https://doi.org/10.31891/2219-9365-2025-84-39
УДК 621.391:004.056.5

РУДИЙ Роман
Чернівецький національний університет імені Юрія Федьковича
https://orcid.org/0009-0002-5161-5032
e-mail: rudyi.roman@chnu.edu.ua
ВОРОБЕЦЬ Георгій
Чернівецький національний університет імені Юрія Федьковича
https://orcid.org/0000-0001-8125-2047
e-mail: g.vorobets@chnu.edu.ua

МОДЕЛЮВАННЯ ТА ВЕРИФІКАЦІЯ ПАРАЛЕЛЬНИХ ПРОЦЕСІВ ОБРОБКИ І ДОСТУПУ ДО ДАНИХ У РЕКОНФІГУРОВАНИХ ПРИСТРОЯХ ЗАСОБАМИ МЕРЕЖ ПЕТРІ

У статті приведено результати досліджень питання організації високоефективного паралелізму при апаратній реалізації криптографічних алгоритмів, що є важливим при реалізації інфокомунікаційних систем регламентованого доступу до персональних даних користувачів. В якості потокового навантаження використано алгоритм AES у режимі лічильника (CTR). Такий підхід забезпечує незалежність блоків і природну придатність до масштабування кількості обчислювальних ядер пристроїв шифрування даних. Запропоновано формалізовану модель архітектури апаратної реалізації модуля шифрування цифрового потоку, яку описано засобами мереж Петрі з відображенням механізмів диспетчеризації задач, взаємного виключення та буферизації даних вхідного потоку. Для практичної оцінки масштабованості реалізовано програмні прототипи паралельного AES-CTR мовою C++ та проведено вимірювання часу, пропускної здатності й прискорення для різної кількості потоків і обсягів даних. Додатково виконано функціональну перевірку на реальних файлах.

Ключові слова: персоніфіковані дані, паралельні обчислення, AES-CTR, синхронізація даних, мережі Петрі, FPGA.

RUDYI Roman, VOROBETS Heorhii
Yuriy Fedkovych Chernivtsi National University

MODELING AND VERIFICATION OF PARALLEL DATA PROCESSING AND ACCESS IN RECONFIGURED DEVICES USING PETRI NETWORKS

The article presents the results of a comprehensive study devoted to the organization of highly efficient parallelism in the hardware implementation of cryptographic algorithms, which is a critical requirement for modern infocommunication systems with regulated and secure access to users' personal data. Special attention is paid to increasing throughput and scalability of encryption modules while maintaining strict security guarantees. As a representative cryptographic primitive, the Advanced Encryption Standard (AES) algorithm operating in counter mode (CTR) was selected and analyzed as a streaming workload. The choice of AES-CTR is justified by its inherent block independence, absence of data dependencies between encryption rounds of different blocks, and natural suitability for parallel execution and dynamic scaling of computing resources.

A formalized architectural model of a hardware-based digital stream encryption module is proposed. The model is described using Petri nets, which provide a rigorous and expressive means for representing concurrent processes, synchronization mechanisms, task dispatching, mutual exclusion, and buffering of input data streams. This approach allows the analysis of system behavior under various workloads and configurations, as well as the identification of potential bottlenecks and contention points in parallel execution.

To validate the proposed architectural solutions and assess their practical effectiveness, software prototypes of parallel AES-CTR implementations were developed in C++. Experimental evaluations were conducted to measure execution time, achieved bandwidth, and acceleration factors for different numbers of parallel streams and varying data volumes. The scalability characteristics of the solution were analyzed in detail, demonstrating the efficiency of parallel processing under increasing computational load. In addition, functional testing was performed using real files of different sizes and formats, confirming the correctness, robustness, and practical applicability of the proposed parallel encryption approach. The obtained results substantiate the feasibility of the developed model for high-performance cryptographic hardware systems.

Keywords: personalized data, parallel computing, AES-CTR, data synchronization, Petri nets, FPGA.

Стаття надійшла до редакції / Received 30.10.2025
Прийнята до друку / Accepted 29.11.2025

ПОСТАНОВКА ПРОБЛЕМИ У ЗАГАЛЬНОМУ ВИГЛЯДІ ТА ЇЇ ЗВ'ЯЗОК ІЗ ВАЖЛИВИМИ НАУКОВИМИ ЧИ ПРАКТИЧНИМИ ЗАВДАННЯМИ

Інтенсивна еволюція сучасних телекомунікаційних стандартів, зокрема мереж п'ятого покоління, технологій Інтернету речей та високошвидкісних інтерфейсів 100G Ethernet, супроводжується підвищенням вимог до швидкодії та мінімізації латентності систем захисту інформації. Особливої важливості у цьому контексті набувають питання швидкого доступу і захисту персональних даних користувачів [1, 2]. При цьому рівень конфіденційності, відповідно, і вимог до захисту широкого спектру даних різних користувачів може змінюватись з часом. Це потребує гнучкого підходу щодо вибору рівня і засобів захисту таких даних [3]. Традиційні підходи на основі послідовної архітектури шифрування, що базуються на процесорах загального призначення або графічних прискорювачах, нерідко виявляють неспроможність забезпечити детерміновану

обробку потоків даних у реальному часі внаслідок архітектурних обмежень, зумовлених механізмами переривань та ієрархією кеш-пам'яті. У цьому контексті реконфігуровані обчислювальні системи стають платформою для побудови високопродуктивних криптопроцесорів завдяки можливостям своїй архітектурі [4].

