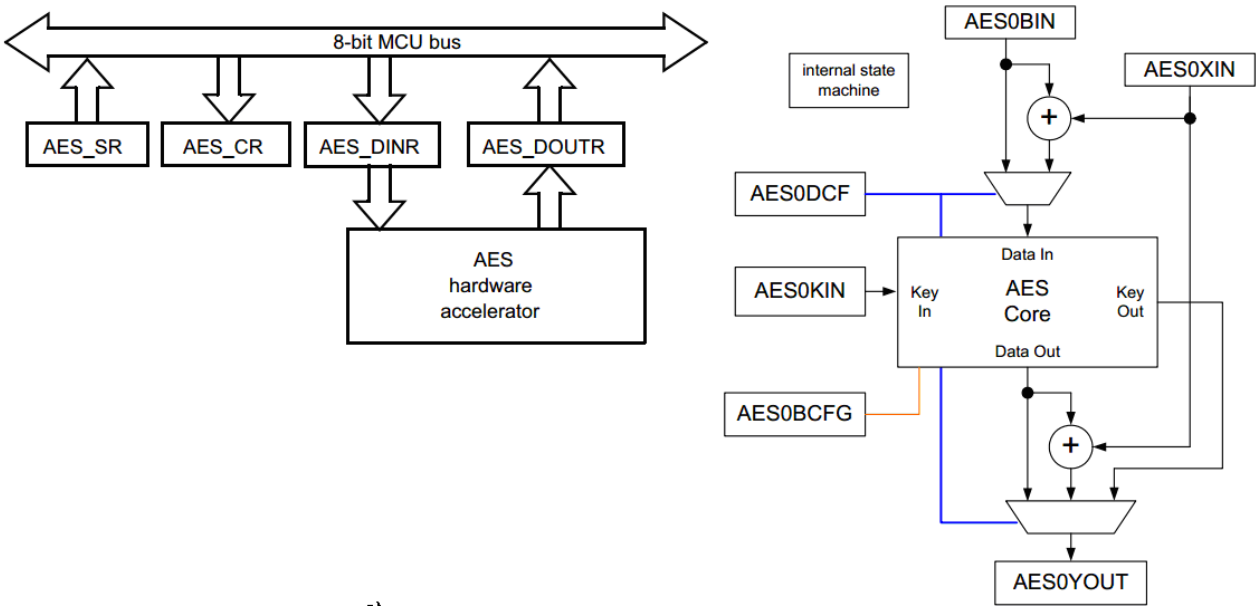
Криптомодуль AES є периферійним модулем мікроконтролера, який шифрує дані блоками по 128 бітів за допомогою 128-бітового ключа. Відповідно криптомодуль AES має пам'ять для зберігання 128-бітового блока даних (*AES State Memory*) та 128-бітового ключа (*AES Key Memory*). Доступ до цих областей пам'яті здійснюється через регістри вводу-виводу *AES_State* та *AES_Key*. Керування та взаємодія з модулем здійснюється через регістр управління *CTRL* та регістр статусу *STATUS*.

Режими роботи алгоритму AES, які підтримуються ECB, CBC. Наявність DMA-контролера прямого доступу у пам'ять (Direct Memory Access, DMA) дає змогу виконувати пересилання вхідних і вихідних даних без втручання центрального процесора.

STM8. У мікроконтролерах родин STM8AL31Exx, STM8AL3LExx та STM8L16xx присутній криптоакселератор алгоритму AES-128 (рис. 3, *а*). Він здатний шифрувати 128-бітові блоки даних, використовуючи 128-бітовий ключ, а також породжувати ключ для розшифрування. Безпосередньо підтримуються тільки режим ECB. КА забезпечує DMA-передачі як для вхідних, так і вихідних даних, що розвантажує центральний процесор від операцій пересилки [19].

Криптоакселератор підтримує чотири режими операцій: зашифрування, породження ключа розшифрування, розшифрування з попередньо обчисленим ключем, породження ключа + розшифрування з використанням ключа шифрування. Режими операцій задаються бітами регістра управління *AES_CR* (AES Control Register).

Відкритий текст, шифртекст або ключ записуються у регістр *AES_DINR* (AES Data Input Register). Після завершення обчислень встановлюється відповідний прапорець у регістрі статусу *AES_SR* (AES Status Register) і може генеруватися переривання. Зчитуються дані з регістра *AES_DOUTR* (AES Data Output Register).



