УДК 004.056:061.68

Я.Р. Совин, В.І. Отенко, Є.Ф. Штефанюк

ЕФЕКТИВНА РЕАЛІЗАЦІЯ АЛГОРИТМУ БЛОКОВОГО СИМЕТРИЧНОГО ШИФРУВАННЯ ДСТУ 7624:2014 («КАЛИНА») ДЛЯ 8/16/32-БІТОВИХ ВБУДОВАНИХ СИСТЕМ

У роботі виконано оцінку швидкодії та розміру коду для нового національного стандарту шифрування ДСТУ 7624:2014 «Калина» при реалізації на 8/16/32-бітових мікроконтролерах загального призначення, проведено порівняння одержаних показників з іншими сучасними алгоритмами з точки зору використання у вбудованих системах. Показано шляхи ефективної програмної реалізації шифру «Калина» в координатах швидкодія/пам'ять.

Ключові слова: ДСТУ 7624:2014, Калина, БСШ, вбудовані системи, швидкодія.

Вступ

Блокові симетричні шифри (БСШ) є основним криптографічним засобом гарантування конфіденційності при обробці інформації в інформаційно-телекомунікаційних системах. Також БСШ використовуються в таких криптографічних примітивах, як генератори випадкових чисел, функції гешування, коди автентифікації повідомлення тощо. На сьогодні відомо багато алгоритмів БСШ з різними принципами побудови (мережа Фейстеля, SPN, схема Лай-Мессі, ARX), рівнями криптографічної стійкості і параметрами шифру (довжина ключа, блоку, число раундів), складністю імплементації (класичні та lightweight-алгоритми), деякі з них затверджені в якості національних і міжнародних стандартів [1].

Крім забезпечення високої стійкості для БСШ також важливі висока швидкодія та малі вимоги до ресурсів при реалізації на широкому спектрі програмно-апаратних платформ. Особливо це актуально для вбудованих систем (Embedded Systems), де ціна та витрати енергії виходять на перший план, а обчислювальна потужність сконцентрована у недорогих центральних процесорах у складі мікроконтролерів загального призначення. Такі системи знаходять все ширше застосування при побудові безпровідних сенсорних мереж (БСМ), індустріальних, споживчих, медичних, автомобільних та кіберфізичних систем, IoT-пристроїв, інтелектуальних карт, OTP-токенів, систем охоронно-пожежної сигналізації, безпеки і контролю доступу, систем промислово-побутової автоматизації і моніторингу, wearable-електроніки (фітнес-трекерів, розумних годинників, окулярів) тощо.

Необхідність захисту інформації у вбудованих системах (ВС) привела до інтенсивних досліджень шляхів ефективної реалізації криптоалгоритмів, за умови обмежень, які накладаються цими системами. Ресурси вбудованих систем обмежені продуктивністю процесорного ядра, споживаною потужністю, розміром доступної постійної та оперативної пам'яті [2, 3]. Розмір коду програми безпосередньо впливає на вартість мікропроцесора, який переважно є найдорожчим компонентом системи. Час виконання є критичним з огляду на енергоспоживання, оскільки, як правило, у ВС центральний процесор більшу частину часу знаходиться в режимі пониженого енергоспоживання, виходячи з нього лише на короткий час для збирання, обробки та передачі інформації. Відповідно, час виконання криптографічного алгоритму прямо пропорційний споживаній потужності пристрою.

З появою нового національного стандарту БСШ ДСТУ 7624:2014 актуальною задачею є його оцінка з точки зору перспективи реалізації на найпоширеніших 8/16/32-бітових мікроконтролерних ядрах загального призначення.

Аналіз останніх досліджень і публікацій

В Україні з 1990 р. в якості БСШ використовувався алгоритм ГОСТ 28147-89 (офіційна назва ДСТУ ГОСТ 28147:2009) з розміром блоку і ключа 64 та 256 біт відповідно [4]. Хоча він все ще забезпечує практичну стійкість, для нього вже відомі теоретичні методи криптоаналізу, зі складністю істотно меншою, ніж повний перебір ключів [5]. Бурхливий розвиток інформаційно-телекомунікаційної сфери та, як наслідок, збільшення обсягів

оброблюваної інформації призвели до того, що 64-бітова довжина вхідного блоку виявилася недостатньою для обробки великих обсягів інформації. Як результат, цей стандарт вже виведений з дії в Білорусі і замінений в РФ (в якості основного тепер використовується 128-бітовий шифр «Кузнечик»).

З точки зору продуктивності, ГОСТ 28147-89 поступається сучасним аналогам, таким, як AES [6], проте заміна ГОСТ 28147-89 на міжнародний стандарт AES не була б ефективним рішенням для України, оскільки світові тенденції свідчать про початок поступової відмови від цього шифру, як на рівні вибору перспективних рішень в міжнародних криптографічних конкурсах, так і в прикладних системах [7]. Зокрема, деякі компанії, лідери IT-індустрії, такі як Google, вже застосовують нові алгоритми замість AES.

З 01.07.2015р. в Україні введено в дію національний криптографічний стандарт блокового симетричного перетворення ДСТУ 7624:2014, що визначає шифр «Калина» [8]. Новий національний стандарт підтримує розмір блоку і довжину ключа шифрування 128, 256 і 512 біт, забезпечуючи нормальний, високий та надвисокий рівень стійкості (зараз це єдиний у світі стандарт блокового шифрування, що підтримує 512-бітові симетричні ключі).

Проаналізуємо доступну інформацію щодо продуктивності цього шифру.

У статті [9] проведено теоретичний розрахунок кількості необхідних процесорних інструкцій для обробки одного байту даних, який вказує на перевагу шифру «Калина» (див. табл. 1). В якості обчислювальної платформи виступав 64-бітовий мікропроцесор загального призначення з архітектурою x86-64 від фірми Intel.

Таблиця 1

Теоретичний розрахунок кількості необхідних процесорних інструкцій для обробки одного байту різними шифрами [9]

	Симетричний блоковий шифр			
	Калина (128/128)	**ГОСТ 28147**	**СТБ 34.101.31**	**AES**
Кількість операцій на 1 байт	40,375	72	40,5	45,375

У роботах [7, 9] наведені результати порівняння продуктивності, що проводилося для шифру «Калина» (всі комбінації розміру блоку і довжини ключа), AES-128, AES-256 (національний стандарт США і найпоширеніший алгоритм у світі), ГОСТ 28147-89 (попередній стандарт в Україні та РФ), СТБ 34.101.31-2011 («BelT», національний стандарт Білорусі) і алгоритму «Кузнечик» (новий стандарт РФ) [10] в однакових умовах роботи (розмір блоку даних 1 Гбайт, режим простої заміни ECB, багатократне шифрування одного блоку).