Водночас широка імплементація паралелізму як на програмному рівні, так і тим більше при апаратній реалізації на одному кристалі породжує низку технологічних викликів. Просте масштабування кількості обчислювальних ядер призводить до актуалізації проблеми обміну даними з пам'яттю. При неупорядкованому зверненні N паралельних модулів до спільної шини даних або зовнішніх інтерфейсів виникають колізії, які, за відсутності належного керування, спричиняють непрогнозовані затримки, зменшення ефективної пропускної здатності або повну відмову системи внаслідок взаємного блокування ресурсів різними процесами.

АНАЛІЗ ДОСЛІДЖЕНЬ ТА ПУБЛІКАЦІЙ

Питання апаратної реалізації паралельних обчислювачів та синхронізації в них конкуруючих процесів, тому числі при використанні хмарних технологій, розглядаються в [4, 5]. Зручним для їх опису є графічні алгоритми. Фундаментальні засади моделювання асинхронних дискретних систем викладено у класичній праці [6], де обґрунтовано переваги графічного формалізму для опису паралельних процесів. Сучасні публікації [4, 5, 7, 8] засвідчують ефективність застосування мереж Петрі при проектуванні логічних контролерів у FPGA, пропонуючи методики прямої трансляції моделей у VHDL-код. Проте більшість зазначених досліджень зосереджено на управлінні станами керуючих автоматів (FSM), залишаючи поза увагою аспекти моделювання потокової обробки великих масивів даних із врахуванням буферизації. З іншого боку, наявні публікації щодо апаратної реалізації алгоритму AES [9, 10, 12] фокусуються переважно на оптимізації внутрішньої мікроархітектури шифратора (глибока конвеєризація, розгортання циклів, оптимізація S-Box). При цьому системні аспекти синхронізації, арбітражу доступу до інтерфейсів AXI/Avalon та вплив черг на інтегральну продуктивність у багатоядерних конфігураціях часто залишаються поза межами детального розгляду. Проте, саме механізми синхронізації стають "вузьким місцем" сучасних систем на кристалі (SoC), детермінуючи їхню реальну, а не пікову продуктивність.

Мета статті полягає у є розробці підходу до моделювання паралельних процесів обробки й доступу до даних у FPGA-орієнтованій архітектурі шифрування, що базується на мережах Петрі та доповнюється експериментальною оцінкою масштабованості програмних прототипів, та дозволяє обґрунтувати оптимальний підхід до вибору апаратних і програмних ресурсів для достатнього рівня захисту персоніфікованих даних в залежності від рівня їх конфіденційності на основі ієрархічної моделі відповідності між рівнем конфіденційності та рівнем захисту даних.

Для обґрунтування такого підходу досліджували питання вибору AES [12 – 14] у режимі CTR як базового режиму для незалежної паралельної обробки блоків даних; запропоновано узагальнену архітектуру мережі Петрі, яка відображає всі цикли обробки даних; сформували відповідність між елементами мережі Петрі та компонентами архітектури; реалізували програмні прототипи паралельного AES-CTR і отримати експериментальні дані часу виконання, пропускної здатності й прискорення на різній кількості потоків і різних обсягах даних; розглянути модифікацію моделі відповідно до вимог ієрархічного підходу захисту персоніфікованих даних (рівень прийняття рішення щодо політики захисту).

ВИКЛАД ОСНОВНОГО МАТЕРІАЛУ
АРХІТЕКТУРА ПАРАЛЕЛЬНОЇ СИСТЕМИ ШИФРУВАННЯ

З метою досягнення максимізації рівня паралелізму та забезпечення незалежності обчислювальних потоків обрано режим шифрування AES-CTR. На відміну від режимів зчеплення блоків, де ініціалізація обробки наступного блоку можлива виключно після завершення попереднього, режим CTR трансформує блоковий шифр у потоковий. Це уможливлює генерацію ключової гами для довільної кількості блоків оперуючи лише значеннями лічильника та ключа, що ідеально корелює з архітектурою FPGA, яка містить значний масив розподілених логічних елементів.

Пропонована архітектура системи для реалізації на ПЛІС (рис. 1) включає такі компоненти:

1. *Вхідний буфер (FIFO In):* забезпечує асинхронну акумуляцію вхідних даних та узгодження швидкостей (rate matching).
2. *Масив обчислювальних ядер:* сукупність K ідентичних, автономних модулів AES. Кожен модуль розглядається як конвеєрний обробник із фіксованою латентністю. Масштабування системи реалізується шляхом варіювання параметра K відповідно до наявних ресурсів кристала.
3. *Арбітр шини:* логічний блок, що імплементує механізм взаємного виключення для керування процесом запису результатів у єдиний вихідний канал. Арбітр гарантує цілісність даних, запобігаючи колізії при надходження байтів від різних пакетів, та визначає пріоритетність доступу у випадку одночасної готовності даних з декількох ядер.

