© Sovyn Ya., Nakonechny Yu., Opirskyy I., Stakhiv M. Analysis of hardware support of cryptography in Internet of Things-devices // Ukrainian Scientific Journal of Information Security, 2018, vol. 24, issue 1, p. 36-48.

DOI: 10.18372/2225-5036.24.12491

АНАЛІЗ АПАРАТНОЇ ПІДТРИМКИ КРИПТОГРАФІЇ У ПРИСТРОЯХ ІНТЕРНЕТУ РЕЧЕЙ

Ярослав Совин, Юрій Наконечний, Іван Опірський, Марта Стахів

Національний університет «Львівська Політехніка»

СОВИН Ярослав Романович, к.т.н.

Рік та місце народження: 1979 рік, м. Львів, Україна.
Освіта: Національний університет «Львівська Політехніка», 2001 рік.
Посада: доцент кафедри захисту інформації з 2011 року.
Наукові інтереси: безпека вбудованих систем, апаратна криптографія, легковагова криптографія, ефективна реалізація криптографічних алгоритмів у вбудованих системах та IoT з підвищеною стійкістю до фізичних атак, атаки через сторонні канали.
Публікації: понад 40 наукових публікацій, серед яких наукові статті, навчальний посібник, тези та матеріали доповідей на конференціях.
E-mail: sovynjarosl@gmail.com

НАКОНЕЧНИЙ Юрій Маркіянович, к.т.н.

Рік та місце народження: 1973 рік, м. Львів, Україна.
Освіта: Державний університет «Львівська Політехніка», 1995 рік.
Посада: доцент кафедри захисту інформації з 2011 року.
Наукові інтереси: синтез нейронних мереж, адаптивні системи захисту інформації з використанням нейромережевих технологій, системи менеджменту в інформаційній безпеці.
Публікації: понад 30 наукових публікацій, серед яких наукові статті, монографія, навчальний посібник, тези та матеріали доповідей на конференціях.
E-mail: nak15@ukr.net

ОПІРСЬКИЙ Іван Романович, к.т.н.

Рік та місце народження: 1987 рік, м. Сімферополь, АР Крим, Україна.
Освіта: Національний університет «Львівська Політехніка», 2008 рік.
Посада: старший викладач кафедри захисту інформації з 2015 року.
Наукові інтереси: методи і засоби технічного захисту інформації, проектування комплексних систем захисту інформації, лазерні системи акустичної розвідки, математичні методи та моделі захисту інформації, спецвимірювання.
Публікації: більше 50 наукових публікацій, серед яких наукові статті, монографія, навчальний посібник, тези та матеріали доповідей на конференціях.
E-mail: iopirsky@gmail.com

СТАХІВ Марта Юріївна, к.т.н.

Рік та місце народження: 1978 рік, м. Львів, Україна.
Освіта: Національний університет «Львівська Політехніка», 2000 рік.
Посада: старший викладач кафедри захисту інформації з 2007 року.
Наукові інтереси: теорія інформації та кодування, методи і засоби технічного захисту інформації, безпека інформаційних технологій, проектування комплексних систем захисту інформації, математичні методи моделювання та оптимізації процесів.
Публікації: понад 20 наукових публікацій, серед яких наукові статті, тези та матеріали доповідей на конференціях.
E-mail: martast75@gmail.com

Анотація. *У даній статті проаналізовано характеристики та функціональні можливості вбудованих криптоакселераторів у 8/16/32-бітових мікроконтролерах загального призначення, покликаних адаптувати традиційну криптографію до вимог пристроїв Інтернету речей. Встановлено, що традиційні криптоалгоритми і протоколи, що застосовуються в мережі Інтернет при програмній реалізації не відповідають вимогам, які ставляться до пристроїв Інтернету речей. Показано тенденції розвитку легковагової криптографії та криптоакселераторів у мікроконтролерах з точки зору балансу безпеки, вартості і продуктивності. Оцінено виграш в продуктивності при застосуванні криптоакселераторів для шифрування, хешування та генерації випадкових чисел у порівнянні з оптимізованими програмними реалізаціями. Звертається увага на методи захисту криптоакселераторів від атак через сторонні канали, у першу чергу атак на енергоспоживання, що становлять головну небезпеку.*

Ключові слова: *криптоакселератори, IoT, мікроконтролери, вбудовані системи, шифрування, хеш, ГВЧ.*

Вступ

Інтернет речей (англ. Internet of Things – IoT) є одним з головних трендів розвитку IT-технологій і поступово стає невід'ємною частиною нашого щоденного життя, оскільки забезпечує швидку і зручну взаємодію з різноманітними сервісами. Згідно прогнозу корпорації Cisco вже у 2020 році число підключених до IoT пристроїв досягне 50 млрд., або 6.6 пристроїв на кожного мешканця планети [1].

Інтернет речей – це мережа фізичних об'єктів, які мають вбудовані технології для взаємодії з зовнішнім середовищем. IoT став популярним терміном опису сценаріїв, у яких Інтернет-з'єднання (переважно бездротові) і обчислювальні можливості поширюються на безліч пристроїв, сенсорів і повсякденних об'єктів. Становлення IoT є однією з основних причин трансформації ринку вбудованих систем (ВС) в напрямку розробки інтелектуальних систем, об'єднаних в єдину глобальну обчислювальну мережу з метою отримання і обробки даних для підвищення ефективності виробництва або комфорту користувачів.

Оскільки IoT привносить мережевий інтелект у фізичні речі навколо нас, особливо гостро постає питання безпеки. Життя сучасної людини залежить від транспортної, промислової, комунальної, цивільної та медичної інфраструктури, незаконне маніпулювання якими може привести до трагічних наслідків. Не менш важливим є захист приватного життя і персональних даних, доступ до яких зловмисники можуть отримати через злам систем промислово-побутової автоматизації, моніторингу, безпеки і контролю доступу (технології smart home), wearable-електроніки (фітнес-трекерів, розумних годинників, окулярів), домашньої електроніки тощо. IoT-пристрої можуть бути не лише об'єктом атаки, але і суб'єктом, наприклад, IoT-ботнети використовуються зловмисниками для організації DDoS-атак, поширення вірусів і т.д. Проблеми пов'язані з безпекою і відповідністю нормативним вимогам є основними стримуючими факторами впровадження IoT-технологій.

Щоб захистити рішення IoT, потрібно забезпечити безпеку пристроїв (цілісність коду з можливістю віддаленого поновлення Over-The-Air) і їх підключення до хмари (захищені протоколи типу TLS), конфіденційність даних в хмарі під час передачі, обробки і зберігання, а також стійкість до віртуальних і фізичних атак. В Інтернеті захист даних від несанкціонованого доступу і збереження інформації своїх основних властивостей (конфіденційність, автентичність та цілісність) реалізується криптографічними методами, такими як шифрування та хешування. Разом з тим для вироблення ключів та векторів ініціалізації необхідні генератори випадкових чисел (ГВЧ).

Проникнення стандартних комп'ютерних мереж у світ IoT-технологій супроводжується значними вимогами до безпеки, захисту та приватності. У вбудованих системах обчислювальна потужність сконцентрована у центральних процесорах мікроконтролерів (МК) загального призначення, для яких ціна та енергоспоживання виходять на перший план. Тому основною проблемою захисту інформації та впровадження криптографічних методів в IoT-пристроях є те, що надзвичайно важко в кінцевому виробі одночасно оптимізувати рівень безпеки, ціну та продуктивність – три неодмінні умови успішного проекту. Особливістю IoT-пристроїв, яка додатково утруднює криптозахист, крім обмежених ресурсів пристроїв є їх гетерогенність і географічний розподіл.