Для отримання найбільшої швидкодії апаратно-незалежної реалізації була обрана мова програмування C++, використаний компілятор gcc version 4.9.2, тестування виконувалось на комп'ютері під управлінням 64-бітової ОС Linux (Ubuntu) з процесором Intel Core i5-4670 на тактовій частоті 3.40 ГГц. При програмній реалізації AES не використовувався набір інструкцій AES-NI.

Результати тестування швидкодії програмної реалізації [11] для версій із найкращою оптимізацією компілятора (-O3 –m64) наведені на рис. 1.

Автори статті прийшли до наступних висновків щодо 64-бітової платформи:

– для 128-бітової довжини ключа швидкодія «Калини» вища за AES на 3% (86 Мбіт/с);

– для 256-бітової довжини ключа швидкодія «Калини» повільніша за AES на 10% (для 128-бітового блоку) та швидше на 1% (для 256-бітового блоку);

– швидкодія «Калини» при відповідній довжині ключа вища за ГОСТ 28147-89 у 2,8 рази (для 128-бітового блоку) і 3,16 рази (для 256-бітового блоку), і приблизно у 2 рази вища, ніж у нових стандартів шифрування Білорусії і РФ.

Як свідчать публікації [7, 9, 11] шифр «Калина» орієнтований на досягнення високої продуктивності на 64-бітових сучасних мікропроцесорах загального призначення (Intel, AMD). Разом з тим, як декларують самі автори шифру, при розробці національного стандарту, що забезпечує високу стійкість і продуктивність, враховуючи відсутність в

Україні власного мікроелектронного виробництва та неможливість надійного контролю іноземного, вимоги до ефективної реалізації нового шифру в системах з обмеженими ресурсами розглядалися як другорядні [7].

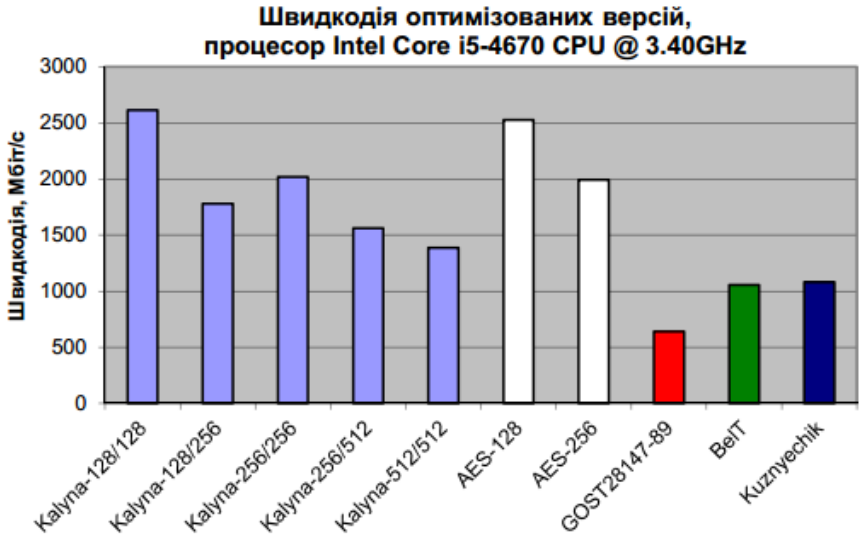

Рис. 1. Швидкодія оптимізованих версій програмної реалізації блокових шифрів [7]

З огляду на відносну новизну шифру «Калина» нам невідомі публікації, які б стосувалися оцінки швидкодії цього шифру для вбудованих систем, а отже, це питання потребує подальшого дослідження.

Мета статті

Метою статті є дослідити шляхи ефективної програмної реалізації БСШ ДСТУ 7624:2014 на найпоширеніших 8/16/32-бітових процесорних платформах, оцінити швидкодію і вимоги до пам'яті одержаних реалізацій та провести порівняння з аналогічними показниками актуальних блокових шифрів, таких як AES, ГОСТ 28147-89, «Кузнечик», що дозволить вибирати оптимальне рішення при розробленні механізмів захисту у вбудованих системах.

Структурні особливості шифру «Калина»

Національний стандарт шифрування ДСТУ 7624:2014 (Калина) [8] належить до SPN, байт-орієнтованих шифрів. Основні параметри шифру, такі як довжина ключа k і блоку даних l, кількість раундів t та кількість стовпців матриці стану c пов'язані залежностями представленими в табл. 2. Розмір блоку і довжина ключа використовуються у позначенні шифру у форматі Калина-l/k.

Таблиця 2

Основні параметри шифру «Калина»

Довжина ключа k, біт	**Довжина блоку l, біт**	**Кількість раундів t**	**Кількість стовпців матриці стану c**
128, 256	128	10	2
256, 512	256	14	4
512	512	18	8

При виконанні зашифрування або розшифрування операції виконуються над двовимірним масивом байт, названим поточним станом шифру (*State)*. Поточний стан шифру можна представити у вигляді матриці розмірністю $8 \times c$ байтів (вісім рядків по c байт): $State = (s_{i,j})$, де $i = 0...7$, $j = 0...c-1$.

В алгоритмі використовуються операції арифметичного додавання ($\boxplus$) та віднімання ($\boxminus$) за модулем 2^{64}, додавання за модулем 2 ($\oplus$), табличної заміни (*SubBytes*, *InvSubBytes*), циклічного зсуву рядків (*ShiftRows*, *InvShiftRows*) та лінійного перетворення (*MixColumns*, *InvMixColumns*). Структура алгоритму «Калина» представлена на рис. 2 [12].

Розглянемо дещо детальніше кожну з цих операцій.

Операції *додавання* та *віднімання* – реалізують арифметичне додавання або віднімання стовпців матриці стану *State* і стовпців циклового підключа за модулем 2^{64}. Числа в стовпцях вважаються представленими у форматі little-endian, тобто менш значущі байти мають менший індекс.