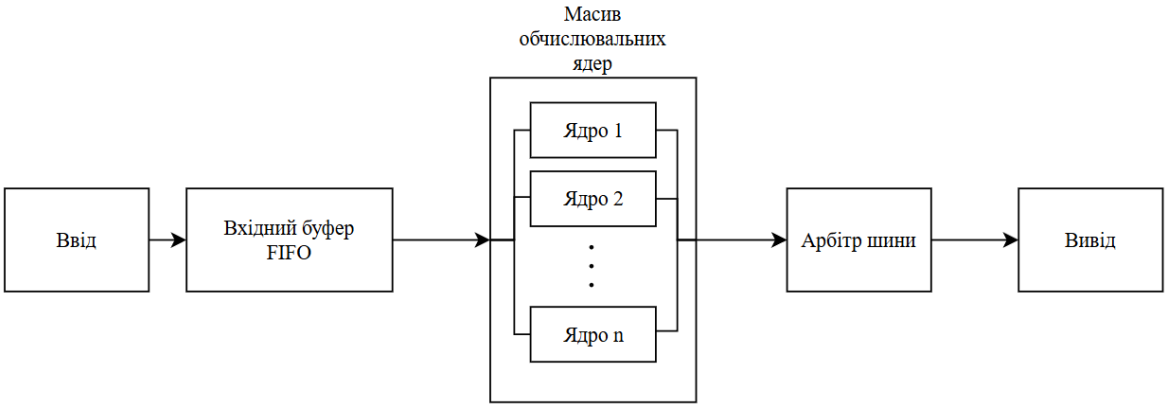


Рис.1. Головні компоненти архітектури паралельної системи

ФОРМАЛІЗАЦІЯ МОДЕЛІ ЗАСОБАМИ МЕРЕЖ ПЕТРІ

Для опису поведінки системи використано класичну *P/T*-мережу Петрі, яка визначається типовим кортежем:

$$PN = (P, T, F, W, M_0),$$

де:

- $P = \{p_1, p_2, ..., p_m\}$ – скінченна множина вузлів, що моделюють стани ресурсів, буферів FIFO, регістрів статусу та семафорів;
- $T = \{l_1, l_2, ..., l_n\}$ – скінченна множина переходів, що відповідають подіям системи (надходженню тактових імпульсів, сигналів готовності, сигналів запису/читання);
- $F = (P \times T) \cup (T \times P)$ – відношення інцидентності, що детермінує потоки даних та сигналів керування;
- W – вагова функція дуг (у даній моделі прийнято $W = 1$, що спрощує апаратну реалізацію);
- M_0 – початкове маркування, що визначає початковий стан системи (після сигналу скидання).

Визначальною особливістю моделі є застосування *ресурсного підходу.* Замість надлишкового графічного дублювання структури для кожного з K ядер, вводиться спеціалізоване місце-ресурс P_{idle} кількість токенів у цьому місці динамічно відображає кількість доступних апаратних модулів у конкретний часовий проміжок.

Детальна специфікація відповідності елементів моделі апаратним компонентам:

- P_{in} *(Вхідна черга):* містить токени, що репрезентують пакети даних, які надійшли на обробку. Початкове маркування $M_0(P_{in}) = N_{data}$ моделює пакетне навантаження. В апаратному забезпеченні це корелює зі станом "Not Empty" вхідного FIFO.
- P_{idle} *(Вільні ядра):* лічильник доступних модулів AES. Початкове значення $M_0(P) = K$ виступає параметром синтезу. Наявність токена в цій позиції сигналізує, що принаймні одне ядро перебуває в стані IDLE і готове до прийому даних.
- P_{mutex} *(Семафор шини):* забезпечує монопольний доступ до вихідного інтерфейсу.

$M_0(P_{mutex}) = 1$ – це аналог апаратного прапорця "Bus Busy". Володіння цим токеном можливе лише для одного процесу, що фізично унеможливлює конфлікт драйверів шини.

- T_{start} *(Початок обробки):* перехід активується (*is enabled*) виключно за умови кон'юнкції: наявності даних ($M(P_{in}) > 0$) та наявності вільного ресурсу ($M(P_{idle}) > 0$). Це моделює логіку функціонування диспетчера завдань *(Dispatcher).*

МОДЕЛЮВАННЯ ТА СИМУЛЯЦІЯ В СЕРЕДОВИЩІ WOPED

З метою верифікації коректності запропонованої архітектури та виявлення потенційних логічних помилок реалізовано комплекс експериментальних досліджень у програмному середовищі *WoPeD.* Застосування спеціалізованого інструментарію дозволяє візуалізувати графічну модель системи (рис. 2). Топологія мережі відображає повний життєвий цикл обробки блоку даних:

1. Захоплення ресурсу ядра та читання з вхідного буфера (T_{start})
2. Виконання операції шифрування (перебування токена в місці $P_{process}$, що імітує часову затримку обчислень).
3. Запит на доступ до шини (конкуренція за токен P_{mutex})
4. Запис результату у *вихідний* буфер (T_{write})
5. Звільнення ресурсів: повернення токена ядра в P_{idle} та токена шини в P_{mutex}.