Перенесення апробованих для Інтернету криптографічних рішень на специфічні пристрої IoT є непростою задачею. Перевірені часом та поширені криптографічні алгоритми і протоколи (AES, SHA, HMAC, RSA, DH, ECC і т.д.) розроблялися, в першу чергу, для імплементації в інформаційних системах, побудованих на високопродуктивних мікропроцесорах загального призначення, для яких характерні значна обчислювальна потужність, великий об'єм пам'яті, доступні джерела живлення. Як наслідок, існуючі криптоалгоритми досить погано пристосовані до застосування у гетерогенних пристроях IoT, більшість з яких базується на 8/16/32-бітових процесорах з малими обчислювальними ресурсами. Операції шифрування і хешування при традиційній програмній реалізації у мікроконтролерах досить повільні та потребують значних витрат постійної і оперативної пам'яті, споживають багато енергії. Генерація випадкових чисел у мікроконтролері, який є детермінованою системою, теж викликає відчутні складнощі, – у результаті відомі ГВЧ або повільні або не якісні. Ще одним фактором тиску на безпеку IoT-пристроїв є їх належність до масового ринку з високим рівнем конкуренції, а це змушує максимально економити ресурси, щоб досягнути прийнятної ціни, тому безпека повинна бути дешевою і функціональною.

Для подолання вищевказаних проблем у мікроконтролери загального призначення стали включатися спеціалізовані апаратні модулі – так звані

криптоакселератори (криптомодулі), які дозволяють значно прискорити (в десятки, сотні і навіть тисячі разів) виконання певних криптоалгоритмів. Криптоакселератори (КА) працюють окремо від ядра і, таким чином, ядро процесора може практично не брати участь в криптографічних обчисленнях, зберігаючи свої ресурси для виконання інших завдань або економлячи енергію.

Криптоакселератори, на відміну від стандартних інтерфейсів (SPI, I²C, SDRAM, USB і т.д.), мають пропрієтарні реалізації, що ускладнює цільовий вибір. Відповідно, порівняльний аналіз криптоакселераторів у популярних, орієнтованих на IoT, лінійках мікроконтролерів дасть змогу підібрати конкретного виробника, оцінити швидкодію та виграш в продуктивності порівняно з програмними реалізаціями криптоалгоритму, а отже зробити висновок про доцільність їх використання для реальної аплікації.

Аналіз останніх досліджень і публікацій

Проблему імплементації традиційних криптоалгоритмів у вбудованих системах спершу намагалися вирішити на програмному рівні, для чого максимально оптимізували код за рахунок використання мови асемблера і особливостей архітектури процесора. Сформувався ряд криптобібліотек, орієнтованих на ВС та IoT, найвідоміші з яких WolfSSL, OpenSSL, GUARD TLS Tiny/Toolkit, Cifra, містять реалізації як окремих алгоритмів так і цілих протоколів з помірними вимогами до ресурсів обчислювача.

Разом з тим і сама криптографія намагалася підлаштуватися під вимоги IoT, і на початку 2000-х років виділився такий окремий напрям як легковагова або малоресурсна криптографія (Lightweight Cryptography) для пристроїв з обмеженими ресурсами. При створенні легковагових криптоалгоритмів на перше місце виходить вартість реалізації при адекватному рівні захисту і потрібній продуктивності, тобто важливий компроміс між цими трьома параметрами, який залежить від конкретних вимог до пристрою. У порівнянні з класичними алгоритмами, легковагові алгоритми за рахунок зменшення розміру ключа, кількості раундів, заміни більш складних операцій простішими або відмови від них дозволяють суттєво збільшити продуктивність та знизити вимоги до ресурсів реалізації. Як приклад програмно-орієнтованих легковагових алгоритмів, які здобули значну популярність останніми роками можна вказати шифри ChaCha20, Speck, хеш Blake2, MAC-функцію Poly1305 та ін.

Крім того з'явилися режими роботи шифрів, покликані комплексно і з мінімальними накладними витратами забезпечити конфіденційність, цілісність і автентифікацію пакетизованих даних. У першу чергу тут варто відзначити режим GCM [2], що належить до режимів автентифікованого шифрування з приєднаними даними (Authenticated Encryption with Associated Data, AEAD), який дозволяє в одному алгоритмі поєднати операції шифрування (AES в режимі лічильника CTR) і вироблення MAC-коду на основі функції GHASH (множення в полях Галуа), при цьому частина даних (заголовок) лишаються у відкритій формі, проте весь пакет є повністю автентифікований. Режим AEAD-AES-GCM став стандартом де-юре в багатьох Інтернет-протоколах (зокрема TLS, IPSec) і де-факто для багатьох криптобібліотек і аплікацій. З метою вдосконалення цього режиму у 2013 р. анонсований відкритий конкурс CAESAR (Competition for Authenticated Encryption: Security, Applicability, and Robustness) покликаний сформувати портфоліо AEAD-шифрів, які б переважали за швидкодією AES-GCM та були придатні для широкого використання [3]. У фінал вийшли 7 алгоритмів з 57 і переможці мали бути оголошені у грудні 2017 р. проте на момент написання статті (березень 2018 р.) цього ще не відбулося.

Попри те, навіть високопродуктивні мікропроцесори загального призначення (Intel, AMD) з високими тактовими частотами, великими об'ємами оперативної та кеш-пам'яті, потужною системою команд і підтримкою багатопоточності зіткнулися з проблемою недостатньої продуктивності при реалізації криптоалгоритмів. Для вирішення цієї проблеми виробники стали переміщати криптографічну обробку даних в апаратні блоки своєї продукції – криптоакселератори. Прискорена апаратна криптографічна обробка замість програмного виконання цих же алгоритмів дозволяє суттєво розвантажити центральний процесор.

Прикладом такого підходу є розширення системи команд x86 шістьма командами AES-NI (AES-New Instructions): *AESENC, AESENCLAST, AESDEC, AESDECLAST, AESKEYGENASSIST*, *AESIMC* з метою прискорення додатків, що використовують AES-шифрування [4]. Поєднання AES-NI з інструкцією множення в полях Галуа *PCLMULQDQ*, для ефективного обчислення функції GHASH, дало змогу суттєво збільшити швидкодію в режимі AEAD-AES-GCM (рис. 1) [5].

Ще один спосіб пришвидшення криптографічних операцій завдяки паралельним обчисленням – це використання векторних інструкцій, що дозволяють виконувати кілька операцій за один такт процесора. Можливість виконання векторних операцій в обчислювальних системах на платформах x86 забезпечується спеціальними процесорними розширеннями SSE і AVX [6].

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 з 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 з 128 до 256 біт (регістри *YMM0-YMM15*), а для роботи з *YMM*-регістрами додані нові 256-бітові AVX-інструкції. Надалі на заміну AVX-256 прийшло AVX-512.

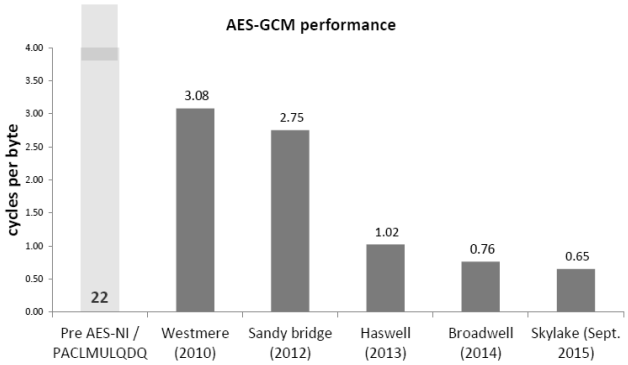


Рис. 1. Вплив апаратної підтримки на швидкодію виконання AEAD-AES-GCM в процесорах Intel [5]

У статті [7] показано, як можна застосувати ці розширення для ефективної реалізації алгоритму блокового симетричного шифрування ГОСТ 28147-89 на серверних і користувацьких ПК, що дало збільшення продуктивності в 2-2,5 рази (рис. 2).