Операція *додавання за модулем 2* – виконує додавання за модулем 2 матриці стану *State* і циклового підключа.

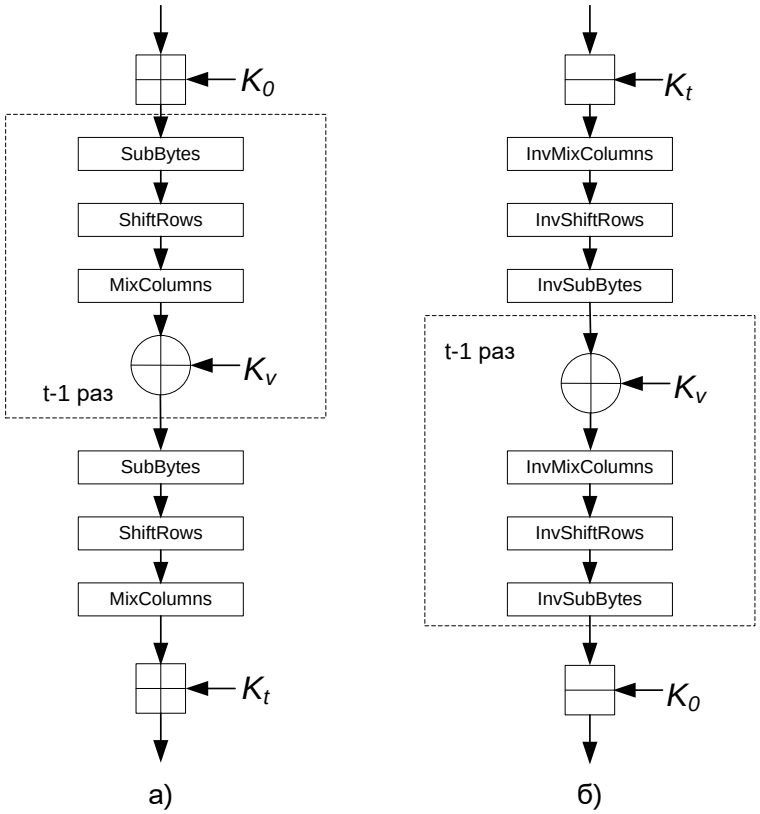

Рис. 2. Структурна схема алгоритму «Калина» в режимах зашифрування (а) і розшифрування (б)

Операції *SubBytes* та *InvSubBytes* – виконують підстановку кожного байту матриці стану на відповідний йому байт з однієї з чотирьох таблиць заміни *S0-S3* та *inv_S0-inv_S3* для операцій зашифрування і розшифрування відповідно. Кожна таблиця має розмір 256 байт. Номер таблиці заміни визначається як індекс рядку байту за модулем 4 ($i \bmod 4$): $s_{i,j} = S_{i \bmod 4}(s_{i,j})$ або $s_{i,j} = inv\_S_{i \bmod 4}(s_{i,j})$.

Операції *ShiftRows* та *InvShiftRows* – здійснюють циклічний зсув байтів рядків вправо чи вліво відповідно. Кількість позицій δ_i, на яку зсувається рядок, залежить від номера рядку *i* та довжини блоку *l* і обчислюється за формулою: $\delta_i = \left[\frac{i \cdot l}{512}\right]$.

Операції *MixColumns* та *InvMixColumns* – здійснюють перетворення стовпців матриці стану шляхом виконання операцій множення і додавання в скінченному полі $GF(2^8)$ за модулем незвідного многочлена $\psi = x^8 + x^4 + x^3 + x^2 + 1$. Кожен елемент результуючої

матриці стану $W = (w_{i,j})$ обчислюється в полі $GF(2^8)$ як скалярний добуток рядка матриці v *(inv_v)* на стовпець матриці стану *State* відповідно до формул:

$$w_{i,j} = (v >>> i) \otimes S_j,$$
$$w_{i,j} = (inv\_v <<< i) \otimes S_j,$$

де v = *(0x01, 0x01, 0x05, 0x01, 0x08, 0x06, 0x07, 0x04)*, inv_v = *(0xAD, 0x95, 0x76, 0xA8, 0x2F, 0x49, 0xD7, 0xCA)*; S_j – j-й стовпець матриці стану *State*; $v >>> i$ та $v <<< i$ – операції циклічного зсуву байт вектора v вправо і вліво на i позицій відповідно.

Для одержання раундових підключів з вихідного майстер-ключа використовується процедура розгортання ключів, в якій задіяні стандартні перетворення, розглянуті вище.

Архітектурні особливості мікроконтролерів для реалізації шифру «Калина»

Для подальших досліджень реалізації шифру «Калина» у ВС на базі 8/16/32-бітових мікроконтролерів (МК) нами було обрано по одній найпоширенішій на ринку архітектурі.

AVR-мікроконтролери (8-бітові). В якості 8-бітової платформи ми розглядаємо родину мікроконтролерів AVR (фірма Atmel). Цей вибір обумовлений вдалою системою команд цих МК, що орієнтована на максимальну ефективність виконання програм, написаних на мовах високого рівня.

Серед особливостей AVR-ядра, важливих в контексті криптографії для вбудованих систем варто відзначити, що пам'ять має Гарвардську організацію з 8-бітовою пам'яттю даних типу SRAM та 16-бітовою пам'яттю програм типу Flash. Регістровий файл містить 32 регістри загального призначення (РЗП) безпосередньо підключених до АЛП. Усі мікроконтролери родини AVR містять однакове процесорне RISC-ядро, що зображено на рис. 3.а. Система команд достатньо розвинена і складається з понад 130 інструкцій, більшість з яких завдяки Гарвардській архітектурі та дворівневому конвеєру виконуються за один такт [13].

