КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 2 (34), 2026
ISSN 2663 – 4023
DOI 10.28925/2663-4023.2026.34.1326
УДК 004.056.55

Івасьєв Степан Володимирович
к.т.н, доцент, доцент кафедри кібербезпеки
Західноукраїнський національний університет, Тернопіль, Україна
ORCID:0000-0003-2243-5956
isv@wunu.edu.ua

Янік Іван Іванович
Аспірант кафедри кібербезпеки
Західноукраїнський національний університет, Тернопіль, Україна
ORCID:0009-0004-3562-3352
i.yanik@st.wunu.edu.ua

АРХІТЕКТУРА RSA-CRT З ПОДВІЙНИМ ЗАХИСТОМ КАНАЛУ ЗВ'ЯЗКУ ТА СТІЙКІСТЮ ДО ІН'ЄКЦІЇ ПОМИЛОК У МОДУЛЬНІ ЗАЛИШКИ

Анотація. Зростання складності кіберзагроз, а також вразливість криптографічних пристроїв до ін'єкції помилок і впливу завад у каналах зв'язку зумовлюють необхідність розроблення комплексних відмовостійких архітектур криптографічного захисту. Незважаючи на те, що алгоритм RSA-CRT забезпечує суттєве прискорення виконання операцій модульного піднесення до степеня завдяки використанню Китайської теореми про залишки, він залишається вразливим до атак типу fault injection, за яких навіть одинична індукована помилка може призвести до компрометації закритого ключа або порушення коректності криптографічних обчислень. Додатковою проблемою є забезпечення цілісності даних під час їх передавання каналами зв'язку, що працюють в умовах випадкових і пакетних помилок.

У роботі запропоновано архітектуру RSA-CRT із подвійним захистом, яка поєднує механізми забезпечення цілісності криптографічних обчислень та корекції помилок під час передавання даних. Перший рівень захисту реалізовано на основі надлишкової системи залишкових класів, що містить три інформаційні та два надлишкові модулі. Запроваджена надлишковість забезпечує можливість виявлення, локалізації та виправлення помилкових модульних залишків шляхом реконструкції результату за процедурою більшості без порушення властивої системі залишкових класів паралельності обчислень. Другий рівень захисту реалізовано за допомогою кодів Reed-Solomon над полем Галуа $\mathrm{GF}(2^8)$, які забезпечують виявлення та виправлення випадкових і пакетних помилок, що виникають у каналі зв'язку до початку криптографічної обробки інформації. Розроблено програмний прототип та проведено серію експериментальних досліджень в умовах одночасної дії завад каналу зв'язку та індукованих помилок у процесі виконання модульних арифметичних операцій RSA-CRT. Результати моделювання підтвердили можливість успішного виправлення помилок передавання даних і локалізації помилкових модульних залишків без повторного виконання криптографічних обчислень. Запропонована архітектура розширює можливості традиційної реалізації RSA-CRT шляхом інтеграції двох взаємодоповнювальних механізмів захисту, що функціонують на різних етапах опрацювання інформації. Такий підхід забезпечує комплексний захист як даних, що передаються каналом зв'язку, так і результатів криптографічних обчислень, що робить запропоноване рішення перспективним для застосування у вбудованих системах, кіберфізичних системах, промислових контролерах та інших інформаційно-керувальних системах, у яких висуваються підвищені вимоги до надійності та безпеки реалізації криптографічних алгоритмів.

Ключові слова: алгоритм RSA-CRT, ін'єкція помилок, надлишкова система залишкових класів, коди Reed-Solomon, відмовостійкість.

ВСТУП

Постановка проблеми. Криптосистема з відкритим ключем RSA є фундаментальним елементом сучасних систем інформаційної безпеки, однак її класична реалізація суттєво вразлива до помилок. Навіть незначне спотворення зашифрованих даних, спричинене завадами в каналі чи збоями апаратури, унеможливлює коректне дешифрування, що обмежує застосування алгоритму в нестабільних середовищах. Для оптимізації та захисту RSA застосовують надлишкову систему залишкових класів (Redundant Residue Number System, RRNS), яка завдяки введенню додаткових модулів дозволяє не лише розпаралелити обчислення але й забезпечити стійкість до цілеспрямованих атак ін'єкції помилок (fault injection attacks).

Водночас захист на рівні RRNS не вирішує проблеми виникнення помилок під час передачі всього блоку шифротексту по каналу зв'язку. Для забезпечення цілісності на цьому етапі доцільно застосовувати другий рівень захисту – ефективні коди, що виправляють помилки, такі як коди Goppa або Reed-Solomon, здатні відновлювати дані навіть після їх значного спотворення. Таким чином, головна проблема полягає у відсутності єдиної архітектури, яка б комплексно поєднувала обчислювальні переваги паралелізації RSA, відмовостійкість до атак та захист від спотворень у каналах передачі.

Аналіз останніх досліджень і публікацій. Для обґрунтування актуальності та новизни запропонованої архітектури необхідно провести аналіз існуючих досліджень за ключовими напрямками. Цей огляд охоплює фундаментальні праці, присвячені вразливостям криптосистеми RSA-CRT до атак внесення помилок, та існуючим методам протидії. Окрему увагу приділено застосуванню RRNS та кодів з високою корекційною здатністю як основи для побудови надійних систем захисту інформації.

У роботі [1] описано розробку програмної моделі для симуляції роботи завадостійких кодів Reed-Solomon на основі об'єктно-орієнтованої технології. Створена програма реалізує кодування та декодування для кодів типу (255, 239) та (255, 223) у скінченному полі $GF(2^8)$, що дозволяє моделювати процес виправлення помилок у каналах зв'язку.