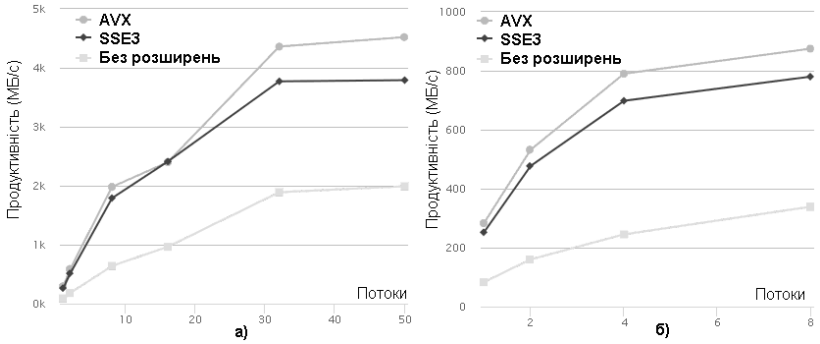


Рис. 2. Шифрування ГОСТ 28147-89 в режимі гамування для Intel Xeon E5-2680 (а) та Intel Core i7 (б) [7]

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 [8]. Intel називає його «цифровий генератор випадкових чисел» (Digital Random Number Generator, DRNG).

DRNG можна розбити на три логічні рівні:

1. Джерело ентропії, яке продукує випадкові біти з недетермінованого апаратного процесу з використанням теплового шуму в напівпровідниках і передає їх блоку підготовки. 2. Блок підготовки даних за алгоритмом AES-CBC-MAC, який здійснює 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. Генерує випадкові дані великого об'єму з високою швидкодією (до 6 Гбіт/сек), використовуючи стандартний алгоритм CTR-DRBG на базі AES. Дані поступають в 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 в буфер, з якого зчитуються інструкціями *RDSEED*.

Аналогічний ГВЧ також реалізований у процесорах AMD у складі AMD Secure Processor.

У вбудованих системах криптоалгоритми довгий час реалізувалися тільки програмним чином, і лише відносно недавно масово почали з'являтися інтегровані в мікроконтролери криптоакселератори, які будуть розглянуті в статті. Використання криптоакселераторів дає наступні переваги: вища швидкодія і енергоефективність, розвантаження центрального процесора, економія пам'яті, більша стійкість до Side-Channel Attacks (в першу чергу CPA і DPA).

Щоб мати базу для порівняння з апаратними прискорювачами коротко розглянемо програмні реалізації симетричних і асиметричних криптоалгоритмів у ВС. Оцінці швидкодії та вимог до пам'яті найпоширеніших з них для різних мікроконтролерних архітектур присвячені численні дослідження, з яких у табл. 1-3 представлені найбільш швидкодіючі реалізації відомі авторам. У цих роботах розглядаються впливи як архітектури, так і різних шляхів оптимізації на рівні алгоритму і компілятора, на продуктивність та обсяг необхідної пам'яті.

Метою статті є порівняльний аналіз криптоакселераторів у найпоширеніших 8/16/32-бітових родинах мікроконтролерів, з точки зору швидкодії та гнучкості роботи, що дозволить обгрунтовано вибирати оптимальне рішення при розробленні механізмів захисту у IoT-пристроях.

Криптоакселератори у 8-бітових мікроконтролерах

AVR. Усі мікроконтролери AVR родини XMega фірми Atmel оснащені криптоакселераторами блокових симетричних шифрів DES і AES [16].

Зокрема у системі команд мікроконтролерів XMega передбачена інструкція *DES K*, яка відповідає одному (*K*-му) з 16-ти раундів алгоритму DES. Вхідні дані для команди розташовуються у регістрах загального призначення (РЗП) *R0-R7*, ключ записується у РЗП *R8-R15*. Прапорець *H* регістру стану *SREG* задає тип операції: *H* = 0 – зашифрування, *H* = 1 – розшифрування (рис. 3.а).

Параметри програмних реалізацій блокових симетричних шифрів Таблиця 1

перація	CPU	Зашифрування, тактів/байт	Розшифрування, тактів/байт	ROM, байт
TDES [9]	AVR (8 біт)	3 946	3 943	1 748
AES-128 [9]	AVR (8 біт)	243	376	2 424
AES-256 [9]	AVR (8 біт)	347	535	2 742
TDES [9]	i8051 (8 біт)	4 688	4 700	1 570
AES-128 [9]	i8051 (8 біт)	188	275	2 246
AES-256 [9]	i8051 (8 біт)	280	404	2 394
AES-128 [10]	MSP430 (16 біт)	223	267	5 160
TDES [11]	PIC24 (16 біт)	1 695	1 695	7 500
AES-128 [11]	PIC24 (16 біт)	176	281	3 018
AES-128 [12]	ARM7TDMI (32 біти)	40	40	5 966
AES-128 [9]	ARM Cortex-M0 (32 біти)	142	271	3 998
AES-256 [9]	ARM Cortex-M0 (32 біти)	194	361	4 240
AES-128 [13]	ARM Cortex-M3 (32 біти)	87	106	1 898
AES-256 [13]	ARM Cortex-M3 (32 біти)	122	150	1 898

Параметри програмних реалізацій хеш-, MAC- і AEAD-алгоритмів Таблиця 2

Операція	CPU	Тактів/байт	ROM, байт
SHA-1 [14]	AVR (8 біт)	177	1 352
SHA-256 [14]	AVR (8 біт)	335	2 720
SHA-1 [9]	i8051 (8 біт)	673	1 153
SHA-256 [9]	i8051 (8 біт)	1 780	1 760
SHA-1 [9]	ARM Cortex-M0 (32 біти)	129	612
SHA-256 [9]	ARM Cortex-M0 (32 біти)	199	980
SHA-1 [9]	ARM Cortex-M3 (32 біти)	116	728
SHA-256 [9]	ARM Cortex-M3 (32 біти)	117	1 040
HMAC-SHA-1 [9]	AVR (8 біт)	2 065	1 458
HMAC-SHA-256 [9]	AVR (8 біт)	3 425	2 224
HMAC-SHA-1 [9]	i8051 (8 біт)	2 713	1 377
HMAC-SHA-256 [9]	i8051 (8 біт)	7 147	2 208
HMAC-SHA-1 [9]	ARM Cortex-M0 (32 біти)	569	812
HMAC-SHA-256 [9]	ARM Cortex-M0 (32 біти)	845	1 380
HMAC-SHA-1 [9]	ARM Cortex-M3 (32 біти)	414	946
HMAC-SHA-256 [9]	ARM Cortex-M3 (32 біти)	498	1 476
AEAD-AES-128-GCM [15]	ARM Cortex-M0 (32 біти)	1 279	2 644
AEAD-AES-256-GCM [15]	ARM Cortex-M0 (32 біти)	1 415	2 728

Параметри програмних реалізацій цифрового підпису на еліптичних кривих Таблиця 3

Операція	CPU	Генерування підпису, тактів	Верифікація підпису, тактів	ROM, байт
ECDSA-192 [9]	AVR (8 біт)	86 000 000	173 000 000	14 426
ECDSA-192 [9]	i8051 (8 біт)	151 000 000	304 000 000	15 869
ECDSA-192 [9]	ARM Cortex-M0 (32 біти)	44 000 000	88 000 000	7 236
ECDSA-192 [9]	ARM Cortex-M3 (32 біти)	8 200 000	17 200 000	6 990

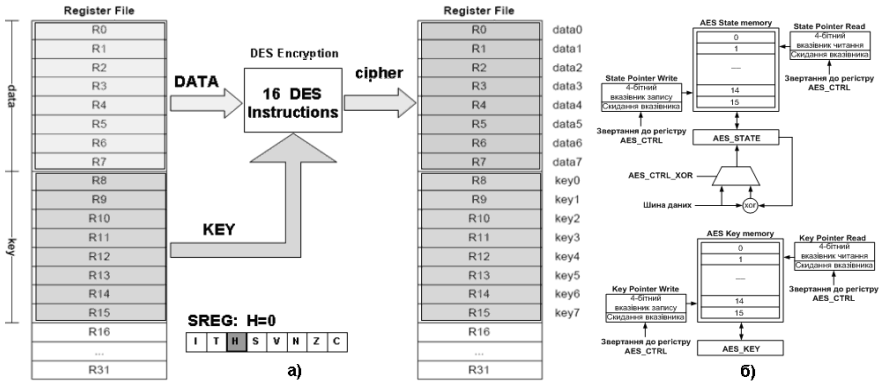


Рис. 3. Виконання операції шифрування алгоритмом DES (а) та структура криптомодуля AES (б) у мікроконтролерах родини XMega

Крім підтримки алгоритму DES на рівні системи команд у мікроконтролерах XMega реалізована апаратна підтримка алгоритму AES з допомогою криптомодуля AES (рис. 3.б).