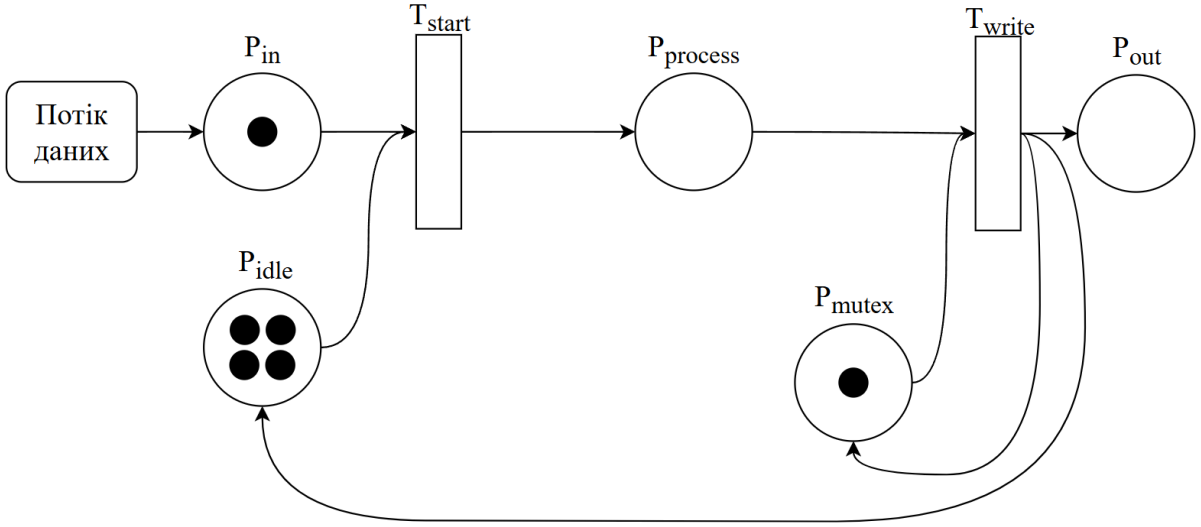


Рис. 2. Узагальнена структура моделі мережі Петрі пропонованої системи

Аналіз простору станів. Побудова графа досяжності уможливила аналіз усіх потенційних динамічних станів системи. Наприклад, для конфігурації з $K = 4$ ядрами граф містить скінченну множину вузлів, що свідчить про стабільність системи. Аналіз симуляції засвідчив відсутність у графі досяжності тупикових станів за винятком цільового стану завершення роботи. Це означає, що за будь-якої послідовності спрацювання переходів система не увійде в стан нескінченного очікування, коли ядра очікують на шину, а шина заблокована ядрами.

Перевірка структурних інваріантів. Окрім динамічного аналізу, здійснено перевірку статичних властивостей через P-інваріанти. Моделювання в середовищі WoPeD дозволяє підтвердити фундаментальні рівняння збереження ресурсів:

$$M(P_{idle}) + M(P_{processS}) = K$$

Фізичний зміст цього співвідношення показує, що сума вільних ядер та ядер, які перебувають у процесі обробки, є константою і дорівнює фізичній кількості імплементованих модулів на кристалі ПЛІС. Це доводить відсутність "витоку ресурсів" (resource leak), коли модуль може опинитися в невизначеному стані.

Математичне співвідношення, яке гарантує відсутність колізій при записі:

$$M(P_{mutex}) + M(P_{writing}) = 1$$

Ця рівність показує, що токен може знаходитися або в місці семафора (шина вільна), або в стані активного запису (шина зайнята). Ситуація $M(P_{writing}) > 1$ є математично неможливою, що повністю виключає ризик пошкодження даних на апаратному рівні.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ МАСШТАБОВАНОСТІ

Проведена симуляція дозволила дослідити поведінку системи за умов змінного навантаження. Встановлено, що при збільшенні кількості вхідних токенів система демонструє два режими функціонування:

1. *Лінійне зростання.* Доки пропускна здатність шини перевищує сумарну продуктивність ядер, додавання нового ядра (K+1) лінійно підвищує загальну швидкість обробки.
2. *Насичення.* При досягненні певного критичного значення K_{crit}, швидкість системи лімітується пропускною здатністю переходу T_{write}. У цьому режимі збільшення кількості ядер не забезпечує приросту продуктивності, а лише призводить до зростання черги очікування на доступ до шини ($P_{process}$ -> T_{write}).

Моделювання дозволяє аналітично визначити це критичне співвідношення на етапі проектування, уникаючи нераціональних витрат ресурсів кристала на створення ядер, які будуть простоювати через обмеження шини.

ПРОГРАМНИЙ ПРОТОТИП ФУНКЦІОНАЛУ МОДЕЛІ FPGA-АРХІТЕКТУРИ

Для практичної апробації запропонованої Petri-моделі та оцінювання масштабованості було реалізовано програмний прототип архітектури паралельного шифрування AES-CTR. Обраний режим AES-CTR забезпечує незалежність обробки блоків і тим самим природну паралельність обчислень, оскільки генерація ключової гами визначається лише значенням лічильника та ключа.

Прототип розглядається не як еквівалент HDL-опису, а як функціональна модель керування потоками даних і конкурентного доступу до ресурсів, типових для FPGA-систем. Його структура узгоджується з базовими компонентами запропонованої архітектури: вхідна буферизація для узгодження швидкостей, масив

обчислювальних ядер AES як незалежних виконавців та арбітр шини, що реалізує взаємне виключення під час запису результатів у спільний канал .