У роботі [2] досліджується ефективність поєднання асиметричної криптосистеми RSA з кодами для корекції помилок з метою підвищення безпеки обміну інформацією. Запропоновано алгоритм, що використовує шифрування RSA для захисту даних та коди Hemming у скінченних полях Галуа $GF(q)$ для виявлення та виправлення помилок передачі.

У роботі [3] представлено новий алгоритм спискового декодування для класичних бінарних кодів Goppa. Цей алгоритм, що працює за поліноміальний час, дозволяє виправити значно більше помилок у порівнянні з попередніми методами, зокрема з алгоритмом Peterson. Автор зазначає, що підвищення корекційної здатності має пряме застосування для посилення безпеки код-базованих криптосистем, таких як McEliece.

У роботі [4] досліджується ефективність пам'яті з кодами корекції помилок (Error-Correcting Code, ECC) як засобу захисту від атак типу Rowhammer. Автори демонструють, що стандартний механізм ECC не є абсолютною гарантією безпеки, оскільки процес виправлення помилок створює часовий побічний канал. На основі цієї вразливості запропоновано нову модель атаки – «Intermittent Fault Attack», яка дозволяє зловмиснику отримувати секретну інформацію.

У роботі [5] надається огляд загроз безпеці для постквантових криптографічних алгоритмів, що виникають на етапі їх практичної реалізації. Автори підкреслюють, що навіть алгоритмічно стійкі схеми, включаючи кодові криптосистеми, можуть бути вразливими до атак побічними каналами та атак внесення помилок.

У роботі [6] детально аналізуються існуючі контрзаходи для захисту реалізацій RSA-CRT від атак другого порядку, заснованих на внесенні помилок. Автори показують, що багато запропонованих рішень не є внутрішньо стійкими до комбінованих атак. У статті запропоновано загальний принцип «блокування» (Lock principle) як ефективний метод для посилення захисту від атак цього класу.

У роботі [7] представлено систематичний та ретельний огляд криптографічних схем на основі RSA. Авторами проведено класифікацію існуючих модифікацій алгоритму за 11 різними категоріями, серед яких зокрема виділено методи, що базуються на Китайській теоремі про залишки (Chinese Remainder Theorem, CRT). Дослідження підкреслює, що основними напрямками удосконалення RSA є підвищення його безпеки та продуктивності.

У роботі [8] описано практичну реалізацію атаки подвійної помилки (double-fault attack) на захищену реалізацію RSA-CRT. Продемонстровано, що перша помилка може бути використана для спотворення результату однієї з модульних експоненціацій, а друга – для обходу процедури перевірки помилок, що дозволяє успішно реалізувати атаку типу Bellcore.

У роботі [9] розглядаються методи контролю, діагностики та виправлення помилок у комп'ютерних системах, що функціонують у модульній системі числення (MNS). Детально аналізується

метод проєкцій та пропонується більш ефективний підхід, заснований на концепції «альтернативної множини чисел», що дозволяє виправляти помилки навіть за наявності єдиної контрольної основи.

У роботі [10] автори класифікують існуючі контрзаходи проти атак з внесенням помилок на CRT-RSA, що дозволяє глибше зрозуміти принципи їхньої роботи. Доведено теоретичну еквівалентність між тестовими та інфекційними підходами до захисту. На основі цього аналізу запропоновано спосіб побудови доказово стійких контрзаходів високого порядку.

У роботі [11] запропоновано метод шифрування даних, що базується на RRNS. Для підвищення криптографічної стійкості існуючого методу введено додатковий етап перемішування залишків за допомогою ключа, згенерованого на основі M-послідовності, та виведено формулу для оцінки криптографічної стійкості.

Аналіз наукових праць підтверджує, що реалізації RSA-CRT, попри свою ефективність, є вразливими до атак внесення помилок, причому існуючі контрзаходи та стандартні апаратні засоби захисту не завжди забезпечують надійний захист від атак вищого порядку [6, 8, 10]. Це обґрунтовує необхідність розробки відмовостійких архітектур на арифметичному рівні, де перспективним напрямом є використання RRNS [9, 11]. Водночас для захисту від спотворень у каналах зв'язку широко застосовуються ефективні алгебраїчні коди, що виправляють помилки, зокрема коди Reed-Solomon [1] та Goppa [3]. Ключовий висновок аналізу полягає в тому, що існуючі дослідження [2, 7] переважно розглядають захист обчислень та захист каналу як два окремі завдання. Тому розробка єдиної дворівневої архітектури, що поєднує відмовостійкість RRNS та завадостійкість ефективних кодів, є актуальним і важливим кроком до створення комплексних та надійних криптографічних систем.

Мета статті. Розробка та дослідження такої гібридної дворівневої криптосистеми, що інтегрує ці механізми для досягнення високого рівня надійності та продуктивності.

МАТЕМАТИЧНІ ОСНОВИ АЛГОРИТМУ RSA

Алгоритм RSA є класичним прикладом асиметричного шифрування, заснованого на математичних властивостях великих простих чисел. В основі RSA лежать два ключових етапи: генерування публічного та приватного ключів і обчислення залишків за модулем великого числа.

Генерація ключів RSA:

1. Вибираються два великі прості числа p і q.
2. Обчислюється $N = p \times q$, яке використовується як модуль для шифрування та розшифрування.
3. Визначається функція Ейлера $\varphi(N) = (p-1) \times (q-1)$.
4. Вибирається число e (відкритий експонент), яке є взаємно простим з $\varphi(N)$.
5. Обчислюється приватний ключ d, такий що $e \times d = 1 \; mod \; \varphi(N)$.

Публічний ключ складається з пари значень (e, N), а приватний — з пари (d, N).