Криптомодуль AES є периферійним модулем, який шифрує дані блоками по 128 біт з допомогою 128-бітного ключа. Відповідно криптомодуль AES має пам'ять для зберігання блоку даних (*AES State Memory*) та 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 в пам'ять (Direct Memory Access, DMA) дозволяє виконувати пересилки вхідних і вихідних даних без втручання центрального процесора.

STM8. У мікроконтролерах родин STM8L16 і STM8AL присутній криптоакселератор алгоритму AES-128 (рис. 4.а). Безпосередньо підтримується тільки режим ECB. КА забезпечує DMA-передачі, як для вхідних так і вихідних даних, що розвантажує центральний процесор від операцій пересилки [17].

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*.

Відкритий текст, шифртекст або ключ записуються у вхідний регістр *AES_DINR*. Після завершення обчислень встановлюється відповідний прапорець в регістрі статусу *AES_SR* і може генеруватися переривання. Зчитуються дані з вихідного регістру *AES_DOUT*.

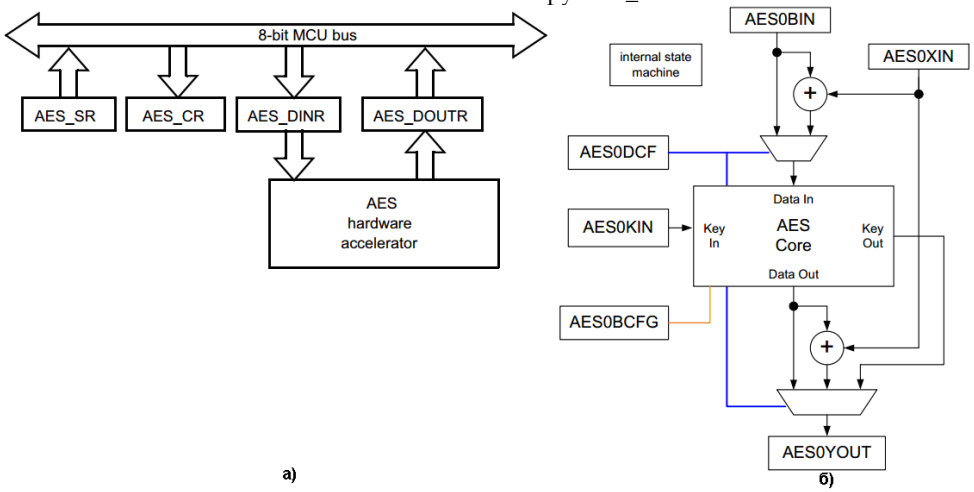


Рис. 4. Блок-схеми криптоакселераторів AES мікроконтролерів STM8 (а) та C8051F96x (б)

i8051. Мікроконтролери родини C8051F96x фірми Silicon Labs з процесорним ядром i8051 містять криптоакселератор алгоритму AES з підтримкою ключів довжиною 128, 192 та 256 біт і безпосередньо можуть працювати в режимах ECB, CBC, CTR [18].

Криптоакселератор складається з таких елементів (рис. 4.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 автомату.

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

У таблицях 4-6 крім кількості тактів на обробку блоку, взятої з технічної документації, наведено перераховану кількість тактів на обробку одного байту *CPB* (cycles per byte) для коректного порівняння швидкодії при різних розмірах блоку. Параметр N_B вказує скільки байт здатний обробити криптоакселератор за 1 мс, при роботі мікроконтролера на максимальній тактовій частоті $N_B = F_{CPU} / (1000 \times CPB)$. .

Характеристики криптоакселераторів 8-бітових МК — Таблиця 4

Операція	CPU	Родина/модель МК	F_{CPU}, МГц	Режими	Тактів/блок	CPB, тактів/байт	N_B, байт
Enc./Dec. DES	AVR	XMega	32	ECB	17	2,1	15 059
Enc./Dec. AES-128				ECB, CBC	375	23,4	1 365
Enc./Dec. AES-128	STM8	STM8L16, STM8AL	16	ECB	892	55,8	287
Enc./Dec. AES-128	i8051	C8051F96x	25	ECB, CBC, CTR	218	13,6	1 835
Enc./Dec. AES-256					298	18,6	1 342

Криптоакселератори у 16-бітових мікроконтролерах

MSP430. В окремих моделях мікроконтролерів родини MSP430F6xx присутній криптоакселератор алгоритму AES-128 (рис. 5.а) [19], а у родинах МК MSP430FR5x/FR6x з FRAM-пам'яттю – AES-128/192/256 [20]. 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 в регістр *AESDIN*, вихідні дані зчитуються з регістру *AESDOUT*, ключ заноситься у регістр *AESAKEY* (рис. 5.б). Криптоакселератори підтримують DMA-передачі та режими ECB, CBC, OFB, CF.

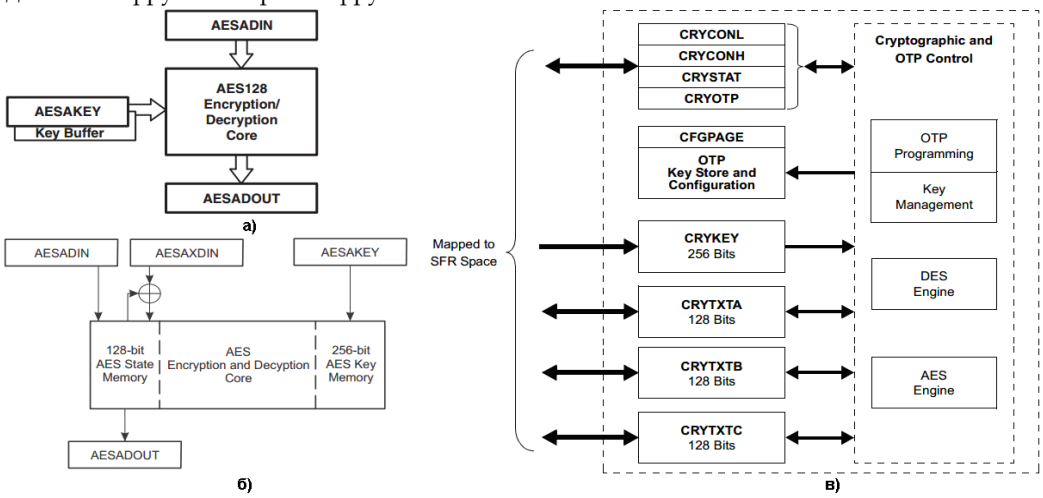


Рис. 5. Блок-схеми криптоакселераторів AES в МК MSP430F6xx (а), MSP430FR5x/6x (б) та PIC24 (в)

PIC24. У мікроконтролерах родин PIC24FJ64/128/256 з ядром PIC24 присутній криптомодуль (Cryptographic Engine) з широкими функціональними можливостями. Криптоакселератори, які входять в цей модуль підтримують шифрування алгоритмами DES, TDES та AES в режимах ECB, CBC, CFB, OFB і CTR. У модулі передбачені кола захисту від злому і проникнення в мікросхему [21]. Також модуль містить 512-біт OTP-пам'яті (One-Time Programmed), яка служить для захищеного зберігання секретних ключів (рис. 5.).