а *б*

Рис. 3. Блок-схеми криптоакселераторів AES мікроконтролерів STM8 (а) та C8051F96x з ядром i8051 (б)

i8051. Мікроконтролери родини C8051F96x фірми Silicon Labs з ядром i8051 містять криптоакселератор алгоритму AES з підтримкою ключів завдовжки 128, 192 та 256 бітів і безпосередньо можуть працювати у режимах ECB, CBC, CTR [20].

Криптоакселератор складається з таких елементів (рис. 3, *б*):

ядра – виконує зашифрування, розшифрування і породження ключа розшифрування:

конфігуруючих регістрів – задають довжину ключа, початок перетворення та маршрут проходження даних;

регістрів ключа, вхідних і вихідних даних;

вхідного і вихідного мультиплексорів з блоками виконання операції XOR;

внутрішнього кінцевого автомата.

Криптоакселератор підтримує DMA-передачі як для вхідних, так і для вихідних даних.

Таблиця 3

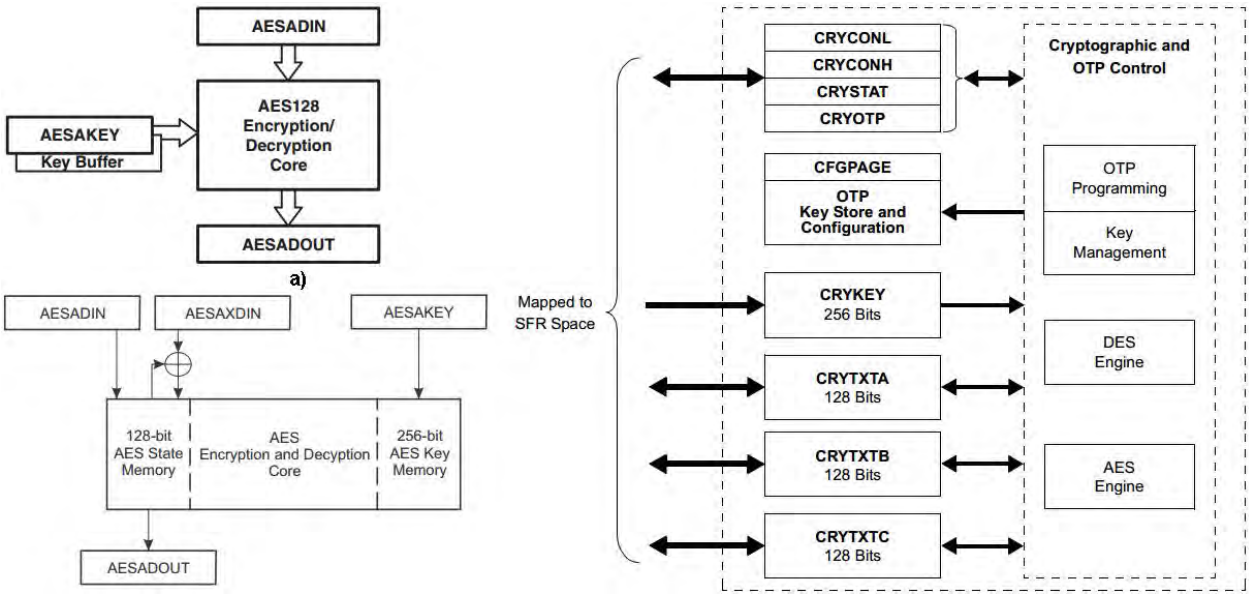
Характеристики криптоакселераторів 8-бітових МК

Операція	Архітектура	Родина МК	F_{CPU}, МГц	Режими	Тактів/ блок
Enc./Dec. DES	AVR	XMega	32	ECB	17
Enc./Dec. AES-128				ECB, CBC	375
Enc./Dec. AES-128	STM8	STM8AL31E, STM8AL3LE, STM8L16	16	ECB	892
Dec. + Key AES-128					1228
Enc./Dec. AES-128	i8051	C8051F96x	25	ECB, CBC, CTR	218
Enc./Dec. AES-192					274
Enc./Dec. AES-256					298

Криптоакселератори у 16-бітових мікроконтролерах

MSP430. В окремих моделях мікроконтролерів родини MSP430F6xx присутній криптоакселератор алгоритму AES-128 (рис. 4, *а*) [21], а у родинах МК MSP430FR59xx/FR69xx з FRAM-пам'яттю – AES-128/192/256 [22]. КА виконують розширення ключа на льоту під час зашифрування і розшифрування та off-line генерацію ключа для розшифрування. Надається гнучкий побайтовий та послівний доступ до ключа вхідних і вихідних даних.