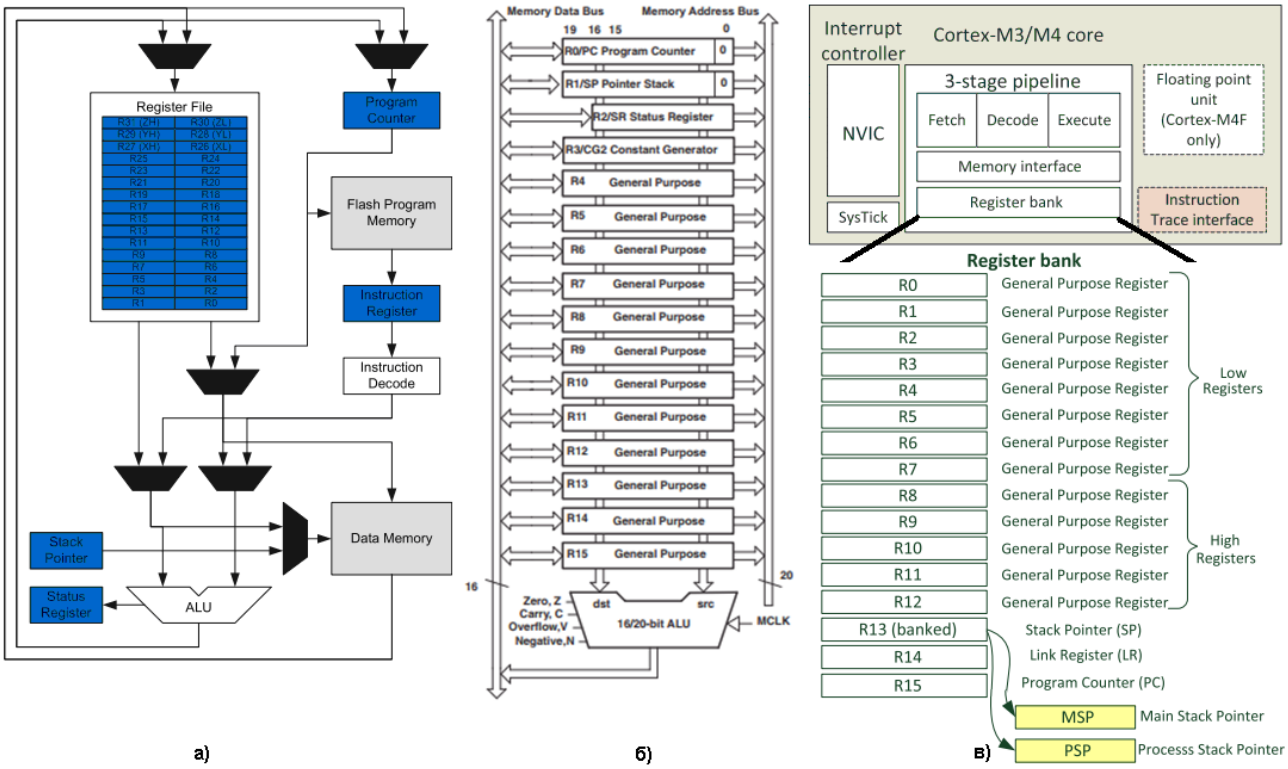

Рис. 3. Архітектура центрального процесора мікроконтролерів AVR (а), MSP430 (б) та ARM Cortex-M3 (в)

AVR-мікроконтролери підтримують безпосередню, пряму та непряму адресації. Наявність режимів предекременту та постінкременту при непрямій адресації дає змогу ефективно обробляти масиви даних в процесі виконання криптоалгоритму, генеруючи компактний програмний код. Для доступу до даних у Flash-пам'яті (S-Box, Look-Up таблиці, ключі) використовується непряма адресація.

MSP430-мікроконтролери (16-бітові). Для тестування швидкодії шифру «Калина» на 16-бітових платформах обрано МК родини MSP430 (фірма Texas Instruments), як одні з найпопулярніших у своєму сегменті. Однією з ключових переваг родини MSP430 є ультранизьке енергоспоживання, що забезпечило їм широку популярність у вбудованих системах і особливо у БСМ.

Компактне високопродуктивне 16-бітове RISC-ядро MSP430 побудоване за Прінстонською архітектурою з єдиним адресним простором для команд і даних та містить 16 регістрів, з яких дванадцять (R4-R15) є регістрами загального призначення (рис. 3.б). Регістри R0-R3 – виконують спеціальні функції (Program Counter, Stack Pointer, Status Register, Constant Genereator). Набір команд дуже простий і представлений 27 оригінальними і 24 емульованими інструкціями, які оптимізовані для ефективного використання мовами програмування високого рівня. Всі команди 16-бітні і можуть обробляти як 8- так і 16-бітові операнди. Підтримується сім режимів адресації [14].

Flash-пам'ять може використовуватися як для зберігання коду програми так і для даних, що виключає необхідність копіювати дані в ОЗП перед подальшим використанням. Завдяки однотактовим регістровим операціям та ортогональній архітектурі забезпечується компактність коду та висока продуктивність. Важливою в контексті криптографії є також така особливість процесора MSP430 як прямий обмін даними між комірками пам'яті, минаючи регістри.

ARM Cortex-M3-мікроконтролери (32-бітові). Реалізацію на 32-бітовій платформі виконано на основі процесора ARM Cortex (фірма ARM), оскільки МК з ARM-ядром становлять до 90% ринку 32-бітових RISC-мікроконтролерів і наразі за енергоефективністю та ціною наблизилися до 8-бітових моделей, складаючи останнім серйозну конкуренцію у їх традиційних сегментах використання.

Процесор ARM Cortex випускається в трьох прикладних профілях:

Cortex A – процесори, призначені для використання в поєднанні зі складними операційними системами для запуску різноманітних додатків;

Cortex R – процесори, призначені для використання у системах реального часу;

Cortex-M – мікроконтролерний профіль, орієнтований на застосування у вбудованих системах.

З огляду на тематику статті, ми розглядатимемо останній профіль у варіанті ARM Cortex-M3.

ARM Cortex-M3 є 32-бітовим процесором на основі Гарвардської архітектури з трирівневим конвеєром, який реалізує системи команд Thumb та Thumb-2. Ядро Cortex-M3 містить 16 регістрів R0-R15, з яких регістри R0-R12 є регістрами загального призначення. Регістр R13, який є покажчиком стеку, насправді складається з двох регістрів, і в кожен момент часу доступний лише один з них. Основний покажчик стеку MSP використовується ядром операційної системи і обробниками виняткових ситуацій, а покажчик стеку процесу PSP використовується прикладною програмою. У регістрі R14 зберігається адреса повернення при виклику підпрограми. Лічильник команд R15 містить адресу виконуваної команди (рис. 3.в). Арифметично-логічний пристрій має 32-бітовий блок зсуву, який дозволяє одночасно з виконанням операції здійснювати зсув одного з операндів [15].