Шифрування повідомлення M з використанням публічного ключа виконується за формулою:

$$C = M^e \; mod \; N \tag{1}$$

де C – шифротекст.

Розшифрування здійснюється з використанням приватного ключа:

$$M = C^d \; mod \; N \tag{2}$$

Надійність алгоритму RSA полягає у використанні односторонніх математичних функцій, які неможливо обчислити у зворотному порядку без секретного ключа. Цей принцип гарантує, що лише власник ключа може розшифрувати дані та підтвердити автентичність цифрового підпису, забезпечуючи конфіденційність та цілісність інформації.

ДОСЛІДЖЕННЯ СИСТЕМИ ЗАЛИШКОВИХ КЛАСІВ ТА КИТАЙСЬКОЇ ТЕОРЕМИ ЗАЛИШКІВ

Система залишкових класів (Residue Number System, RNS) є ефективним математичним інструментом, що дозволяє представити велике ціле число у вигляді вектора його залишків за низкою взаємно простих модулів. Однією з ключових переваг RNS є можливість незалежного виконання основних арифметичних операцій (додавання, віднімання, множення) у кожному з модульних каналів, що відкриває широкі можливості для реалізації паралельних обчислень. Такий підхід є особливо актуальним для сучасних обчислювальних архітектур, криптографічних схем та систем цифрової обробки сигналів, де критично важливою є продуктивність і стійкість до помилок.

Нехай задано набір модулів $m_1, m_2, \dots, m_K$, які попарно взаємно прості: $gcd(m_i, m_j) = 1$ для всіх $i \neq j$, де $i, j \in \{1, 2, \dots, K\}$. Для деякого числа $M \in \mathbb{N}$, його представлення в системі залишкових класів задається системою конгруенцій:

$$\begin{cases} M \equiv r_1 \ (mod\ m_1) \\ M \equiv r_2 \ (mod\ m_2) \\ \vdots \\ M \equiv r_K \ (mod\ m_K) \end{cases} \quad (3)$$

Відновлення числа M з набору залишків $\{r_i\}$ здійснюється за допомогою CRT, яка гарантує існування та єдиність розв'язку у межах динамічного діапазону $[0, M_{max})$, де $M_{max} = \prod_{i=1}^{K} m_i$. Однак класична форма RNS є вразливою до втрати або спотворення навіть одного залишку, що стає причиною неможливості розв'язання системи (3). Для усунення цієї вади можна скористуватися RRNS, яка містить додаткові модулі $m_{K+1}, \dots, m_L$, де $L > K$. Додані модулі створюють надлишковість, що дозволяє виявляти та коригувати помилки у залишках, не втрачаючи загальної здатності до відновлення числа.

Формалізовано, система в RRNS має вигляд:

$$\begin{cases} M \equiv r_1 \ (mod\ m_1) \\ M \equiv r_2 \ (mod\ m_2) \\ \vdots \\ M \equiv r_K \ (mod\ m_K) \\ M \equiv r_{K+1} \ (mod\ m_{K+1}) \\ \vdots \\ M \equiv r_L \ (mod\ m_L) \end{cases} \quad (4)$$

Надлишкові модулі повинні бути попарно взаємно простими як між собою, так і з усіма базовими модулями. Це забезпечує застосовність CRT навіть у разі втрати або спотворення деяких залишків. Як правило, модулі обираються з множини простих чисел P за критерієм: $m_j = min\{p \in P \mid p > max(m_1, m_2, \cdots, m_K)\}$, $\forall j \in \{K+1, \dots, L\}$.

CRT у контексті RRNS виконує подвійну функцію: не лише перетворення залишків у стандартне числове представлення, але й перевірку консистентності залишкової інформації. Зокрема, при наявності надлишкових модулів можлива детекція помилок: якщо число, відновлене з базових залишків, не дає правильних залишків за надлишковими модулями, це вказує на спотворення. Виправлення помилок в RRNS реалізується за допомогою спеціалізованих алгоритмів, більшість з яких базуються на послідовному виключенні підозрілих залишків. Один з підходів полягає у наступному: кожен залишок виключається по черзі, після чого число відновлюється за рештою залишків, включаючи надлишкові. Якщо реконструйоване значення узгоджується з усіма перевірочними залишками, то вважається, що саме виключений залишок є помилковим. Така процедура може бути автоматизована і використана у системах виявлення та виправлення одиничних і кратних помилок.

RRNS надає широкі можливості не лише для високопродуктивних обчислень, але й для підвищення криптостійкості. У криптографічних системах, таких як RSA або інші схеми з відкритим ключем, RRNS може бути використана як механізм захисту від атак побічних каналів або порушення цілісності під час передачі за рахунок відмовостійкого кодування. Таким чином, поєднання RRNS та CRT створює фундамент для побудови надійних систем з корекцією помилок і паралельною обробкою, що є критично важливим для сучасної кібербезпеки.

КОДИ REED-SOLOMON ДЛЯ ЗАХИСТУ КАНАЛУ ЗВ'ЯЗКУ

Коди Reed–Solomon належать до класу недвійкових циклічних блокових кодів, які забезпечують виявлення та виправлення множинних символьних помилок. На відміну від двійкових кодів, у яких коригуються окремі біти, коди Reed–Solomon виконують обробку символів над скінченним полем Галуа $GF(2^m)$, що робить їх особливо ефективними для виправлення пакетних помилок, характерних для каналів передавання цифрової інформації.

Код Reed–Solomon позначають як $RS(n, k)$, де k – кількість інформаційних символів, n – довжина кодового слова після додавання перевірочних символів. Кількість перевірочних символів визначається виразом

$$n - k = 2t,$$

де t – максимальна кількість символьних помилок, які можуть бути гарантовано виправлені.