Вхідні дані записуються у регістр *AESDIN*, вихідні дані зчитуються з регістра *AESDOUT*, ключ заноситься у регістр *AESAKEY* (рис. 4, *б*). Криптоакселератори підтримують DMA-передачі та режими ECB, CBC, OFB, CFB.



а *б*

Рис. 4. Блок-схеми криптоакселераторів AES у МК MSP430F6xx (а), MSP430FR59xx/69xx (б) та PIC24 (в)

PIC24. У мікроконтролерах родини PIC24FJ64/128 з ядром PIC24 присутній криптомодуль (Cryptographic Engine) з широкими функціональними можливостями. Критоакселератори, які входять у цей модуль, підтримують шифрування алгоритмами DES і TDES (з довжиною ключа 56, 112 і 168 бітів) та AES (з довжиною ключа 128, 192 і 256 бітів) у режимах ECB, CBC, CFB, OFB і CTR. У модулі передбачені кола захисту від зламу і проникнення у мікросхему [23].

Також модуль містить 512 бітів OTP-пам'яті (One-Time Programmed), яка недоступна для зчитування з інших ділянок пам'яті і слугує для захищеного зберігання секретних ключів (рис. 4, *в*).

Таблиця 4

Характеристика криптоакселераторів 16-бітових МК

Операція	Архітектура	Родина МК	FCPU, МГц	Режими	Тактів/ блок
Enc./Dec. AES-128	MSP430	MSP430F6xx	25	ECB, CBC, OFB, CFB	167
Dec. + Key AES-128					214
Enc./Dec. AES-128		MSP430FR59xx, MSP430FR69xx	16	ECB, CBC, OFB, CFB	168
Dec. + Key AES-128					215
Enc./Dec. AES-192					204
Dec. + Key AES-192					255
Enc./Dec. AES-256					234
Dec. + Key AES-256					292
Enc./Dec. DES	PIC24	PIC24FJ64, PIC24FJ128	16	ECB, CBC, OFB, CFB, CTR	10
Enc./Dec. TDES					26
Enc./Dec. AES-128					219
Enc./Dec. AES-192					275
Enc./Dec. AES-256					299

Криптоакселератори у 32-бітових мікроконтролерах

ARM7TDMI. До складу мікроконтролерів родини AT91SAM7XC з ядром ARM7TDMI входять криптоакселератори алгоритмів AES і DES/TDES. Обидва криптоакселератори підтримують режими шифрування ECB, CBC, CFB, OFB, CTR (лише для AES). Наявний блок DMA для пересилання вхідних і вихідних даних без втручання процесора [24].

У мікроконтролерах передбачені апаратні заходи з протидії диференціальним атакам аналізу енергоспоживання, хоча виробник не розкриває інформацію про деталі.

ARM Cortex-M. Мікроконтролери родини SAM4E фірми Atmel з ядром ARM Cortex-M4 мають криптоакселератор алгоритму AES з підтримкою 128/192/256-бітових ключів. КА може працювати у режимі ECB, CBC, OFB, CFB та CTR [25].

Мікроконтролери родини SAME70 фірми Atmel з ядром ARM Cortex-M7 оснащені криптоакселераторами шифрування AES (128/192/256-бітові ключі), хешування SHA-1, SHA-224, SHA-256 та генератором справжніх випадкових чисел (True Random Number Generator, TRNG). TRNG відповідає тестам, спеціально розробленим для криптографічних ГВЧ NIST SP 800-22. Кола онлайн-перевірки якості генерації та виявлення збоїв у TRNG відсутні [26].

Криптоакселератором AES підтримуються режими ECB, CBC, OFB, CFB, CTR та GCM.