Характеристики криптоакселераторів 16-бітових МК Таблиця 5

Операція	CPU	Родина/модель МК	F_{CPU}, МГц	Режими	Тактів/ блок	CPB, тактів/байт	N_B, байт
Enc./Dec. AES-128	MSP430	MSP430F6xx	25	ECB, CBC, OFB, CFB	167	10,4	2 395
Enc./Dec. AES-128		MSP430FR5x, MSP430FR6x	16	ECB, CBC, OFB, CFB	168	10,5	1 524
Enc./Dec. AES-256					234	14,6	1 094
Enc./Dec. TDES	PIC24	PIC24FJ64, PIC24FJ128, PIC24FJ256	32	ECB, CBC, OFB, CFB, CTR	26	3,3	9 846
Enc./Dec. AES-128					219	13,7	2 338
Enc./Dec. AES-256					299	18,7	1 712

Криптоакселератори у 32-бітових мікроконтролерах

ARM7TDMI. До складу мікроконтролерів родини SAM7XC фірми Atmel з ядром ARM7TDMI, входять криптоакселератори алгоритмів AES і DES/TDES [22]. Обидва криптоакселератори підтримують режими шифрування ECB, CBC, CFB, OFB, CTR (лише для AES). У мікроконтролерах передбачені апаратні заходи з протидії атакам аналізу енергоспоживання, хоча виробник не розкриває інформацію про деталі.

ARM Cortex-M. Мікроконтролери фірми Atmel родини SAM L21 (з ядром ARM Cortex-M0+), SAM E5x/D5x (ARM Cortex-M4) і SAM S70/V70/V71 (ARM Cortex-M7) мають криптоакселератор алгоритму AES з підтримкою 128/192/256-бітових ключів. КА може працювати в режимах ECB, CBC, OFB, CFB, CTR та GCM (включно з апаратним виконанням операції GHASH) [23-25].

У цих МК також присутній справжній генератор випадкових чисел (True Random Number Generator, TRNG), що відповідає тестам NIST SP 800-22, спеціально розробленими для оцінки криптографічних ГВЧ. Кола онлайн-перевірки якості генерації та виявлення збоїв в цих TRNG відсутні.

У родинах МК SAM S70, SAM V70, SAM V71 з ядром ARM Cortex-M7 наявний блок перевірки цілісності пам'яті, що обчислює хеш-функції SHA-1, SHA-224 і SHA-256.

Також у мікроконтролерах SAM E5x/D5x/L під час шифрування AES передбачені апаратні заходи з протидії атакам аналізу енергоспоживання, які можуть включати:

– випадкове додавання одного такту в процесі обробки даних;

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

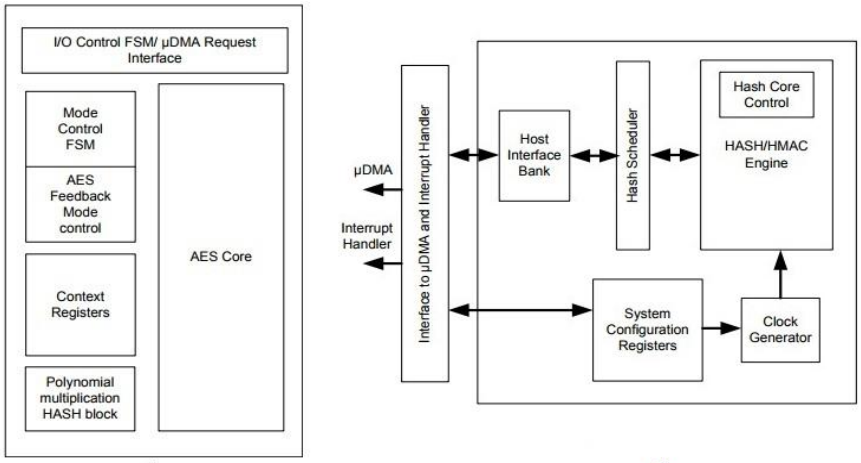
– додавання випадкового енергоспоживання в

процесі обробки даних.

ARM Cortex-M. Мікроконтролери родини MSP432P4xx фірми Texas Instruments з ядром ARM Cortex-M4F використовують криптоакселератор алгоритму AES з 128/192/256-бітними ключами [26]. Безпосередньо підтримуються режими ECB, CBC, OFB, CFB.

Для IoT фірма Texas Instruments позиціонує мікроконтролери родини Tiva C Series TM4C129x з ядром ARM Cortex-M4F оснащені КА алгоритмів AES/DES/HASH/HMAC. Акселератор AES-128/192/256 працює в режимах ECB, CBC, CTR, CFB, GCM (включно з операцією GHASH), CCM, XTS (рис. 6.а). Акселератор DES/TDES підтримує режими ECB, CBC, CFB. Хеш-акселератор (рис. 6.б) здатний обчислювати хеш MD5, SHA-1, SHA-224/256 і код автентифікації повідомлення за алгоритмом HMAC [27].

Аналогічні з Tiva C Series TM4C129x криптографічні акселератори з однаковими характеристиками має також родина МК MSP432E4 з ядром Cortex-M4F.

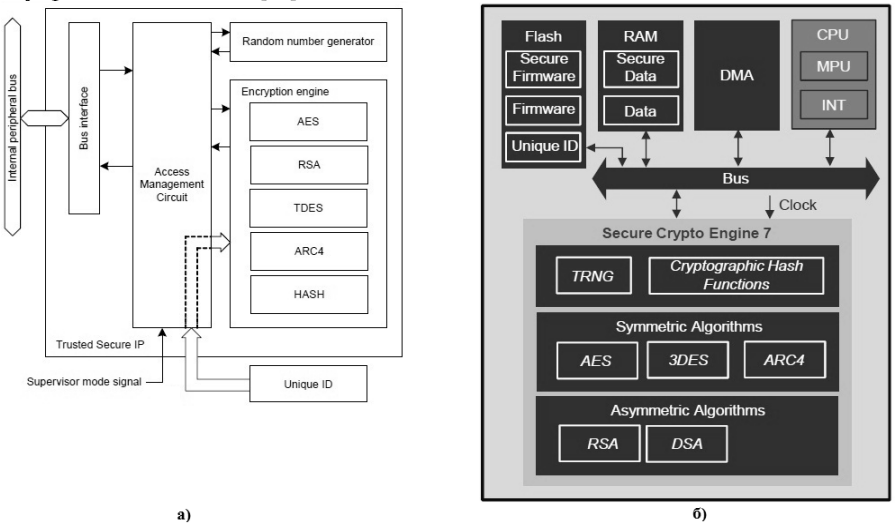


а) б)

Рис. 6. Блок-схеми криптоакселераторів в МК TM4C129x: AES (а) та HASH/HMAC (б)

RXv2. Мікроконтролери родини RX600 груп RX65N і RX651 фірми Renesas з пропрієтарним ядром RXv2 використовують криптомодуль Trusted Secure IP (рис. 7.а) для алгоритму AES-128/192/256 в режимах ECB, CBC, CTR, CCM, GCM, XTS, GCTR. Також він здатний шифрувати відповідно до DES/ TDES в режимах ECB, CBC. Для хешування доступні алгоритми MD5, SHA-1, SHA-224/256, GHASH. Є можливість роботи з RSA-алгоритмом з довжиною ключа і блоку до 2048 біт. У модуль Trusted Secure IP закладено значні можливості з управління ключами [28].

ARM Cortex-M. Мікроконтролери родини Synergy S7G2 фірми Renesas з ядром ARM Cortex-M4 використовують комплексний апаратний модуль Secure Crypto Engine 7 (SCE7) для вирішення широкого спектру криптографічних задач (рис. 7.б). SCE7 підтримує алгоритми AES-128/192/256 в режимах ECB, CBC, CTR, GCM, XTS, GCTR і DES/TDES в режимах ECB, CBC. Для хешування доступні алгоритми MD5, SHA-1, SHA-224, SHA-256, GHASH. Асиметрична криптографія в модулі представлена алгоритмами RSA і DSA з довжинами ключа до 2048 біт [29].



а) б)

Рис. 7. Криптоакселератори в МК Renesas: Trusted Secure IP (а) і Secure Crypto Engine 7 (б)

ARM Cortex-M. Мікроконтролери родини SiM3U1xx/SiM3C1xx фірми Silicon Laboratories з ядром ARM Cortex-M3 мають криптоакселератор алгоритму AES з підтримкою 128/192/256-бітних ключів [30]. Вони можуть працювати в режимах ECB, CBC, CTR (рис. 8.а).