Кодування здійснюється шляхом формування полінома повідомлення:

$$M(x) = m_0 + m_1 x + \cdots + m_{k-1} x^{k-1},$$

коефіцієнти якого належать полю $GF(2^m)$. Для отримання кодового слова повідомлення множиться на x^{n-k} та доповнюється залишком від ділення на породжувальний поліном $g(x)$:

$$C(x) = M(x)x^{n-k} + R(x),$$

де

$$R(x) = M(x)x^{n-k} \bmod g(x).$$

Таким чином формується систематичне кодове слово, що містить інформаційну та перевірочну частини.

Під час передавання каналом зв'язку кодове слово може зазнавати впливу завад, унаслідок чого приймач отримує поліном

$$Y(x) = C(x) + E(x),$$

де $E(x)$ — поліном помилок.

На першому етапі декодування обчислюються синдроми

$$S_i = Y(\alpha^i), \qquad i = 1,2,\dots,2t,$$

де α – примітивний елемент поля Галуа. Якщо всі синдроми дорівнюють нулю, вважається, що помилки відсутні.

У випадку ненульових синдромів визначаються локатори та значення помилок із використанням алгоритму Берлекемпа-Мессі або розширеного алгоритму Евкліда. Після цього виконується корекція пошкоджених символів і відновлюється початкове повідомлення.

Коди Reed-Solomon широко застосовуються у цифрових системах зв'язку, мережевих протоколах, накопичувачах інформації, супутниковому зв'язку та інших системах, де необхідне забезпечення високої достовірності передавання даних навіть за наявності пакетних помилок.

У запропонованій архітектурі коди Reed-Solomon використовуються для забезпечення цілісності даних під час їх передачі каналом зв'язку. Після успішного декодування отримані дані надходять до криптографічного модуля RSA-CRT. На відміну від кодів Reed-Solomon, які компенсують помилки передачі, надлишкова система залишкових класів забезпечує виявлення та виправлення помилок, що виникають безпосередньо під час виконання модульних арифметичних операцій, зокрема внаслідок ін'єкції помилок у процесі криптографічних обчислень. Таким чином, використання двох незалежних механізмів захисту дозволяє забезпечити комплексний контроль цілісності як даних, що передаються, так і результатів криптографічної обробки.

ПРОЦЕС ШИФРУВАННЯ І КОДУВАННЯ В АРХІТЕКТУРІ RSA-CRT З ПОДВІЙНИМ ЗАХИСТОМ

Запропонована архітектура реалізує комплексний підхід до захисту даних шляхом поєднання криптосистеми RSA, реалізованої на базі RRNS із завадостійким кодуванням. Такий синтез формує дворівневу (двоешелонну) систему захисту, де кожен рівень відповідає за протидію загрозам різного типу.

Рівень 1: Завадостійке кодування для захисту каналу зв'язку. Код Reed-Solomon. Цей рівень призначений для забезпечення цілісності шифротексту (нейтралізації помилок) під час його зберігання або передачі у зашумлених каналах зв'язку. Теоретично для цих цілей можна використовувати широкий клас алгебраїчних кодів, зокрема коди Goppa, які відомі своїм застосуванням у криптографії. У цій практичній реалізації, для забезпечення ефективності та відповідності до стандартів передачі даних, було обрано коди Reed-Solomon, що реалізовані за допомогою бібліотеки galois. Ця реалізація здатна виявити та виправити до t спотворених символів (байтів) у кодовому слові. Якщо кількість помилок у каналі перевищує корекційну здатність коду ($> t$) алгоритм декодування виконає максимально допустиму корекцію, повернувши неправильний, але формально коректний результат.

Рівень 2: Відмовостійкість на основі RRNS. Цей рівень є внутрішнім механізмом захисту самих обчислень і призначений для корекції масштабних помилок на рівні окремих модульних залишків. Такі помилки можуть виникати внаслідок:

– Помилкового декодування на Рівні 1. Код Reed-Solomon повертає некоректно відновлений блок даних, що призводить до появи одного чи кількох повністю хибних зашифрованих залишків C_i'.

– Цілеспрямованої атаки ін'єкції помилок. Зловмисник втручається в апаратне забезпечення, щоб спотворити результат однієї з паралельних операцій модульного піднесення до степеня.

Завдяки наявності надлишкових модулів, система здатна відновити оригінальне повідомлення, навіть якщо частина розшифрованих залишків є хибною. Процедура відновлення використовує CRT на різних комбінаціях залишків та обирає правильний результат за принципом «більшості голосів».

Нижче наведено детальний алгоритм, що описує послідовність операцій шифрування та кодування в рамках запропонованої дворівневої архітектури.

Алгоритм шифрування та кодування на стороні відправника має наступні кроки.

1. Підготовка до RSA-шифрування з використанням RRNS. На цьому етапі обирається набір попарно простих модулів $m_1, m_2, \dots, m_K, m_{K+1}, \dots, m_L$, де перші K модулів утворюють робочий базис, достатній для представлення операндів в операціях RSA, а решта є надлишковими. Для кожного модуля m_i з обраного набору генерується індивідуальна пара ключів RSA: відкритий (e, m_i) та закритий (d_i, m_i). Після цього всі ключові параметри та вихідне повідомлення подаються у форматі RRNS за обраним базисом.

2. RSA-шифрування в базисі RRNS. Відкрите повідомлення M перетворюється у відповідне RRNS-представлення $r_1, r_2, \dots, r_L$, де $r_i = M(mod\ m_i)$. Далі для кожного залишку паралельно виконується операція модульного експоненціювання з використанням відповідного відкритого ключа: $C_i = r_i^e (mod\ m_i)$ для $i = 1, 2, \dots, L$. Результатом є зашифроване повідомлення у вигляді кортежу залишків $C_{RRNS} = C_1, C_2, \dots, C_L$.