Мікроконтролери родини SAML21 фірми Atmel з ядром ARM Cortex-M0+ мають криптоакселератор алгоритму AES для 128/192/256-бітових ключів і TRNG [27]. Підтримуються режими ECB, CBC, OFB, CFB та CTR. TRNG аналогічний до TRNG у SAME70.

Також у мікроконтролерах SAME70 і SAML21 передбачені апаратні заходи з протидії диференціальним атакам аналізу енергоспоживання, які можуть включати:

• випадкове додавання одного такту у процесі обробки даних;

• додавання випадкової кількості тактів (максимум 11/13/15, відповідно до довжини ключа 128/192/256-біт) у процесі обробки даних;

• додавання випадкового енергоспоживання у процесі обробки даних.

Ці заходи можуть програмно включатися або відключатися. Додавання випадкових тактів призводить до зменшення продуктивності.

Мікроконтролери родини LPC18S5x/S3x фірми NXP Semiconductors з ядром ARM-Cortex M3 мають криптоакселератор алгоритму AES з підтримкою 128-бітових ключів та режимів ECB або CBC [28].

Мікроконтролери родини MSP432P4xx фірми Texas Instruments з ядром ARM Cortex-M4 використовують криптоакселератор алгоритму AES з 128/192/256-бітовими ключами [29]. Безпосередньо підтримуються режими ECB, CBC, OFB, CFB.

Мікроконтролери родини SiM3U1xx/SiM3C1xx фірми Silicon Laboratories з ядром ARM Cortex-M3 мають криптоакселератор алгоритму AES з підтримкою 128/192/256-бітових ключів [30]. Вони можуть працювати у режимах ECB, CBC, CTR (рис. 5, *а*).

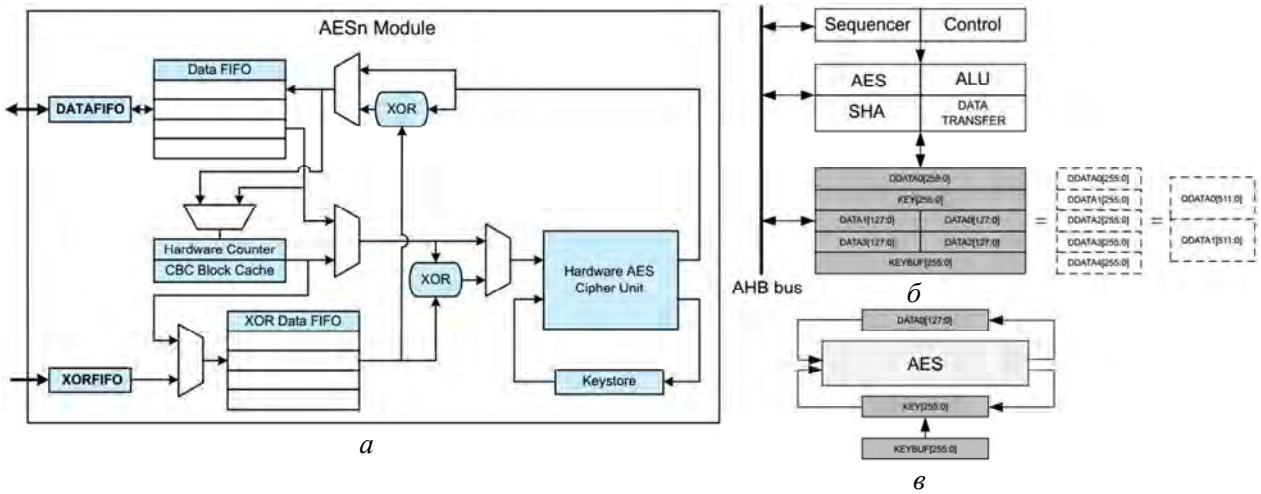


Рис. 5. Блок-схеми криптоакселераторів у МК SiM3U1xx/SiM3C1xx (а), EFM32PG1 (б) та блока AES у EFM32PG1 (в)

Мікроконтролери родини EFM32PG1 з ядром ARM Cortex-M4 фірми Silicon Laboratories мають доволі потужний криптомодуль CRYPTO, який дає змогу реалізувати більшість стандартних криптографічних операцій [31]. Операції у CRYPTO виконуються в окремому АЛП згідно з заданими послідовностями спеціалізованих інструкцій (занесених у Sequencer) над 128/256/512-бітовими регістрами (рис. 5, *б*). Усього є п'ять 256-бітових регістрів. Команди CRYPTO включають базові АЛП-інструкції (арифметичні та логічні *ADD, SUB, MUL, SHIFT, XOR*, модульні *MADD, MMUL, MSUB* та ін.), пересилки даних, умовні та спеціальні інструкції (*AESENC, AESDEC, SHA* тощо).