Ключовим мотивом такого підходу є те, що в багатоядерних конфігураціях реальна продуктивність часто визначається не піковою швидкодією ядра, а механізмами синхронізації, арбітражу та черг, які формують системні “вузькі місця”. Ці механізми і відтворюються у програмному прототипі через моделі диспетчеризації задач, буферизації та серіалізованого запису.

МЕТОДИКА ЕКСПЕРИМЕНТІВ ТА ОБРАНІ МЕТРИКИ

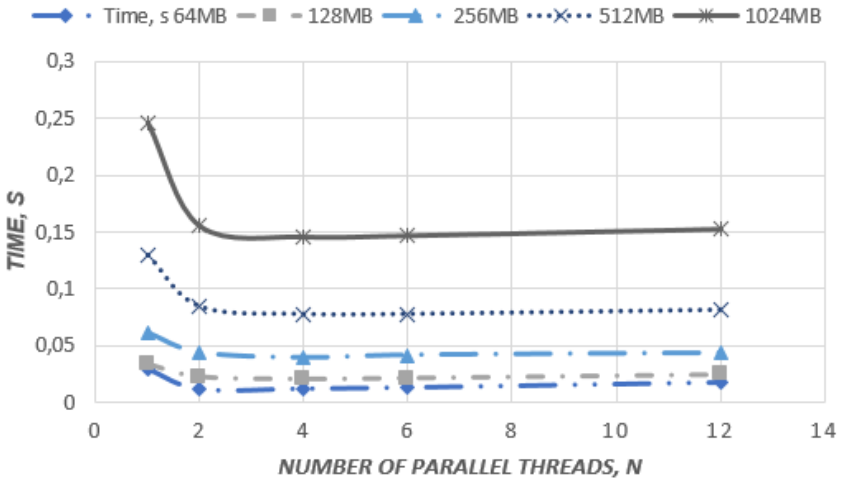
Експерименти виконувались для наборів даних фіксованого обсягу та для різної кількості паралельних потоків (ядер AES у моделі), що відповідає ідеї масштабування K в апаратній конфігурації масиву AES-модулів. При цьому застосовували три конфігурації програм моделювання: базову програмну модель пакету OpenSSL, власний розроблений адаптований програмний пакет (АПП) та АПП з застосування віртуального прискорення (АПП/SpeedUp). У всіх випадках вхідні дані розбиваються на блоки по 128 біт, для кожного блока формується значення лічильника, генерується ключовий потік та виконується XOR-комбінація з відкритим текстом, що узгоджується з потоковою природою AES-CTR.

Для кожної конфігурації вимірювався час виконання шифрування, пропускна здатність моделі та прискорення відносно однопотокового режиму.

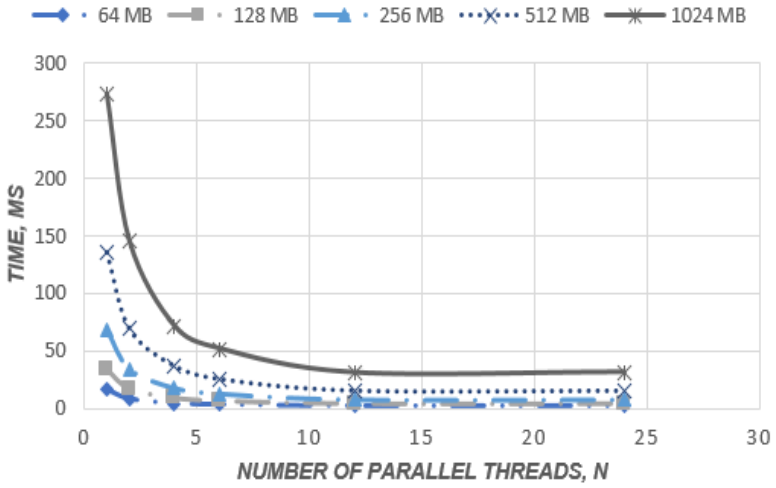
РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТІВ ТА ЇХ ІНТЕРПРЕТАЦІЯ

Отримані результати подано у вигляді трьох узагальнювальних діаграм: залежність часу виконання від кількості потоків, залежність прискорення відносно однопотокового режиму та залежність пропускної здатності пристроїв шифрування/дешифрування від кількості потоків.

Зауважимо, що як і передбачалось вище, спостерігається закономірне зменшення часу реалізації процесів шифрування/дешифрування зі зростанням кількості запущених потоків (рис. 3), що запускають індивідуальні ядра AES, та наступний вихід на насичення для всіх досліджених обсягів даних.