3. Двійкове перетворення зашифрованого повідомлення. Отримане RRNS-представлення C_{RRNS} перетворюється у двійковий вектор C_{binary} шляхом конкатенації бінарних еквівалентів кожного залишку C_i.

4. Кодування алгоритмом Reed-Solomon. Для забезпечення цілісності даних двійковий вектор C_{binary} піддається завадостійкому кодуванню Reed-Solomon. Визначаються параметри коду над полем Галуа $GF(2^w)$ (довжина кодового слова n, довжина інформаційної частини k_{RS} та здатність до виправлення t помилок), на основі яких формується породжуючий поліном $g_{RS}(x)$ та породжуюча матриця G. Вектор C_{binary} розбивається на інформаційні блоки відповідної довжини, кожен з яких кодується за допомогою матриці G для отримання кодових слів. Отримані кодові слова конкатенуються у фінальне зашифроване та закодоване повідомлення C_{final}.

Алгоритм декодування та дешифрування на стороні отримувача складається з наступних кроків.

1. Декодування за алгоритмом Reed-Solomon. Прийняте повідомлення C'_{final}, що може містити помилки, розбивається на блоки, які відповідають кодовим словам. До кожного блоку застосовується процедура декодування Reed-Solomon для виявлення та виправлення можливих помилок. З відновлених декодованих блоків реконструюється виправлений двійковий вектор зашифрованого повідомлення C'_{binary}.

2. Зворотне перетворення у формат RRNS. Відновлений двійковий вектор C'_{binary} сегментується на частини, що відповідають розмірності залишків системи. Ці сегменти перетворюються назад у RRNS-представлення $C'_{RRNS} = (C'_1, C'_2, \dots, C'_L)$.

3. RSA-розшифрування в базисі RRNS. Процедура дешифрування виконується паралельно для кожного модульного залишку з використанням відповідного закритого ключа: $r'_i = {C'}_i^{d_i} (mod\ m_i)$ для $i = 1, 2, \dots, L$. Результатом операції є розшифроване повідомлення у вигляді RRNS-представлення $R'_{RRNS} = (r'_1, r'_2, \dots, r'_L)$.

4. Відновлення вихідного повідомлення. За допомогою китайської теореми про залишки (CRT) або її варіантів для RRNS здійснюється зворотне перетворення кортежу R'_{RRNS} у ціле число M'. За умови успішного декодування та виправлення помилок відновлене значення M' повинно дорівнювати вихідному відкритому повідомленню M.

Для наочної візуалізації взаємодії описаних вище рівнів захисту та послідовності операцій, нижче наведено деталізовані блок-схеми алгоритмів. На рис. 1 зображено повний цикл обробки даних на стороні відправника (шифрування та кодування), а на рис. 2 показано зворотний процес на стороні отримувача, що враховує потенційний вплив шуму в каналі передачі.

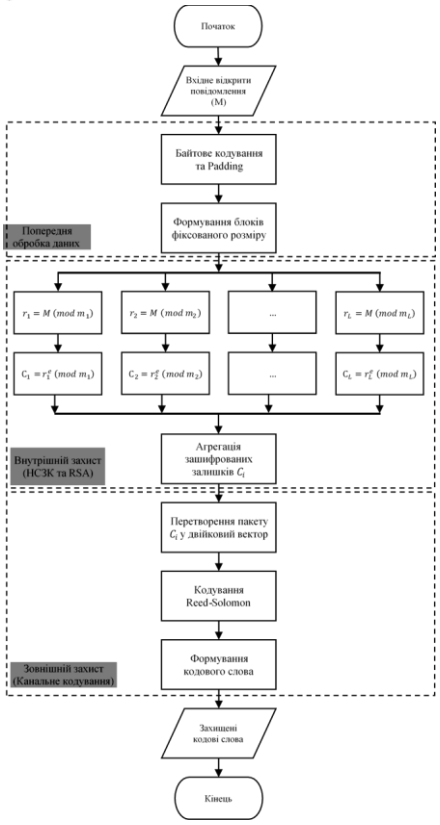

Рис. 1. Блок-схема процесу шифрування та кодування

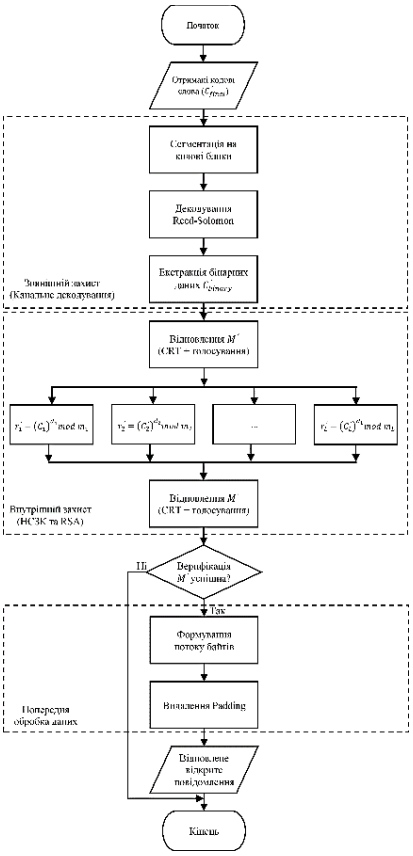

Рис. 2. Блок-схема процесу декодування та дешифрування за умов шуму в каналі

Для демонстрації функціонування запропонованої архітектури наводиться приклад роботи алгоритму в умовах комбінованої атаки. Ілюструється послідовна робота обох рівнів захисту: спочатку завадостійке кодування на основі кодів Reed-Solom on виправляє помилки, внесені в канал передачі, після чого механізм RRNS забезпечує корекцію помилок обчислень, імітуючи стійкість до атак ін'єкції помилок та дозволяючи відновити коректний результат.