Ядро підтримує два рівні доступу до коду програми (привілейований і користувацький), що забезпечують безпечне звернення до критичних областей пам'яті, а також реалізують базову модель механізму захисту. У ядрі використовується фіксований розподіл адресного простору.

Загальні підходи до проведення досліджень

Оскільки шифр «Калина» має багато варіантів розміру блоку і ключа, то для зручності порівняння з іншими алгоритмами дослідження проводилися на розмірах блоку/ключа 128/128 біт та 128/256 біт відповідно. Таким чином, аналізувалися наступні алгоритми з параметрами представленими в табл. 3.

Таблиця 3

Параметри досліджуваних БСШ

Алгоритм	**Довжина блоку/ключа, біт**	**Кількість раундів**
«Калина»	128/128	10
	128/256	14
AES	128/128	10
	128/256	14
ГОСТ 28147-89	64/256	32
«Кузнечик»	128/256	10

Для кожного з типів мікроконтролерів: AVR (8 біт), MSP430 (16 біт), ARM Cortex-M3 (32 біти) ці алгоритми були реалізовані на мові С з допомогою останніх версій інтегрованих середовищ розробки IAR Embedded Workbench for AVR (v6.70), IAR Embedded Workbench for MSP430 (v6.40) та IAR Embedded Workbench for ARM (v7.60) відповідно. У цих середовищах також здійснено оцінку кількості тактів і розміру коду.

Параметрами, які вимірювалися були: **швидкодія зашифрування/розшифрування**, виражена в тактах/байт і усереднена для 100 блоків, **розмір постійної пам'яті** (*Flash*), який складається з розміру самої програми та таблиць розташованих у Flash-пам'яті, і **розмір оперативної пам'яті** (*SRAM*), що задається таблицями в SRAM та розміром стеку.

Оскільки в пристроях різного призначення основними вимогами до криптоалгоритму можуть бути як економне використання пам'яті, так і висока швидкодія, то доцільно дослідити алгоритми під цим кутом зору. Для досягнення максимальної швидкодії програми слід оптимізувати ті частини алгоритму, які є найскладнішими в обчислювальному плані, використавши для них команди та способи адресації, що потребують мінімальної кількості тактів МК. Оскільки операції додавання та сумування за модулем два є по-суті атомарними і реалізуються з допомогою відповідних команд процесора, то швидкодія алгоритмів в цілому буде визначатися швидкістю виконання лінійного перетворення.

Для алгоритмів «Калина», AES та «Кузнечик» було запропоновано декілька шляхів керування ефективністю шифрування, які в першу чергу відрізняються підходами до виконання найскладнішої в обчислювальному плані операції множення в полях Галуа у функції лінійного перетворення *MixColumns*. На цій підставі були виділені наступні профілі:

- **Профіль *SOFT***. Програмна реалізація операції множення в полях Галуа – ця версія програми орієнтована на досягнення мінімального розміру коду за рахунок низької швидкодії.
- **Профіль *FAST***. Це теж програмна реалізація операції множення в полях Галуа, проте використовуються різні евристичні прийоми для її пришвидшення і ефективної реалізації операції *MixColumns* – ця версія програми збалансована між швидкістю та розміром коду.
- **Профіль *LOCK-UP***. Тут використовується таблична реалізації самої операції множення – ця версія програми орієнтована на досягнення високої швидкодії при помірному зростанні об'єму коду (за рахунок передобчислених таблиць).
- **Профіль *MDS***. Використовуються передобчислені таблиці для операцій множення та нелінійної заміни. Ця версія орієнтована на максимальну швидкодію за рахунок суттєвого зростання розміру коду. Суть даної методики – об'єднання операцій *SubBytes*, *ShiftRows* та *MixColumns* в одну з використанням обчислених наперед таблиць. Для операції розшифрування використовується еквівалентний до

зашифрування алгоритм. Для забезпечення симетрії процедур зашифрування та розшифрування необхідною є попередня обробка раундових ключів при операції розшифрування.

Щоб збільшити швидкодію для всіх профілів реалізацій і архітектур МК ми намагалися по можливості розташовувати проміжні значення та стан шифру в регістрах загального призначення. Також використовувалися такі прийоми як розгортання циклу, використання макросів замість функцій, врахування симетрії у передобчислених таблицях та ін.

При вимірюванні швидкодії шифрування використовувалися наперед згенеровані раундові підключі, які зберігалися в ОЗП. Це досить реалістичний підхід для вбудованих систем, що переважно використовують або один секретний ключ впродовж усього життєвого циклу пристрою або сесійні ключі з досить тривалими сеансами обміну даними. Разом з тим, при оцінці об'єму пам'яті, функції розгортання ключа враховувалися.

Алгоритм ГОСТ 28147-89 не містить операції множення в полях Галуа, тому для нього було виділено: профіль ***SOFT***, в якому використовуються 4-бітові вузли заміни; профіль ***FAST***, в якому використовуються 8-бітні вузли заміни; профіль ***MDS***, в якому використовуються 8-бітні вузли заміни та табличне представлення операції нелінійної заміни і циклічного зсуву на 11 біт.

Для мікроконтролерів AVR кількість тактів для зчитування даних з Flash- і SRAM-пам'яті відрізняється (3 проти 2 відповідно). Тому залежно від того, в якій області пам'яті розташовуються таблиці, ми будемо мати різні результати за швидкодією. Через це для профілів націлених на високу швидкодію, таких як ***FAST***, ***LOCK-UP*** і ***MDS,*** таблиці по можливості розташовуються в SRAM. Для MSP430 і ARM Cortex-M3 такої залежності немає і результати будуть ідентичні. Враховуючи, що оперативна пам'ять є, як правило, більш обмеженим і цінним ресурсом ніж постійна, то всі таблиці для цих архітектур розташовувалися у Flash-пам'яті.

Результати дослідження реалізацій криптоалгоритмів БСШ

Вимірювані параметри реалізацій досліджуваних БСШ зібрані в табл. 4.