а)



б)

Рис. 3. Залежності часу шифрування від кількості запущених потоків: а) згідно моделі пакету OpenSSL; б) за моделлю АПП/SpeedUp

Найбільш виражений ефект паралелізації проявляється при переході від малих до середніх значень N (порядку кількох потоків). Проте, якщо для моделі пакету OpenSSL (рис. 1, а) цей процес виходить на насичення вже при $N = 2 \div 4$, то застосування програмного прискорення за моделлю АПП/SpeedUp (рис. 1, б) це відбувається при значеннях $N \geq 12$. За абсолютними часовими параметрами значення для моделі АПП/SpeedUp порівняно зі значеннями для моделі пакету OpenSSL дають майже у 4 рази кращий результат: вихід на насичення за рахунок конкуренції процесів паралельної обробки коду і синхронізації процесів розпаралелювання спостерігається при $t_{SP}=40$ *мс* порівняно з $t_{OpenSSL}$=150 мс при тих же тестових вибірках обсягом 1024 МВ.

Діаграма (рис. 4) відображає практичну межу масштабування: пропускна здатність зростає разом із кількістю потоків, однак після певного значення $N=12$ приріст стає незначним. З точки зору архітектури це відповідає ситуації, коли масив AES-ядер перестає бути вузьким місцем, а домінуючим обмеженням стає підсистема синхронізації процесів запису та передачі результату.

Залежність продуктивності системи від обсягу даних в тестових вибірках при збільшенні кількості паралельних потоків не виражена так сильно. Однак спостерігається деяке пониження продуктивності при середніх значеннях обсягів вибірок (256 і 512 МВ) за N рівне 2, 4 і 24, тоді як в однопоточному режимі та при N рівне 6 і 12 продуктивність системи для шифрування/дешифрування даних майже вирівнюється (рис. 4).

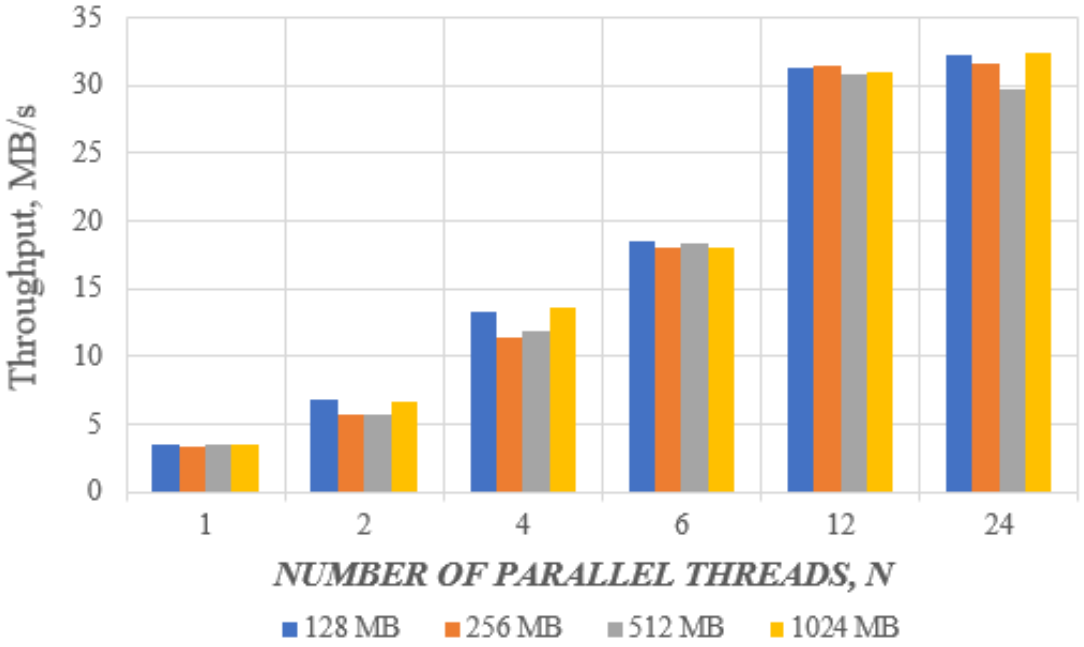
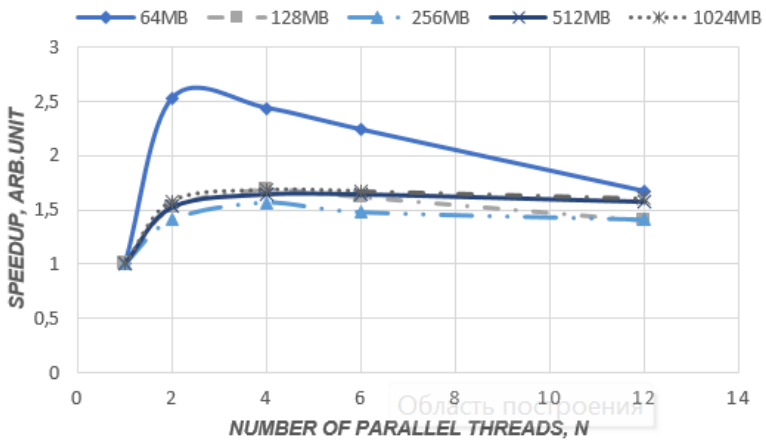
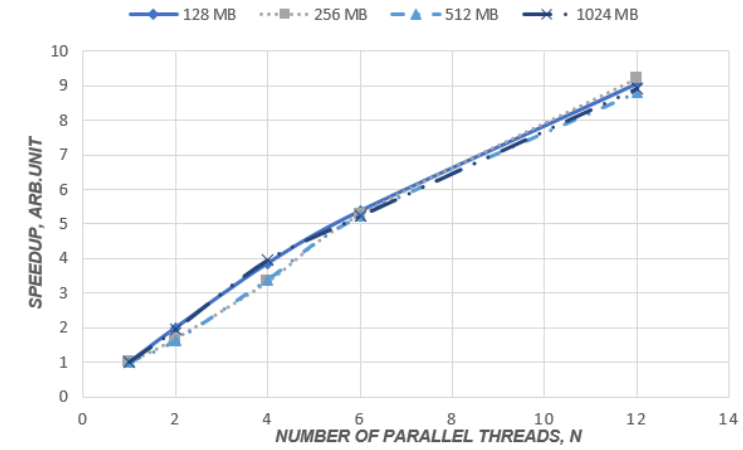