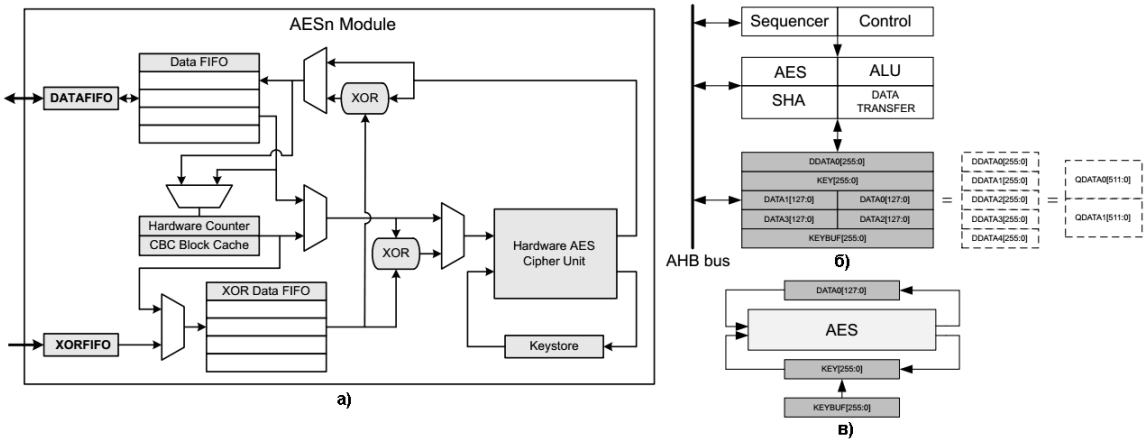


Рис. 8. Блок-схеми криптоакселераторів в МК SiM3U1xx/SiM3C1xx (а), EFM32GG11 (б) та блоку AES в EFM32GG11 (в)

ARM Cortex-M. Мікроконтролери родини EFM32GG11 з ядром ARM Cortex-M4 фірми Silicon Laboratories мають багатофункціональний криптомодуль CRYPTO, який дає змогу реалізувати більшість стандартних криптографічних операцій [31]. Операції в CRYPTO виконуються в окремому АЛП згідно заданих послідовностей спеціалізованих інструкцій (занесених в *Sequencer*) над 128/256/512-бітовими регістрами (рис. 8.б). Всього 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 довжиною 128/256-біт та буфер для їх зберігання (рис. 8.в). Є можливість працювати в 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 та SHA-224/256.

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

ARM Cortex-M. Компанія STMicroelectronics додала в свої мікроконтролери STM32F2xx/F4xx/F7xx/H7xx, з 32-бітовими ядрами ARM Cortex-M3/M4F/M7F/M7F відповідно, криптопроцесор, який дозволяє шифрувати дані, обчислювати хеш повідомлення і генерувати випадкові числа. При майже однакових функціональних можливостях криптопроцесорів суттєво відрізняються максимальні тактові частоти цих родин мікроконтролерів, які становлять 120, 168, 216 і 400 МГц відповідно.

Криптографічний процесор складається з ядра, що реалізує алгоритми, буферів вхідних/вихідних даних, регістрів зберігання ключів та IV, регістрів стану і регістрів управління.

До ядра криптопроцесора належать [32, 33]:

1. CRYP – КА шифрування, який реалізує на апаратному рівні алгоритми DES/TDES/AES;

2. HASH – криптоакселератор обчислення хеш-функцій і кодів автентифікації повідомлень;

3. RNG – генератор 32-розрядних випадкових чисел на базі фізичного джерела шуму.

Криптоакселератор шифрування (CRYP). КА CRYP призначений для шифрування даних в режимах ECB або CBC для алгоритмів DES/TDES і додатково в режимі CTR для алгоритму AES-128/192/256 (рис. 9.а). Модуль CRYP забезпечує автоматичний контроль потоку даних з підтримкою прямого доступу до пам'яті, має вхідний і вихідний буфери FIFO, глибиною вісім слів кожен, які відповідають чотирьом блокам DES або двом блокам AES.

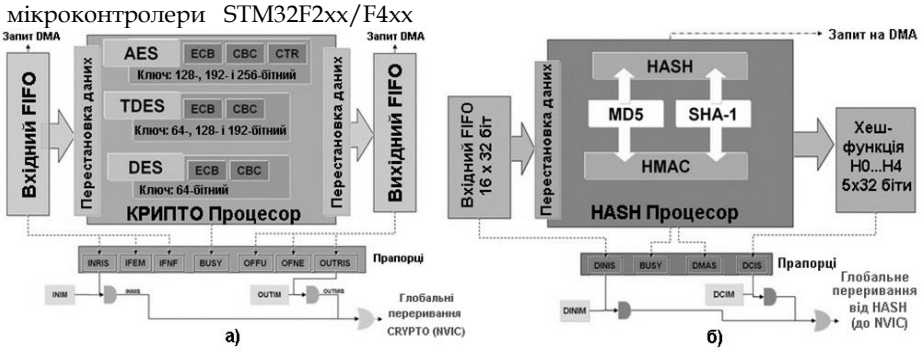


Рис. 9. Блок-схеми криптоакселераторів CRYP (а) та HASH (б) мікроконтролерів STM32F2xx/F4xx/F7xx/H7xx

Хеш-процесор (HASH). HASH-процесор представляє собою КА з реалізацією алгоритмів SHA-1, SHA-224/256, MD5 і HMAC для повідомлень довжиною до 2^{64}-1 біт (рис. 9.б). Алгоритм HMAC надає спосіб підтвердження автентичності повідомлень шляхом 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 блоку.

Генератор випадкових чисел (Random Number Generator, RNG). Модуль RNG є генератором випадко-

вих чисел, заснованим на аналоговому шумі (рис. 10), який проходить тести NIST SP 800-22 [34].

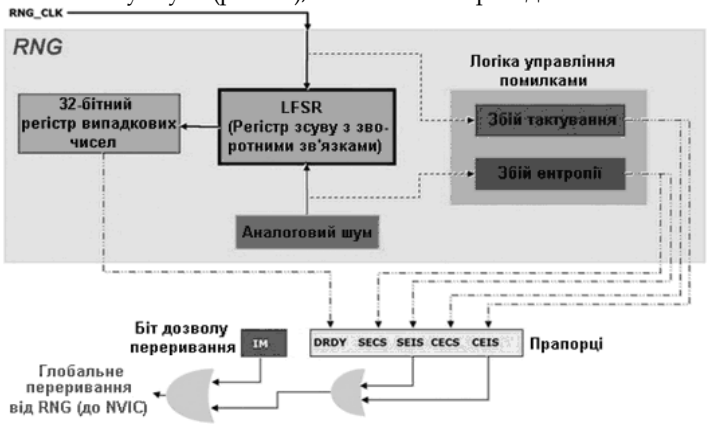


Рис. 10. Блок-схема RNG мікроконтролерів STM32F2xx/F4xx/F7xx/H7xx

Аналогові кола генерують зародок (*Analog seed*), що поступає на лінійний регістр зсуву зі зворотними зв'язками (*LFSR*). Аналогові кола побудовані на незалежних кільцевих генераторах, чиї виходи об'єднуються операцією XOR. Для тактування *LFSR* використовується окремий тактовий сигнал (*RNG_CLK*). Паралельно здійснюється моніторинг тактового сигналу *RNG_CLK* та ентропії зародку. 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 біт з однаковим значенням (0 або 1), або 32 послідовні пари 0 і 1 (01010101…01). При виявленні збою ГВЧ слід перезапустити з допомогою відповідних бітів регістра управління.

У всіх розглянутих криптоакселераторах 32-бітових МК є можливість DMA-пересилок даних.