Таблиця 4

Параметри програмних реалізацій досліджуваних БСШ

Алгоритм	**Швидкодія зашифрування/розшифрування, тактів/байт**				**Пам'ять Flash (ROM)/SRAM (RAM), Кбайт**			
	Soft	**Fast**	**Luck-up**	**MDS**	**Soft**	**Fast**	**Luck-up**	**MDS**
CPU AVR (8-біт)								
AES-128/128	569/1034	221/340	206/338	240/238	1,5/0,21	3,8/0,95	4,7/2,19	13,0/8,71
AES-128/256	797/1469	308/479	286/477	334/331	1,6/0,28	3,9/1,01	4,8/2,25	12,9/8,77
GOST 28147-64/256	585/585	262/283	–	359/359	0,5/0,05	2,3/1,05	–	4,7/4,04
Kalyna-128/128	2335/7983	439/843	449/624	390/411	3,5/0,28	7,3/2,30	17,0/5,04	23,6/15,26
Kalyna-128/256	3266/11165	608/1173	621/865	539/564	3,5/0,36	7,7/2,37	17,1/5,11	23,7/15,33
Kuznechik-128/256	8983/9103	903/901	698/712	1022/1018	1,2/0,26	2,8/0,73	8,7/2,48	130,0/0,48
CPU MSP430 (16-біт)								
AES-128/128	782/3724	284/398	241/229	140/141	1,6/0,21	3,4/0,23	3,8/0,20	11,5/0,22
AES-128/256	1202/5450	401/566	339/323	195/196	1,6/0,28	3,4/0,29	3,9/0,27	11,6/0,28
GOST 28147-64/256	499/495	279/294	–	229/239	0,5/0,06	2,0/0,08	–	5,0/0,08
Kalyna-128/128	2616/9796	482/1010	388/469	266/283	3,5/0,29	5,5/0,29	6,8/0,29	51,5/0,28
Kalyna-128/256	3360/13697	673/1411	540/654	372/386	3,5/0,37	5,6/0,37	6,9/0,37	51,8/0,36
Kuznechik-128/256	9876/9926	1243/1269	807/813	555/615	1,3/0,27	2,3/0,27	7,8/0,24	129,4/0,25
CPU ARM Cortex-M3 (32-біт)								
AES-128/128	166/321	101/248	101/189	63/63	1,3/0,23	1,9/0,23	3,2/0,22	10,8/0,22

AES-128/256	234/459	141/354	141/269	87/87	1,4/0,30	2,0/0,29	3,3/0,28	10,9/0,28
GOST 28147-64/256	267/267	89/89	–	92/89	0,3/0,11	1,9/0,07	–	6,9/0,11
Kalyna-128/128	2221/7944	183/349	291/409	88/93	3,1/0,30	4,7/0,31	7,4/0,31	53,0/0,28
Kalyna-128/256	3109/11111	254/488	407/572	122/126	3,2/0,37	4,8/0,39	7,5/0,39	52,9/0,36
Kuznechik-128/256	7440/7463	690/673	443/436	126/168	1,3/0,28	2,5/0,24	7,2/0,24	129,9/0,24

Для кращого сприйняття і порівняння результатів, представлених в табл. 4 на рис. 4-6 зображені профілі з максимальною швидкодією шифрування, але представленою не у тактах на байт, а у кількості байт, шифрованих за 10000 тактів, і відповідно більша швидкодія відповідає більшому значенню на графіку.

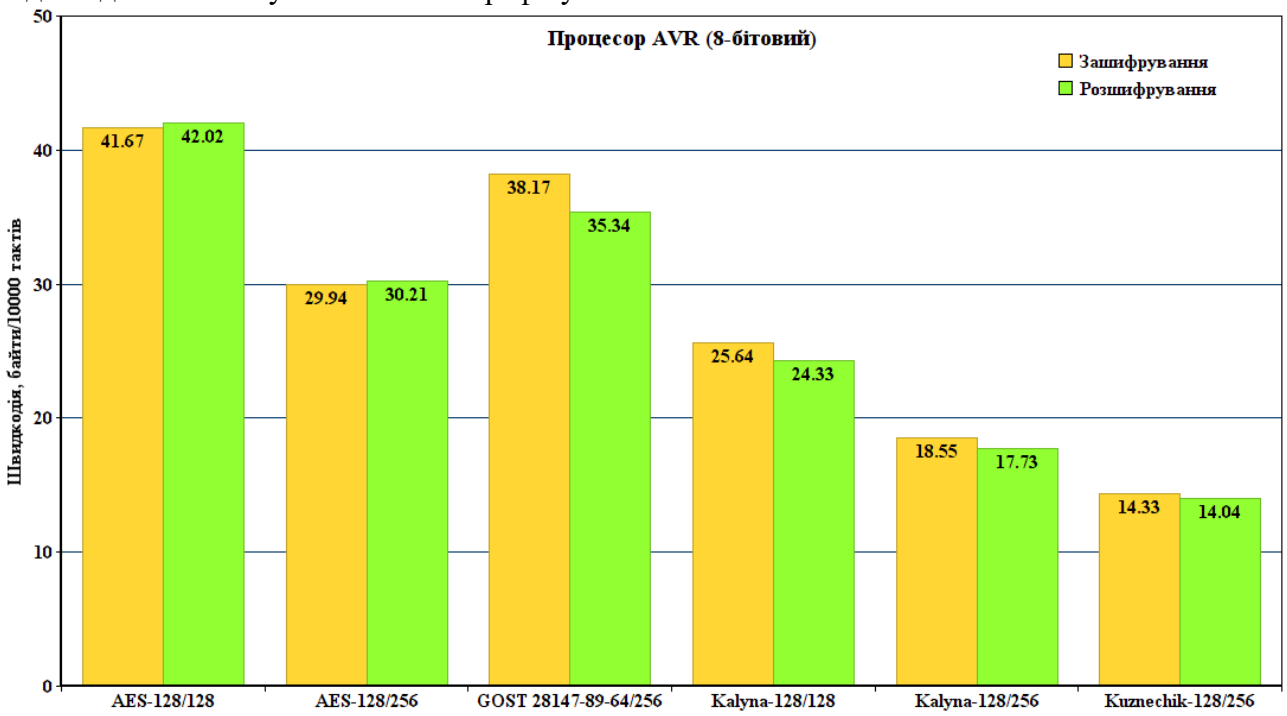

Рис. 4. Порівняльний аналіз швидкодії БСШ на процесорі AVR (8-бітовий)