Етап 1. Конфігурація та ініціалізація параметрів системи. Перед початком моделювання встановлюються базові параметри підсистем арифметичного захисту у RRNS та канального кодування. Для криптографічного алгоритму RSA визначено бітову довжину ключа, що дорівнює 256, з метою оптимізації часу обчислювальної симуляції в межах експерименту, тоді як у промислових системах за стандартами NIST застосовується довжина від 2048 біт. Структура RRNS передбачає використання 3 робочих модулів, робочий базис яких згенеровано у вигляді [1047...6167, 1035...8219, 7237...9341], та 2 надлишкових модулів [1108...0301, 7629...0917], необхідних для контролю та виправлення помилок. Захист каналу зв'язку забезпечується завадостійким кодом Reed-Solomon з довжиною кодового слова $n = 255$ байтів, інформаційною частиною $k = 235$ байтів та корекційною здатністю $t = 10$ байтів над скінченним полем $GF(2^8)$. Поле задається незвідним поліномом $x^8 + x^4 + x^3 + x^2 + 1$, а структуру коду визначає породжуючий поліном 20-го степеня $g_{RS}(x) = x^{20} + 45x^{19} + 222x^{18} + \cdots + 166x + 89$. Для операцій кодування розраховано породжуючу матрицю G розмірністю (235, 255). На стороні приймача для обчислення синдрому застосовується перевірочна матриця H розмірністю (20, 255).

Етап 2. Процедура шифрування та кодування першого блоку даних. Вихідне текстове повідомлення «Це тестове повідомлення для детальної демонстрації.» конвертується в еквівалентне велике ціле число $M = 49428622\ldots55022487$. На етапі непозиційного представлення дане число розщеплюється на п'ять модульних залишків r_i: [5264...1633, 5626...4847, 4941...0014, 1481...4514, 3006...6753]. Шляхом покомпонентного застосування алгоритму RSA формується вектор зашифрованих залишків C_i: [2864...0256, 1015...3954, 5193...6919, 1010...5778, 5214...348]. Для подальшої трансляції ці п'ять залишків об'єднуються в один інформаційний потік, що вирівнюється нульовими бітами до розміру 235 байтів, набуваючи шістнадцяткового вигляду 3f557d...64e01c000...000. Кодер Reed-Solomon

обчислює 20 байтів надлишковості та додає їх до кінця вектора, формуючи результуюче 255-байтне кодове слово 3f557d...f8e8...805abb.

Етап 3. Декодування та дешифрування з виправленням похибок і збоїв. При симуляції передачі даних у канал зв'язку вноситься 10 випадкових байтових помилок на позиціях [2, 32, 85, 121, 129, 140, 168, 197, 220, 230], унаслідок чого на приймальній стороні фіксується спотворена послідовність 3f55ee...b700...805abb. Оскільки кратність завади не перевищує межу $t = 10$, декодер Reed-Solomon успішно ідентифікує та усуває спотворення, відновлюючи вихідне значення 3f557d...64e01c000...000, що дозволяє реконструювати неушкоджені шифротексти залишків C_i': [2864...0256, 1015...3954, 5193...6919, 1010...5778, 5214...348]. Далі моделюється атака штучного введення збоїв (Fault Injection) шляхом фізичного впливу на пристрій пам'яті, яка призводить до модифікації другого зашифрованого залишку C_2', змінюючи вектор шифротекстів на [2864...0256, 1015...3954, 5193...6919, 1010...5778, 5214...348]. Після виконання паралельного RSA-дешифрування одержують набір розшифрованих залишків r_i', де $r_1' = 5264\ldots1633$ (істинний), $r_2' = 2225\ldots9570$ (хибний), $r_3' = 4941\ldots0014$ (істинний), $r_4' = 1481\ldots4514$ (істинний) та $r_5' = 3006\ldots6753$ (істинний). Для локалізації та усунення помилки запускається процедура мажоритарного голосування в RRNS шляхом перебору комбінацій по 3 базові модулі. Комбінації, що містять аномальний залишок r_2', призводять до некоректних числових значень, тоді як підмножини суто правильних залишків генерують однаковий результат. У результаті голосування правильне числове значення 4942...2487 набирає мажоритарну кількість у 4 голоси, що підтверджує успішну верифікацію даних на основі співвідношення кількості модулів та внесених збоїв.

Для другого інформаційного блоку технологічний цикл повторюється без активування моделі зловмисного впливу, завдяки чому всі 5 дешифрованих залишків залишаються інваріантними. За цих умов усі $\binom{5}{3} = 10$ можливих комбінацій модулів дають тотожний правильний результат, доводячи стабільність криптосистеми у штатному режимі роботи. Шляхом об'єднання та зворотного перетворення блоків повністю відновлюється початковий текстовий масив: «Це тестове повідомлення для детальної демонстрації.».