Характеристики криптоакселераторів 32-бітових МК Таблиця 6

Операція	CPU	Родина/ модель МК	F_{CPU}, МГц	Режими	Тактів/ блок	СРВ, тактів/ байт	N_B, байт
Enc./Dec. TDES	ARM7TDMI	SAM 7XC	55	ECB, CBC, OFB, CFB	50	6,3	8 800
Enc./Dec. AES-128				ECB, CBC, OFB, CFB, CTR	12	0,8	73 333
Enc./Dec. AES-256					14	0,9	62 857
Enc./Dec. AES-128	ARM Cortex-M0+	SAM L21	48	ECB, CBC, OFB, CFB, CTR, GCM	57	3,6	13 474
Enc./Dec. AES-256					77	4,8	9 974
TRNG-32					84	21,0	2 286
Enc./Dec. AES-128	ARM Cortex-M4F	SAM E5x, SAM D5x	120	ECB, CBC, OFB, CFB, CTR, GCM	57	3,6	33 684
Enc./Dec. AES-256					77	4,8	24 935
Hash SHA-1					85	1,3	90 353
Hash SHA-256					72	1,1	106 667
TRNG-32					84	21,0	5 714
Enc./Dec. AES-128	ARM Cortex-M7	SAM E70, SAM S70, SAM V70, SAM V71	300	ECB, CBC, OFB, CFB, CTR, GCM	10	0,6	480 000
Enc./Dec. AES-256					14	0,9	342 857
Hash SHA-1					85	1,3	225 882
Hash SHA-256					72	1,1	266 667
TRNG-32					84	21,0	14 286
Enc./Dec. AES-128	ARM Cortex-M4F	MSP432P4xx	48	ECB, CBC, OFB, CFB	168	10,5	4 571
Enc./Dec. AES-256					234	14,6	3 282
Enc./Dec. AES-128		TM4C129x, MSP432E4x	120	ECB, CBC, CFB, CTR, GCM, CCM, XTS	32	2,0	60 000
Enc./Dec. AES-256					44	2,8	43 636
Hash/HMAC SHA-1					81	1,3	94 815
Hash/HMAC SHA-256					65	1,0	118 154
Enc./Dec. TDES	RXv2	RX65N, RX651	120	ECB, CBC,	48	6,0	20 000
Enc./Dec. AES-128				ECB, CBC, CTR, CCM, GCM, XTS, GCTR	11	0,7	174 545
Enc./Dec. AES-256					15	0,9	128 000
Hash SHA-1					80	1,3	96 000
Hash SHA-256					64	1,0	120 000
Enc./Dec. TDES	ARM Cortex-M4	Synergy S7G2	240	ECB, CBC	48	6,0	40 000
Enc./Dec. AES-128				ECB, CBC, CTR, GCM, XTS, GCTR	11	0,7	349 091

Продовження таблиці 6

Enc./Dec. AES-256					15	0,9	256 000
Hash SHA-1					80	1,3	192 000
Hash SHA-256					64	1,0	240 000
Enc./Dec. AES-128	ARM Cortex-M3	SiM3U1xx, SiM3C1xx	80	ECB, CBC, CTR	54	3,4	23 704
Enc./Dec. AES-256					75	4,7	17 067
Enc./Dec. AES-128	ARM Cortex-M4	EFM32GG11	72	ECB, CTR, CBC, CFB, CCM, GCM	54	3,4	21 333
Enc./Dec. AES-256					75	4,7	15 360
Enc./Dec. TDES	ARM Cortex-M7F	STM32H7xx	400	ECB, CBC	64	8,0	50 000
Enc./Dec. AES-128				ECB, CBC, CTR, GCM, CCM, GMAC	14	0,9	457 143
Enc./Dec. AES-256					18	1,1	355 555
Hash SHA-1					82	1,3	312 195
Hash SHA-256					66	1,0	387 879
TRNG-32					54	13,5	29 630

Висновки

На підставі проведеного аналізу можна відзначити чітку тенденцію щодо апаратної підтримки криптографічних примітивів для обмежених у ресурсах мікроконтролерів, що широко використовуються в IoT. Наведені в роботі дані забезпечують краще розуміння як оцінювати, розробляти та імплементувати криптографічний захист для нижнього і середнього сегментів мікроконтролерних IoT-пристроїв.

Використання криптоакселераторів дає змогу підняти швидкодію шифрування AES в 10-20 разів для 8/16-бітових МК та до 150 разів для 32-бітових МК порівняно з програмними реалізаціями алгоритму. Зростання швидкодії обчислення алгоритмів SHA-1, SHA-256 у 32-бітових МК становить більше ніж в 100 разів, а для HMAC наближається до 500. На сьогодні, за рахунок використання криптоакселераторів, IoT-пристрої з 8/16-бітними процесорами можуть забезпечити продуктивність шифрування з врахуванням накладних витрат на рівні сотень Кбайт/с, тоді як для 32-бітних мікроконтролерних ядер можна підтримувати швидкість на рівні десятків-сотень Мбайт/с.

У 32-бітових мікроконтролерах спостерігається тренд до впровадження комплексних рішень безпеки, які б не тільки пришвидшували широке коло симетричних і асиметричних алгоритмів і протоколів, але і надавали можливість захищеного зберігання та генерування ключів, безпечного завантаження і оновлення коду, підтримки цифрових підписів та сертифікатів. Заявлені виробниками характеристики дають змогу використовувати традиційні криптоалгоритми і протоколи без суттєвих обмежень, залишаючи легковагову криптографію для 8/16-бітних процесорів і ультрамалоресурсних пристроїв, типу RFID-міток та смарт-карток.

Виробники мікроконтролерів все частіше приділяють увагу захисту криптографічних блоків від атак на реалізацію, в першу чергу таких як аналіз енергоспоживання, що дуже характерні і небезпечні для вбудованих систем. Цілком природно для цього вибрано методи приховування (hiding), як найпростіших в реалізації.

Підтримка мікроконтролерних криптоакселераторів вже присутня у деяких легковагових SSL/TLS криптобібліотеках, орієнтованих на вбудовані системи, IoT та RTOS, наприклад, у wolfSSL. Враховуючи, що більшість програмних реалізацій у відомих криптобібліотеках є вразливими до side-channel атак, то перехід до апаратного виконання криптопримітивів додатково підвищує їх захищеність до атак на реалізацію.

Поданий в статті опис характеристик криптоакселераторів покликаний допомогти розібратися з програмуванням прикладних задач із захисту інформації для мікроконтролерних пристроїв Інтернету речей.

Література

[1] D. Evans, «The Internet of Things: How the Next Evolution of the Internet Is Changing Everything». URL: https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf.

[2] NIST SP 800-38D: Recommendations for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC. Gaithersburg: National Institute of Standards and Technology, 39 p., 2007.

[3] CAESAR: Competition for Authenticated Encryption: Security, Applicability, and Robustness. URL: http://competitions.cr.yp.to /caesar.html.

[4] S. Gueron, «Intel Advanced Encryption Standard (AES) Instructions Set». URL: https://software.intel.com/sites/default/files/m/d/4/1/d/8/AES_WP_Rev_03_Final_2010_01_26.pdf.

[5] S. Gueron, «AES-GCM for Efficient Authenticated Encryption Ending the Reign of HMAC-SHA-1?». URL: https://crypto.stanford.edu/RealWorldCrypto/slides/gueron.pdf.

[6] Intel, «Intel Architecture Instruction Set Extensions Programming Reference». URL: https:// software.intel.com/sites/default/files/managed/b4/3a/319433024.pdf?_ga=1.118002441.1853754838.1418826886.

[7] С. Смышляев, Е. Алексеев, А. Прохоров, «ГОСТ 28147-89: «Не спеши его хоронить». Часть 2. Эффективные реализации алгоритма». URL: http://www.cryptopro.ru/en/blog/2015/01/14/gost-28147-89-ne-speshi-ego-khoronit-chast-2-effektivnye-realizatsii-algoritma.

[8] Intel, «Digital Random Number Generator. Software Implementation Guide». URL https://software.intel.com/sites/default/files/manage/4d/91/DRNG_Software_Implementation_Guide_2.0.pdf.

[9] Cryptovia cryptographic libraries for embedded systems. URL: http://cryptovia.com/cryptographic-libraries-for-avr-cpu/.

[10] S. Didla, A. Ault, S. Bagchi, «Optimizing AES for embedded devices and wireless sensor networks», *Proceedings of the 4th Int. Conf. on Testbeds*

and research infrastructures for the development of networks & communities, p. 1-10, 2008.