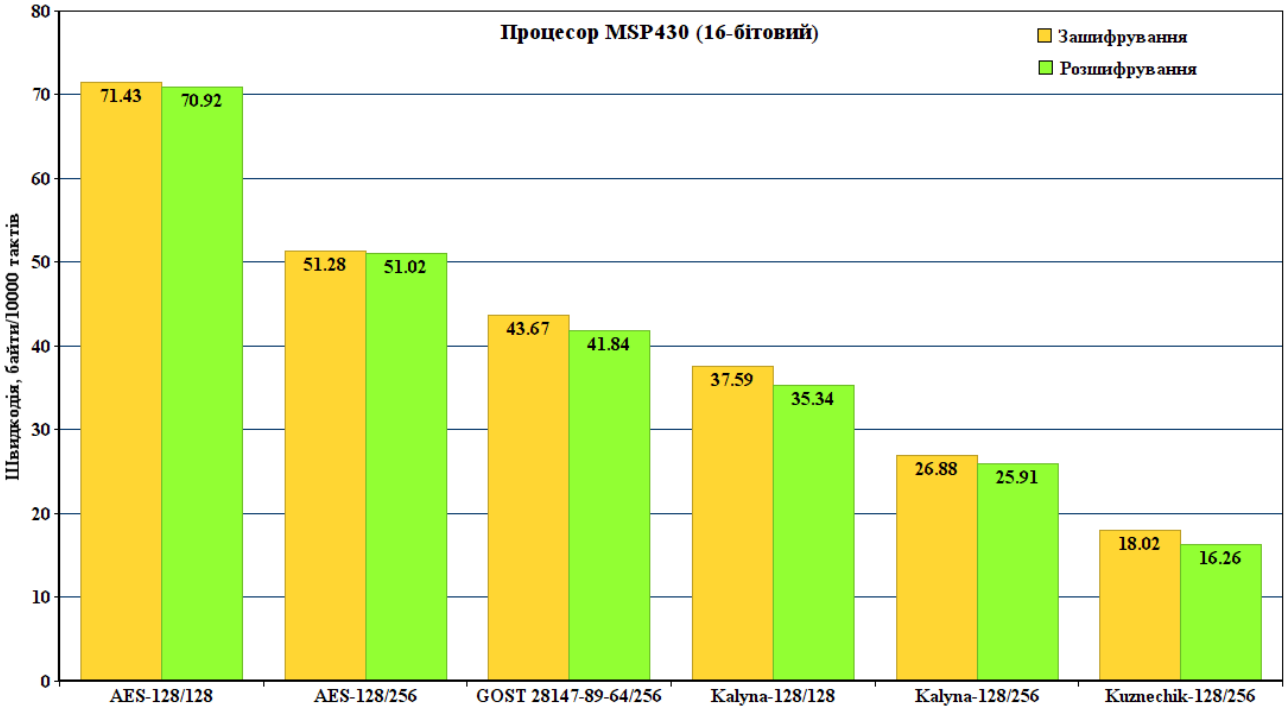

Рис. 5. Порівняльний аналіз швидкодії БСШ на процесорі MSP430 (16-бітовий)

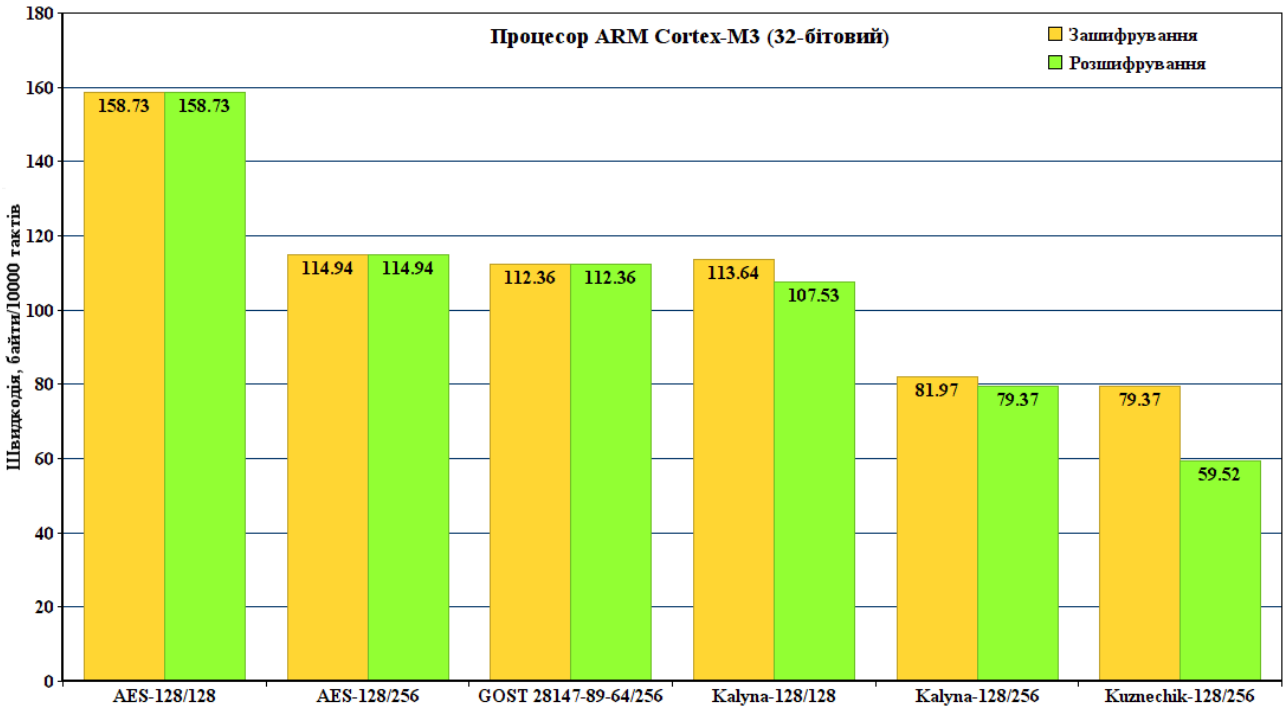

Рис. 6. Порівняльний аналіз швидкодії БСШ на процесорі ARM Cortex-M3 (32-бітовий)

З табл. 4 випливає, що для всіх архітектур і профілів реалізації шифр «Калина» поступається AES та ГОСТ 28147-89 як за швидкодією, так і за вимогами до пам'яті, проте випереджає «Кузнечик».