Рис. 4. Діаграма залежності пропускної здатності системи відносно кількості потоків

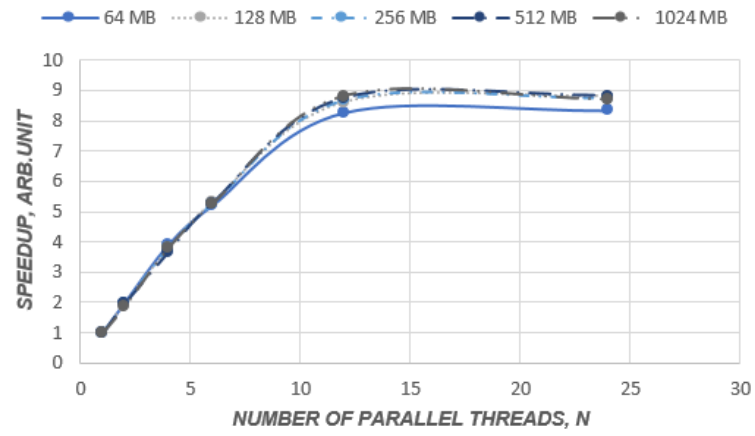
Отримані результати моделювання прискорення обробки тестових щодо використовуваних тестових вибірок при застосуванні різних програмних пакетів приведені на (рис. 5). Цікаво, що для пакету OpenSSL (рис. 5, а) спостерігається різке зростання прискорення процесу шифрування для файлів меншого обсягу даних (64 МВ), яке далі зменшується зі зростанням кількості використовуваних ядер. Для вибірок більшого об'єму даних (128 – 1024 МВ) прискорення зростає при $N = 2 \div 4$ і в подальшому зменшується на 5-10%. Така поведінка узгоджується з системною природою обмежень – при зростанні кількості паралельних виконавців збільшується конкуренція за спільні ресурси системи.



а)



б)



в)

Рис. 5. Залежності прискорення шифрування від кількості паралельних потоків для різних пакетів програм та обсягу шифрованих файлів вибірок даних: а) для моделі пакету OpenSSL; б) для моделі АПП; в) за моделлю АПП/SpeedUp

Для моделей АПП та АПП/SpeedUp прискорення прогнозовано зростає практично для файлів всіх тестових вибірок при розпаралелюванні потоків до значень $N = 12$, і далі виходить на насичення. Однак, якщо для звичайного пакету АПП спостерігається певна залежність прискорення шифрування від обсягу даних при $N = 2 \div 4$ (рис. 5, б), то для пакету з прискоренням АПП/SpeedUp така залежність спостерігається вже при $N = 10 \div 12$ і дещо зменшується (приблизно на 5-7%) для вибірок даних меншого обсягу (рис. 5, в).

Загалом отримані результати підтверджують можливість оптимізації апаратно-програмних засобів для систем захисту персоніфікованих даних в інформаційних та телекомунікаційних мережах за рахунок застосування реконфігурованих пристроїв. Важливим при цьому є підхід на основі використання моделі ієрархічної класифікації конкретних персональних даних за критеріями наборами критеріїв рівня конфіденційності, швидкості доступу, важливості та іншими в залежності від яких вибирається конкретна модель програмного чи апаратного захисту інформації та глибини цього захисту.

ВИСНОВКИ З ДАНОГО ДОСЛІДЖЕННЯ І ПЕРСПЕКТИВИ ПОДАЛЬШИХ РОЗВІДОК У ДАНОМУ НАПРЯМІ

У дослідженні розроблено та теоретично обґрунтовано методику моделювання паралельних процесів обробки даних у реконфігурованих системах із використанням формалізму мереж Петрі. Запропонований підхід дозволяє подолати розрив між високорівневим описом алгоритмів та їх низькорівневою реалізацією.

Основні результати дослідження:

1. Розроблена ресурсна модель уможливлює врахування конкуренції за спільні ресурси (пам'ять, шини) на рівні системної архітектури, що є недосяжним при використанні традиційних методів симуляції окремих модулів.
2. Шляхом автоматизованої верифікації у WoPeD доведено властивості обмеженості та живості моделі. Це гарантує, що синтезована на її основі апаратура буде вільною від взаємних блокувань та переповнень буферів.
3. Реалізовано програмні прототипи паралельного AES-CTR і виконано серію експериментів масштабованості для різної кількості потоків і обсягів даних. Отримані залежності часу виконання, пропускної здатності та прискорення демонструють характерне насичення при збільшенні числа потоків, що узгоджується з ресурсними обмеженнями та підтверджує роль арбітражу та пам’яті як домінуючого фактору в багатоядерних конфігураціях.