Запропонована дворівнева архітектура демонструє низку суттєвих переваг порівняно з класичною реалізацією RSA. Ключовою перевагою є комплексний захист та відмовостійкість, оскільки синергія двох рівнів забезпечує стійкість до ширшого спектра загроз – від завад у каналах зв'язку до цілеспрямованих атак ін'єкції помилок – ніж будь-який із механізмів окремо. З точки зору продуктивності, використання системи залишкових класів дозволяє розпаралелити найбільш ресурсомісткі операції модульного піднесення до степеня, замінюючи одну складну операцію на декілька простіших, що виконуються одночасно, і цим суттєво прискорюючи процеси шифрування та дешифрування. Крім того, архітектура є гнучкою та масштабованою. Вона дозволяє адаптувати параметри, такі як кількість базових та надлишкових модулів, а також параметри ECC, зокрема корекційну здатність коду Reed-Solomon, для досягнення оптимального балансу між швидкістю, надійністю та обчислювальними витратами для конкретних умов застосування.

На стороні відправника послідовно виконуються представлення повідомлення в RRNS, паралельне RSA-шифрування та кодування кодом Reed-Solomon. На стороні отримувача відбувається зворотний процес: декодування ECC, паралельне RSA-дешифрування та фінальне відновлення повідомлення за допомогою CRT з процедурою верифікації.

Оцінка практичної ефективності запропонованої архітектури була проведена на основі серії тестів результати яких зафіксовано у Таблиці 1 за наступними сценаріями:

- «Без помилок» – 0 помилок ECC та 0 помилок RRNS;
- «10 помилок ECC» – 10 помилок ECC та 0 помилок RRNS;
- «1 помилка RRNS» – 0 помилок ECC та 1 помилка RRNS).

Таблиця 1

Залежність часу обробки від розміру повідомлення та типу помилок

Розмір повідомлення, (байт)	Сценарій	Час кодування, мс	Час декодування, мс	Загальний час, мс
128	Без помилок	47.9	49.6	97.5
	10 помилок ECC	48.0	50.8	98.8
	1 помилка RRNS	48.1	63.8	111.9
1024	Без помилок	240.5	244.7	485.2
	10 помилок ECC	241.1	246.3	487.4
	1 помилка RRNS	240.3	303.8	544.1

Продовження таблиці 1

4096	Без помилок	967.6	975.3	1942.9
	10 помилок ECC	967.7	976.2	1943.9
	1 помилка RRNS	966.3	1207.9	2174.2

Аналіз експериментальних даних, представлених у Таблиці 1, дозволяє зробити кілька ключових висновків щодо продуктивності розробленої архітектури. По-перше, спостерігається очікуване зростання загального часу обробки зі збільшенням розміру повідомлення. По-друге, механізм ECC у каналі передачі на основі кодів Reed-Solomon демонструє високу ефективність: час декодування повідомлення з 10 помилками ECC лише незначно перевищує час обробки повідомлення без помилок (наприклад, 98.8 мс проти 97.5 мс для 128-байтного блоку).

Натомість активація механізму корекції на рівні RRNS, що є відповіддю на більш серйозні пошкодження даних, пов'язана з помітними обчислювальними витратами. Як видно з таблиці, час декодування при виправленні однієї помилки RRNS суттєво зростає (з 244.7 мс до 303.8 мс для 1024-байтного блоку) через виконання процедури «голосування» (за рахунок перебору комбінацій додаткові ітерації за CRT). Кількісний аналіз накладних витрат показує, що активація механізму відновлення помилки через RRNS додає в середньому близько 12% до загального часу обробки (наприклад, зростання з 485.2 мс до 544.1 мс для блоку 1024 байт). Такий рівень затримки є прогнозованим і виправданим для систем критичної інфраструктури, де цілісність даних превалює над незначним зниженням швидкодії. Тим не менш, ці витрати є виправданою платою за забезпечення відмовостійкості системи проти цілеспрямованих атак та критичних збоїв.

Слід також враховувати, що процедура відновлення методом голосування має комбінаторну складність, яка залежить від кількості поєднань $\binom{K}{L}$. При значному збільшенні кількості модулів L час відновлення зростатиме нелінійно через збільшення кількості ітерацій CRT. Тому обрані в роботі параметри ($K = 3$, $L = 5$) становлять оптимальний баланс між обчислювальною складністю та здатністю виправляти помилки.

Експериментальні дані підтверджують, що архітектура є не тільки надійною, але й ефективною, а витрати часу на корекцію помилок є прогнозованими та прийнятними.

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

У роботі запропоновано архітектуру RSA-CRT із подвійним захистом, яка поєднує два незалежні механізми забезпечення цілісності даних на різних етапах їх обробки. На рівні каналу зв'язку використано коди Reed-Solomon, що забезпечують виявлення та виправлення помилок, які виникають під час передавання повідомлень. Для захисту криптографічних обчислень застосовано надлишкову систему залишкових класів (RRNS), яка дозволяє виявляти та виправляти помилки, індуковані у модульні залишки під час виконання алгоритму RSA-CRT.

Запропоноване архітектурне рішення забезпечує комплексний захист інформації шляхом поєднання механізмів корекції помилок каналу зв'язку та відмовостійкого виконання модульних арифметичних операцій. Використання RRNS дозволяє локалізувати та усунути помилки без необхідності повторного виконання криптографічних обчислень, що підвищує стійкість реалізації RSA-CRT до ін'єкції помилок у модульні залишки..