Для 8-бітових процесорів на максимальній швидкодії шифр «Калина» повільніший за AES приблизно у 1,6 раз, а за ГОСТ 28147-89 до 2 разів, для 16-бітових у 2 рази та до 1,6 раз, а для 32-бітових у 1,4 раз відповідно. Для 32-бітових процесорів «Калина» наближається впритул за швидкодією до AES-128/256 та ГОСТ 28147-89.

Для 8-бітового профілю з максимальною швидкодією ***MDS*** шифр «Калина» вимагає 24 і 15 Кбайт ПЗП і ОЗП відповідно, що є досить відчутним для такого класу вбудованих систем, і потребує використання мікроконтролерів верхнього цінового діапазону. Для 16- та 32-бітових профілів ***MDS*** потрібно порядку 52-54 Кбайт ПЗП, що хоча не є критичним, проте і тут значно обмежує вибір моделей МК. Якщо максимальна швидкодія не потрібна, то вимоги до Flash-пам'яті в «Калина» є прийнятними і співмірними з іншими алгоритмами БСШ.

Відзначимо також, що шифр «Кузнечик» з вимогами 130 Кбайт ПЗП робить досить проблематичною його ефективну в обчислювальному плані реалізацію у вбудованих системах, особливо 8- і 16-бітових.

Висновки

Результати досліджень показали, що для ВС, у яких достатньо нормального і високого рівня стійкості, алгоритм «Калина» поступається AES та ГОСТ 28147-89 як за швидкодією, так і за вимогами до пам'яті. Особливо доцільно застосовувати алгоритм ГОСТ 28147-89 замість «Калини» за жорстких вимог щодо об'єму коду. Фактично підтверджено тезу самих розробників шифру про орієнтованість «Калини» на 64-бітні високопродуктивні мікропроцесори загального призначення, а не системи з обмеженими ресурсами.

Разом з тим, алгоритм «Калина» більш захищений від атак на реалізацію, в першу чергу таких як аналіз енергоспоживання, що дуже характерні та небезпечні для ВС. Реалізація спеціальних заходів для протидії атакам через сторонні канали (Side-Channel атакам) в алгоритмах AES і ГОСТ 28147-89 буде приводити до зменшення швидкодії і паритету за цим показником з шифром «Калина».

Розглянуті методи реалізації БСШ ДСТУ 7624:2014 «Калина» на 8/16/32-бітових вбудованих платформах дають змогу досягнути компромісу між параметрами швидкодія/ціна/енергоспоживання в залежності від конкретного застосування.

Список використаної літератури

1. Панасенко С. П. Алгоритмы шифрования. Специальный справочник / С. П. Панасенко. – СПб.: БХВ-Петербург, 2009. – 576 с.

2. A Survey of Lightweight Cryptography Implementations / [T. Eisenbarth, S. Kumar, C. Paar et al] // IEEE Design & Test of Computers – Special Issue on Secure ICs for Secure Embedded Computing. – 2007. – Vol. 24, Nr. 6. – pp. 522-533.

3. Performance Analysis of Contemporary Light-Weight Block Ciphers on 8-bit Microcontrollers / Rinne S., Eisenbarth T., Paar C. // ECRYPT Workshop Software Performance Enhancement for Encryption and Decryption. – 2007. – pp. 33-43.

4. Системы обработки информации. Защита криптографическая. Алгоритмы криптографического преобразования: ДСТУ ГОСТ 28147:2009. – [Чинний від 2009-02-01]. – К.: Держспоживстандарт України, 2008. – 28 с. – (Національний стандарт України).

5. Takanori I. A Single-Key Attack on the Full GOST Block Cipher / I. Takanori // Fast Software Encryption 18th International Workshop (FSE-2011), Springer LNCS 6733, 2011. – pp. 290-305.

6. FIPS-197: Advanced Encryption Standard (AES). Federal Information Processing Standard, National Institute of Standards and Technology, U.S. Dept. of Commerce, 2001. – 47 p.

7. Принципи побудови і основні властивості нового національного стандарту блокового шифрування України / [Р. Олійников, І. Горбенко, О. Казимиров та ін.] // Захист інформації. – 2015. – № 2(17). – С. 142-157.

8. ДСТУ 7624:2014. Інформаційні технології. Криптографічний захист інформації. Алгоритм симетричного блокового перетворення. [Текст]. – Введ. 01–07–2015. – К.: Мінекономрозвитку України, 2015.

9. Обґрунтування вимог, побудування та аналіз перспективних симетричних криптоперетворень на основі блочних шифрів / [О. О. Кузнецов, Р. В. Олійников, Горбенко Ю. І. та ін.] // Вісн. Нац. ун-ту "Львів. політехніка". – 2014. – № 806. – С. 124-140.

10. Национальный стандарт РФ ГОСТ Р 34.12-2015. Информационная технология. Криптографическая защита информации. Блочные шифры. М.: Стандартинформ, 2015. – 25 с.

11. R. Oliynykov, O. Kazymyrov, O. Kachko et al. Source code for performance estimation of 64-bit optimized implementation of the block ciphers Kalyna, AES, GOST, BelT, Kuznyechik. 2015 [Electronic resource] // Mode of access: www. URL: https://github.com/Roman-Oliynykov/ciphers-speed/ (20.06.2017).

12. A New Encryption Standard of Ukraine: The Kalyna Block Cipher / [R. Oliynykov, I. Gorbenko, O. Kazymyrov et al] // Norwegian Information Security Conference (NISK-2015). – 2015. – 113 p.

13. Евстифеев А. В. Микроконтроллеры AVR семейства Mega. Руководство пользователя / А. В.Евстифеев. – М.: Издательский дом «Додэка-XXI», 2007. – 592 с.

14. User's Guide. MSP430x5xx and MSP430x6xx Family / Texas Instruments. – Texas Instruments, 2014. – 1145 p.

15. Yiu J. The Definitive Guide to the ARM Cortex-M3 and ARM Cortex-M4 Processors / J. Yiu. – [Third Edition]. – Elsevier, 2014. – 1055 p.

Надійшла 01.05.2017 р. Рецензент: к.т.н., доц. Курченко О.А.