Практичне значення проведених досліджень та отриманих результатів полягає у можливості використання розробленої формальної моделі та експериментально підтверджених закономірностей масштабування для подальшого синтезу керуючих модулів у складі високопродуктивних криптосистем на FPGA, а також для інтеграції таких прискорювачів у програмно-апаратні рішення захисту персоніфікованих даних для моделей на основі ієрархічного підходу до класифікації персоніфікованих даних за критеріями конфіденційності, важливості і своєчасності, та застосування до них різних рівнів апаратного і програмного захисту.

Для подальших досліджень цікавим є застосування апробованого в даній статті підходу до досліджень впливу розпаралелювання обробки даних на швидкість, продуктивність і прискорення процесів шифрування/дешифрування інформації при використанні інших крім AES методів шифрування та інших обсягів вибірок, а також їх застосування для практичної їх реалізації в реконфігуровних середовищах.

References

1. Limar, I., Basov, V., Kireev, I., Golev, D., & Sevasteev, E. (2025). Analysis Of User Password Security Using Python-Script. // Measuring And Computing Devices In Technological Processes. Vol.3. pp.58–66. – https://doi.org/10.31891/2219-9365-2025-83-8
2. Rudy, R., & Vorobets, H. (2025). Hardware And Software Solutions For Protection Of Personalized Data And Information In Telecommunication Systems And Networks. // Herald of Khmelnytskyi National University. Technical Sciences. 357(5.2), pp. 48-54. – https://doi.org/10.31891/2307-5732-2025-357-65
3. Dzyuban, O. Multi-level system for protecting information and telecommunication networks from information security threats based on structural segmentation. // Herald of Khmelnytskyi National University. Technical Sciences. – 2025. – 355(4). – P.140-146. https://doi.org/10.31891/2307-5732-2025-355-20
4. Renaud Pacalet, Matteo Bertolino, Ludovic Apvrille, Andrea Enrici. Multi-resource scheduling for FPGA systems. // Microprocessors and Microsystems: Embedded Hardware Design, 2021, 87, pp.104373. 10.1016/j.micpro.2021.104373. hal-03483455. - https://telecom-paris.hal.science/hal-03483455/file/micpro.pdf
5. Romina Soledad Molina, Veronica Gil-Costa, María Liz Crespo, & Giovanni Ramponi (2022). High-Level Synthesis Hardware Design fo FPGA-Based Accelerators: Models, Methodologies, and Frameworks. // IEEE Access (Volume: 10) https://ieeexplore.ieee.org/document/9864576/authors#authors
6. Peterson J.L. Petri Nets. ACM Computing Surveys. 1977. Vol. 9, no. 3. P. 223-252. – https://doi.org/10.1145/356698.356702
7. Wisniewski R. Design of Petri Net-Based Cyber-Physical Systems Oriented on the Implementation in FPGA. Energies. 2021. Vol. 14, no. 21. – https://www.mdpi.com/1996-1073/14/21/7054
8. Sujoy Sinha Roy, Furkan Turan, Kimmo J¨arvinen, Frederik Vercauteren and Ingrid Verbauwhede (2019). FPGA-based High-Performance Parallel Architecture for Homomorphic Computing on Encrypted Data. // https://eprint.iacr.org/2019/160.pdf
9. Hodjat A., Verbauwhede I. A 21.54 Gbits/s Fully Pipelined AES Processor on FPGA. / Proceedings of the 12th Annual IEEE Symposium on Field-Programmable Custom Computing Machines. 2004. https://ieeexplore.ieee.org/document/1364653
10. Sklavos N., Koufopavlou O. Implementation of the AES encryption core in FPGA. // Journal of Supercomputing. 2002. Vol. 22. P. 213-228.
11. Sudacevschi V., Gutuleac L., Ababii V. A Hardware Implementation Of Petri Nets Models. / 7th International Conference on Development And Application Systems. Suceava, Romania, May 27-29, 2004. – pp. 24-28. https://dasconference.ro/papers/2004/05_A5_Viorica_Sudacevski.pdf.
12. Reshma Nadaf & Satish S. Bhairannawar. An efficient controller-based architecture for AES algorithm using FPGA. // Indonesian Journal of Electrical Engineering and Computer Science. Vol. 35, No. 1, July 2024, pp. 397~404. ISSN: 2502-4752, DOI: 10.11591/ijeecs.v35.i1.pp397-404.
13. Sarita Sanap & Vijayshree More. An Ultra-High Throughput and Efficient Implementation of Advanced Encryption Standard. // International Journal of Electrical and Electronic Engineering & Telecommunications. – 2022. – pp. 1-7. – https://www.ijeetc.com/uploadfile/2022/0920/20220920031233442.pdf.
14. Sovin Ya. R., Khoma V. V. & Otenko V. I. Comparison of AEAD algorithms for embedded systems of the Internet of Things. // Computer Systems and Networks. – 2019. – T.1., No. 1. - P. 76-91. – https://science.lpnu.ua/sites/default/files/journal-paper/2020/feb/21055/var1ksm-19-78-93.pdf