До складу модуля CRYPTO входить КА AES з підтримкою ключів завдовжки 128/256 бітів та буфер для їх зберігання (рис. 5, *в*). Є можливість працювати у таких режимах роботи, як ECB, CBC, PCBC, CFB, CTR, CBC-MAC, GMAC, CCM, GCM.

Також у складі CRYPTO присутній КА хешування за алгоритмами SHA-1, SHA-224 та SHA-256.

Криптомодуль CRYPTO може використовуватися і як КА для еліптичної криптографії з підтримкою бінарних *GF(2^m)* та простих полів *GF(p)*. Передбачена підтримка таких рекомендованих NIST еліптичних кривих: P-192, P-224, P-256, K-163, K-233, B-163 та B-233.

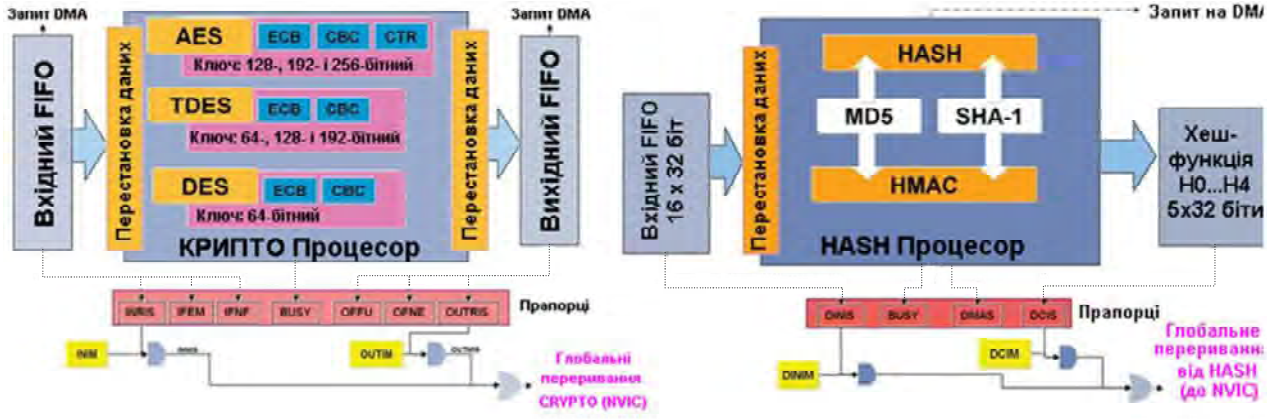
Компанія STMicroelectronics додала у свої мікроконтролери STM32F2xx/F4xx/F7xx з 32-бітовими ядрами ARM Cortex-M3/M4F/M7, відповідно, криптопроцесор, який дає змогу шифрувати дані, обчислювати хеш-повідомлення і генерувати випадкові числа. Криптографічний процесор складається з ядра, що реалізує алгоритми, буферів вхідних/вихідних даних, регістрів зберігання ключів/векторів ініціалізації, регістрів стану і регістрів управління.

До ядра криптопроцесора належить [32, 33]:

1. КА шифрування CRYP, який реалізує на апаратному рівні алгоритми DES/TDES/AES.
2. КА обчислення хеш-функцій HASH, що дає змогу обчислювати хеш-функції за алгоритмами MD5/SHA-1/SHA-224/SHA-256 і коди автентифікації повідомлень HMAC.
3. Генератор випадкових чисел RNG, що дає можливість на основі аналогових генераторів шуму отримувати 32-розрядні випадкові числа.

Криптоакселератор шифрування (CRYP). КА CRYP призначений для зашифрування/розшифрування даних у режимі ECB або CBC для алгоритмів DES, Triple-DES і додатково у режимі CTR для алгоритму AES (рис. 6, *а*). Для алгоритму DES довжина ключа становить 64 біти, для TDES – 64, 128 або 192 біти, для AES – 128, 192 або 256 бітів.