[11] D. Flowers, H. Schlunder, «Data Encryption Routines for PIC24 and dsPIC Devices». URL: http://ww1.microchip.com/downloads/en/AppNotes/AN1044a.pdf.

[12] K. Atasu, L. Breveglieri, M. Macchetti, «Efficient AES implementations for ARM based platforms», *Proceedings of the ACM symposium on Applied computing*, p. 841-845, 2004.

[13] Ekelund. Low Energy AES Hardware for Microcontroller, Thesis, Moss, 96 p., 2009.

[14] D. Osvik, «Fast embedded software hashing». URL: https://eprint.iacr.org/ 2012/156.pdf.

[15] J. Birr-Pixton, «Benchmarking Modern Authenticated Encryption on €1 devices». URL: https://jbp.io/2015/06/01/modern-auth entica ted-encryption-for-1-euro.html.

[16] Atmel, «XMEGA AU Manual». URL: https://eewiki.net/download/attachments/31588436/XMEGAAU_Manual.pdf?version=1&modificationDate=1396389661997&api=v2.

[17] STMicroelectronics, «RM0031. Reference Manual». URL: http://www.st.com/content/cc/resource/technical/document/reference_manual/2e/3b/8c/8f/60/af/4b/2c/CD00218714.pdf/files/CD00218714.pdf/jcr:content/translations/en.CD00218714.pdf.

[18] Silicon Labs, «User Manual. C8051F96x». URL: https://www.silabs.com/documents/public/data-sheets/C8051F96x.pdf.

[19] Texas Instruments, «User's Guide. MSP430x5xx/6xx Family». URL: http://www.ti.com/lit/ug/slau208q/slau208q.pdf.

[20] Texas Instruments, «User's Guide. MSP430FR58xx/59xx/68xx, and MSP430FR69xx Fa mily». URL: http://www.ti.com/lit/ug/slau367o/sla u367o.pdf.

[21] Microchip Technology, «PIC24FJ128GA204 Family. Datasheet». URL: http://ww1.micrchip.com/downloads/en/DeviceDoc/30010038c pdf.

[22] Atmel, «SAM7XC512/256/128 Datasheet». URL: http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-6209-32-bit-ARM7TDMIMicro controller-SAM7XC512-SAM7XC256-SAM7XC128_Da tasheet.pdf.

[23] Atmel, «SAM L21 Family Datasheet». URL: http://www.farnell.com/data sheets/2014285.pdf.

[24] Atmel, «SAM D5x/E5x Family Datasheet». URL: https://www.mouser.com/ds/2/268/60001507A-1130176.pdf.

[25] Atmel, «SAM E70/S70/V70/V71 Family Datasheet».URL: https://www.mous er.com/ds/2/268/60001527A-1284321.pdf.

[26] Texas Instruments, «Technical Reference Manual. MSP432P4xx Family». URL: http://www.ti.com/lit/ug/slau356h/slau356h.pdf.

[27] Texas Instruments, «Tiva TM4C129D NCPDT Microcontroller Datasheet». URL: http://www.ti.com/lit/ds symlink/tm4c129dncpdt.pdf.

[28] Renesas Electronics, «Renesas 32-Bit MCU RX Family / RX600 Series. RX65N Group, RX651 Group User's Manual: Hardware». URL: https://media.digkey.com/pdf/Data%20Sheets/Renesas/RX65N_RX651_Group_HM_Rev2.10_Oct2017.pdf.

[29] Renesas Electronics, «Advanced Synergy Security». [Online]. Available at: https://www2.renesas.eu/syn_conf_downloads/it/Lectures/15%20Advanced%20Synergy%20Security.pdf.

[30] Silicon Labs, «SiM3U1xx/C1xx Reference Manual». URL: https://www.silabs.com/documents/public/data-sheets/SiM3U1xx-SiM3C1xx-RM.pdf.

[31] Silicon Labs, «EFM32 Giant Gecko 11 Family Reference Manual». URL: https://ww.silabs.com/documents/public/referencemanuals/EFM32GGRM.pdf.

[32] STMicroelectronics, «STM32H7x3 advanced ARM-based 32-bit MCUs. Reference Manual». URL: http://www.st.com/content/ccc/resource/technical/document/reference_manual/group0/c9/a3/76/fa/55/46/45/fa/DM00314099/files/DM00314099.pdf/jcr:content/translations/en.DM00314099.pdf.

[33] А. Самоделов, «Криптография в отдельном блоке: криптографический сопроцессор семейства STM32F4xx», *Новости Электроники*, № 6 (108), с. 12-25, 2012.

[34] Я. Совин, Ю. Наконечний, М. Стахів, «Дослідження характеристик вбудованого генератора випадкових чисел мікроконтролерів родини STM32F4XX згідно з методикою NIST STS», *Вісник НУ «Львівська політехніка». Серія «Автоматика, вимірювання та керування»*, № 753, с. 37-44, 2013.

УДК 004.056:061.68 (045)

Совын Я.Р., Наконечный Ю.М., Опирский И.Р., Стахив М.Ю. Анализ аппаратной поддержки криптографии в устройствах Интернета вещей

Аннотация. *В данной статье проанализированы характеристики и функциональные возможности встроенных криптоакселераторов в 8/16/32-битных микроконтроллерах общего назначения, призванных адаптировать традиционную криптографию к требованиям устройств Интернета вещей. Установлено, что традиционные криптоалгоритмы и протоколы, применяемые в сети Интернет при программной реализации не соответствуют требованиям, предъявляемым к устройствам Интернета вещей. Показано тенденции развития легковесной криптографии и криптоакселераторов в микроконтроллерах с точки зрения баланса безопасности, стоимости и производительности. Оценен выигрыш в производительности при применении криптоакселераторов для шифрования, хеширования и генерации случайных чисел по сравнению с оптимизированными программными реализациями. Обращается внимание на методы защиты криптоакселераторов от атак по побочным каналам, в первую очередь атак на энергопотребление, представляющих главную опасность.*

Ключевые слова: *криптоакселераторы, IoT, микроконтроллеры, встроенные системы, шифрование, хеш, ГСЧ.*

Sovyn Ya., Nakonechny Yu., Opirskyy I., Stakhiv M. Analysis of hardware support of cryptography in Internet of Things-devices

Abstract. *This article analyzes the features and functionality of embedded cryptographic accelerators in 8/16/32-bit general purpose microcontrollers designed to adapt traditional cryptography to the requirements of IoT-devices. It is established that traditional*

cryptographic algorithms and protocols used on the Internet in the case of software implementation do not meet the requirements of things related to –devices, the speed, the amount of memory required, and power consumption. The tendencies of development of light weight cryptography and cryptoaccelerators in microcontrollers from the point of view of balance of safety, cost and productivity are shown. The performance gain in the use of cryptographic accelerators for encryption, hashing and generation of random numbers in comparison with optimized software implementations is estimated. In particular, it is noted that the use of cryptographic accelerators allows to raise the speed of AES encryption 10-20 times for 8/16-bit processors and up to 150 times for 32-bit compared with software implementations of the algorithm. The growth of the SHA-1, SHA-256 hash rate algorithms in 32-bit microcontrollers is more than 100 times faster, and the HMAC is approaching 500. This allows 32-bit processors to use traditional cryptographic algorithms and protocols without significant constraints. It has also been shown that 32-bit microcontrollers have a trend towards the implementation of comprehensive security solutions that not only accelerate a wide range of symmetric and asymmetric algorithms and protocols, but also provide the ability to securely store and generate keys, securely download and update code, support digital signatures, and certificates. It is noted that manufacturers of microcontrollers are increasingly forced to pay attention to physical and algorithmic methods of protecting cryptographic accelerators from attacks through side-channels, in the first place attacks of analysis of power consumption, which constitute the main danger to devices of the Internet of things.
Key words: *cryptoaccelerators, IoT, microcontrollers, embedded systems, encryption, hash, RNG.*

Отримано *06 лютого 2018 року*, затверджено редколегією *04 березня 2018 року*