Модуль CRYP забезпечує автоматичний контроль потоку даних з підтримкою прямого доступу до пам'яті (використовуються два канали: один – для прийому вхідних даних, інший – для видачі оброблених даних), має вхідний і вихідний буфери FIFO, завбільшки вісім слів кожен, які відповідають чотирьом блокам DES/TDES або двом блокам AES.



а *б*

Рис. 6. Блок-схеми криптоакселераторів CRYP (а) та HASH (б) мікроконтролерів STM32F2xx/F4xx/F7xx

Хеш-процесор (HASH). HASH-процесор являє собою КА з реалізацією алгоритмів хешування SHA-1/SHA-224/SHA-256, MD5 і HMAC (Keyed-Hash Message Authentication Code). Він обчислює хеш-функцію (160 бітів для SHA-1; 128 бітів для MD5) для повідомлень завдовжки до 2^{64} - 1 біт

(рис. 6, *б*). Алгоритм HMAC надає спосіб підтвердження автентичності повідомлень за допомогою обчислення однієї з хеш-функцій з використанням вибраного користувачем ключа.

HASH-процесор автоматично переставляє вхідні рядки і доповнює вхідний бітовий рядок до довжини, кратної довжині блока.

Час обробки останнього блока повідомлення (або ключа у режимі HMAC) може бути тривалішим, оскільки включає операцію доповнення блока.

Генератор випадкових чисел (Random Number Generator, RNG). Модуль RNG є генератором випадкових чисел, що ґрунтується на безперервному аналоговому шумі (рис. 7), який проходить тести NIST SP 800-22 [34].

Аналогові кола генерують зародок (*Analog seed*), що поступає на лінійний регістр зсуву із 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.

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

Паралельно здійснюється моніторинг тактового сигналу *RNG_CLK* та ентропії зародка. Регістр статусу містить спеціальні прапорці, які сигналізують про атипову послідовність зародків або про те, що тактова частота є занизькою. За збій приймаються дві ситуації: коли згенеровано 64 і більше послідовних бітів з однаковим значенням (0 або 1), або 32 послідовні пари 0 і 1 (01010101…01). Під час виявлення збою ГВЧ потрібно перезапустити за допомогою відповідних бітів регістра управління (*RNG_CR*).

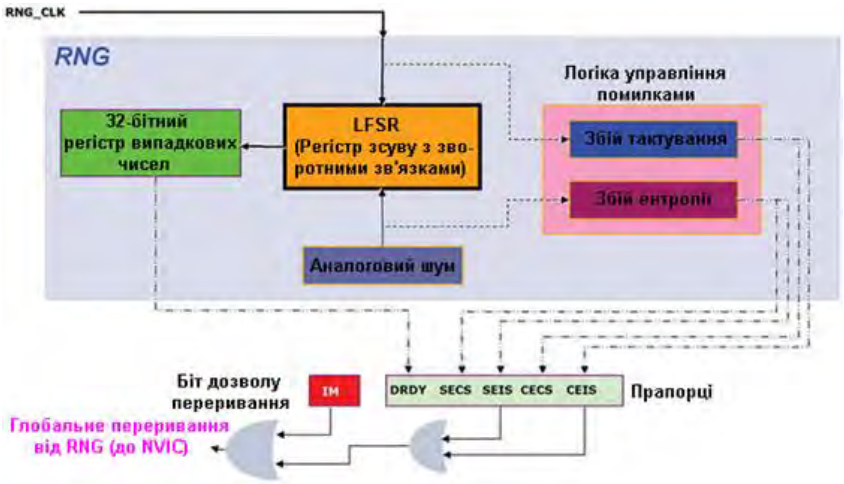


Рис. 7. Блок-схеми RNG мікроконтролерів STM32F2xx/F4xx/F7xx

В усіх розглянутих криптоакселераторах мікроконтролерів з ядром ARM Cortex-M є можливість DMA-пересилань даних.

Таблиця 5

Характеристики криптоакселераторів 32-бітових МК

Операція	Архітектура	Родина МК	FCPU, МГц	Режими	Тактів/ блок
1	2	3	4	5	6
Enc./Dec. DES	ARM7TDMI	AT91SAM7XC	55	ECB, CBC, OFB, CFB	18
Enc./Dec. TDES					50
Enc./Dec. AES-128				ECB, CBC, OFB, CFB, CTR	12
Enc./Dec. AES-192					13
Enc./Dec. AES-256					14

Продовження табл. 5

1	2	3	4	5	6
Enc./Dec. AES-128	ARM Cortex-M4	SAM4E	120	ECB, CBC, OFB, CFB, CTR	12
Enc./Dec. AES-192					14
Enc./Dec. AES-256					16
Enc./Dec. AES-128	ARM Cortex-M7	SAME70	300	ECB, CBC, OFB, CFB, CTR, GCM	12
Enc./Dec. AES-192					14
Enc./Dec. AES-256					16
Hash SHA-1					85
Hash SHA-224					72
Hash SHA-256					72
TRNG-32					84
Enc./Dec. AES-128	ARM Cortex-M0+	SAML21	48	ECB, CBC, OFB, CFB, CTR	57
Enc./Dec. AES-192					67
Enc./Dec. AES-256					77
TRNG-32					84
Enc./Dec. AES-128	ARM Cortex-M3	LPC18S3x, LPC18S5x	180	ECB, CBC	8
Enc./Dec. AES-128	ARM Cortex-M4	MSP432P4xx	48	ECB, CBC, OFB, CFB	168
Dec. + Key AES-128					215
Enc./Dec. AES-192					204
Dec. + Key AES-192					255
Enc./Dec. AES-256					234
Dec. + Key AES-256					292
Enc./Dec. AES-128	ARM Cortex-M3	SiM3U1xx, SiM3C1xx	80	ECB, CBC, CTR	54
Enc./Dec. AES-192					65
Enc./Dec. AES-256					75
Enc./Dec. AES-128	ARM Cortex-M4	EFM32PG1	40	ECB, CTR, CBC, CFB, CBC-MAC, GMAC, CCM, GCM	54
Enc./Dec. AES-256					75
Enc./Dec. DES	ARM Cortex-M3	STM32F2xx	120	ECB, CBC	16
Enc./Dec. TDES					48
Enc./Dec. AES-128				ECB, CBC, CTR	14
Enc./Dec. AES-192					16
Enc./Dec. AES-256	ARM Cortex-M4F	STM32F4xx	168		18
Hash MD5					50
Hash SHA-1					66
Hash SHA-224					50
Hash SHA-256	ARM Cortex-M7F	STM32F7xx	216		50
TRNG-32					40

Висновки

Використання криптоакселераторів дає змогу підняти швидкодію у 10–20 разів для 8/16-бітових МК та до 100 разів – для 32-бітових МК, порівняно з програмними реалізаціями алгоритмів шифрування.

Продуктивність криптоакселераторів для більшості архітектур є достатньою для практичних цілей, а у разі 32-бітових ядер отримана продуктивність на рівні сотень Мбайт/сек є навіть надлишковою, оскільки ці мікроконтролери, як правило, не мають таких високошвидкісних інтерфейсів аналогового чи цифрового вводу-виводу даних, щоб забезпечити неперервний потік. Максимально продуктивними для шифрування алгоритмом AES є мікроконтролери SAME70 фірми Atmel з ядром ARM-Cortex M7 і тактовою частотою 300 МГц, вони здатні забезпечити до 400 Мбайт/сек.

Виробники МК усе ще приділяють недостатню увагу захисту від атак – аналізу енергоспоживання, що дуже характерно і небезпечно для ВС.

Поданий у роботі опис характеристик криптоакселераторів повинен допомогти розібратися з програмуванням прикладних задач із захисту інформації для розглянутих мікроконтролерів.

1. Shay Gueron. Intel Advanced Encryption Standard (AES) New Instructions Set // Intel White Paper. – 2012. – 94 p. 2. "Crypto++ 5.6.0 Pentium 4 Benchmarks". // Crypto++ Website. – 2009. 3. Intel Architecture Instruction Set Extensions Programming Reference // Intel, 1180 p. – 2016. 4. ГОСТ 28147-89: "Не спеши его хоронить". – Ч. 2: Эффективные реализации алгоритма // С. В. Смышляев, Е. К. Алексеев, А. С. Прохоров. – 2015. 5. Intel Digital Random Number Generator (DRNG). Software Implementation Guide // Intel. – 2014. – 35 p. 6. Rinne S., Eisenbarth T., Paar C. Performance Analysis of Contemporary Light-Weight Block Ciphers on 8-bit Microcontrollers // ECRYPT Workshop Software Performance Enhancement for Encryption and Decryption, 2007. – P. 33–43. 7. Feldhofer M. Efficient Data Protection for Devices with Scarce Resources // Workshop Developing Secure Applications for Mobile Wireless Environments, 2010. 8. Advanced Encryption Standard // Silicon Labs, AN324. – 2007. – № 6. – P. 9. Performance-optimized AES Implementation for 8051-based Microcontrollers // Institute for Applied Information Processing and Communication, 2011. 10. Didla S., Ault A., Bagchi S. Optimizing AES for embedded devices and wireless sensor networks // Proceedings of the 4th International Conference on Testbeds and research infrastructures for the development of networks & communities. – 2008. – P. 1–10. 11. J. Hall. C Implementation of Cryptographic Algorithms // Texas Instruments, SLAA547A. – 2013. – 28 pp. 12. D. Flowers, H. Schlunder. Data Encryption Routines for PIC24 and dsPIC Devices // Microchip Technology, AN1044, 2006. – 18 p. 13. Atasu K., Breveglieri L., Macchetti M. Efficient AES implementations for ARM based platforms // Proceedings of the ACM symposium on Applied computing. – 2004. – P. 841–845. 14. Ø. Ekelund. Low Energy AES Hardware for Microcontroller // Thesis M.Sc. in Electronics, NTNU. – 2009. – 96 p. 15. Ø. Ekelund. Low Energy Cryptographic Hardware - Advanced Encryption Standard // Project report, NTNU. – 2008. 16. D. A. Osvik. Fast embedded software hashing // Cryptology ePrint Archive, Report 2012/156. – 2012. – 15 p. 17. Efficient Implementation of Security Systems // European Network of Excellence in Cryptology (ECRYPT II). – 2013. – 60 p. 18. XMEGA AU Manual // Atmel, 2013, 478 p. 19. RM0031. Reference Manual //STM. – 2015. – 595 p. 20. User Manual. C8051F96x // Silicon Labs. – 2013. – 492 p. 21. User's Guide. MSP430x5xx/6xx Family // Texas Instruments. – 2014. – 1145 p. 22. User's Guide. MSP430FR58xx/59xx/68xx, and MSP430FR69xx Family // Texas Instruments. – 2015. – 808 p. 23. PIC24FJ128GA204 FAMILY. Datasheet // Microchip Technology. – 2015. – 438 p. 24. Datasheet Atmel SAM7XC512/256/128 // Atmel. – 2015. – 759 p. 25. Datasheet Atmel SAM4E Series // Atmel. – 2015. – 1452 p. 26. Datasheet Atmel SAM E70 Series // Atmel. – 2015. – 1790 p. 27. Datasheet Atmel SAML21E/G/J Series // Atmel. – 2015. – 1177 p. 28. User Manual. LPC18xx ARM Cortex-M3 microcontroller // NXP. – 2015. – 1278 p. 29. Technical Reference Manual. MSP432P4xx Family // Texas Instruments. – 2015. – 802 p. 30. SiM3U1xx/C1xx Reference Manual // Silicon Labs. – 2012. – 854 p. 31. EFM32 Pearl Gecko Family EFM32PG1 Reference Manual // Silicon Labs. – 2016. – 953 p. 32. Reference Manual. STM32F405xx/407xx/415xx/417xx advanced ARM-based 32-bit MCUs // STMicroelectronics. – 2011. – 1316 p. 33. Самоделов А. Криптография в отдельном блоке: криптографический сопроцессор семейства STM32F4xx // Новости электроники. – 2012. – № 6 (108). – С. 12–25. 34. Совин Я. Р., Наконечний Ю. М, Стахів М. Ю. Дослідження характеристик вбудованого генератора випадкових чисел мікроконтролерів родини STM32F4XX згідно з методикою NIST STS // Вісник НУ "Львівська політехніка" "Автоматика, вимірювання та керування". – 2013. – № 753. – С. 37–44.