Результати експериментальних досліджень підтвердили працездатність запропонованої архітектури та можливість її використання для підвищення надійності реалізації RSA-CRT у системах, що працюють в умовах підвищеної ймовірності виникнення випадкових або навмисно індукованих помилок. Поєднання кодів Reed-Solomon та RRNS дозволяє забезпечити захист як даних, що передаються каналом зв'язку, так і результатів внутрішніх криптографічних обчислень, формуючи єдину систему подвійного контролю цілісності.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Vavruk, Ye. Ya., Popovych, B. R., & Popovych, R. B. (2021). Software model of Reed-Solomon codes. *Computer Systems and Networks, 3*(1), 1-6. https://doi.org/10.23939/csn2021.01.001
2. Davletova, A. (2024). Data encryption algorithm with error correction. *Herald of Khmelnytskyi National University. Technical Sciences, 341*(5), 168-176. https://doi.org/10.31891/2307-5732-2024-341-5-26
3. Bernstein, D. J. (2011). List decoding for binary Goppa codes. In Y. M. Chee et al. (Eds.), *Coding and cryptology: Third International Conference, IWCC 2011* (Lecture Notes in Computer Science, Vol. 6639, pp. 62-80). Springer. https://doi.org/10.1007/978-3-642-20901-7_4

4. Chakraborty, A., Bhattacharya, S., Saha, S., & Mukhopadhyay, D. (2020). Rowhammer induced intermittent fault attack on ECC-hardened memory. *IACR Transactions on Cryptographic Hardware and Embedded Systems, 2020*(1), 328-350. https://doi.org/10.46586/tches.v2020.i1.328-350
5. Cintas Canto, A., Kaur, J., Mozaffari Kermani, M., & Azarderakhsh, R. (2023). *Algorithmic security is insufficient: A comprehensive survey on implementation attacks haunting post-quantum security*. arXiv. https://arxiv.org/abs/2305.13544
6. Dottax, E., Giraud, C., Rivain, M., & Sierra, Y. (2009). On second-order fault analysis resistance for CRT-RSA implementations. In O. Markowitch et al. (Eds.), *Information security theory and practice* (Lecture Notes in Computer Science, Vol. 5746, pp. 68–83). Springer. https://doi.org/10.1007/978-3-642-03944-7_6
7. Imam, R., Areeb, Q. M., Alturki, A., & Anwer, F. (2021). Systematic and critical review of RSA based public key cryptographic schemes: Past and present status. *IEEE Access, 9*, 155949-155976. https://doi.org/10.1109/ACCESS.2021.3129224
8. Kim, C. H., & Quisquater, J.-J. (2007). Fault attacks for CRT based RSA: New attacks, new results, and new countermeasures. In D. Sauveron et al. (Eds.), *Information security theory and practice* (Lecture Notes in Computer Science, Vol. 4462, pp. 215-228). Springer. https://doi.org/10.1007/978-3-540-72354-7_18
9. Krasnobayev, V., Yanko, A., & Kovalchuk, D. (2023). Control, diagnostics and error correction in the modular number system. In S. Subbotin (Ed.), *Proceedings of the Sixth International Workshop on Computer Modeling and Intelligent Systems (CMIS-2023)* (CEUR Workshop Proceedings, Vol. 3392, pp. 219–233). CEUR-WS. https://ceur-ws.org/Vol-3392/paper17.pdf
10. Rauzy, P., & Guilley, S. (2014). Countermeasures against high-order fault-injection attacks on CRT-RSA. *Journal of Cryptographic Engineering, 4*(3), 173–185. https://doi.org/10.1007/s13389-013-0065-3
11. Yatskiv, V., Nyemkova, E., Kulyna, S., Kulyna, H., & Ivasiev, S. (2024). Data encryption method based on the redundant residue number system. In S. Caron et al. (Eds.), *Proceedings of the 5th International Workshop on Intelligent Information Technologies and Systems of Information Security (IntelITSIS 2024)* (CEUR Workshop Proceedings, Vol. 3675, pp. 195–204). CEUR-WS. https://ceur-ws.org/Vol-3675/paper16.pdf

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 2 (34), 2026
ISSN 2663 – 4023

Stepan Ivasiev
PhD in Engineering, Associate Professor, Associate Professor of the Department of Cybersecurity West Ukrainian National University, Department of Cybersecurity, Ternopil, Ukraine
ORCID:0000-0003-2243-5956
isv@wunu.edu.ua

Ivan Yanik
PhD Student, Department of Cybersecurity
West Ukrainian National University, Department of Cybersecurity, Ternopil, Ukraine
ORCID:0009-0004-3562-3352
i.yanik@st.wunu.edu.ua

DUAL-LAYER PROTECTED RSA-CRT ARCHITECTURE: GOPPA CODES FOR COMMUNICATION CHANNEL AND RRNS FOR FAULT INJECTION RESISTANCE

Abstract. The growing complexity of cyber threats and the increasing susceptibility of cryptographic hardware to fault injection attacks and communication channel disturbances require the development of integrated fault-tolerant cryptographic architectures. Although the Chinese Remainder Theorem optimization of the RSA algorithm (RSA-CRT) significantly accelerates modular exponentiation, it remains highly vulnerable to computational faults, where even a single induced error may reveal confidential information or compromise the private key. In addition, communication channels are exposed to random and burst errors that reduce the reliability of transmitted cryptographic data. Therefore, ensuring both computational integrity and transmission reliability is an important challenge for modern secure information systems.

This paper proposes a dual-protection architecture for RSA-CRT that combines computational fault tolerance with communication error correction. The first protection layer employs a Redundant Residue Number System (RRNS) consisting of three information moduli and two redundant moduli. The introduced redundancy enables the detection, localization, and correction of erroneous residues through a majority-voting reconstruction procedure while preserving the inherent parallelism of modular arithmetic. The second protection layer utilizes Reed–Solomon error-correcting codes over the Galois field GF(28)GF(2^8)GF(28) to protect transmitted data against channel noise and burst errors, thereby ensuring reliable delivery of encrypted information before cryptographic processing.

A software prototype of the proposed architecture was developed and experimentally evaluated under combined fault scenarios including communication channel errors and injected computational faults. The obtained results demonstrate reliable correction of transmission errors together with successful localization and recovery of corrupted modular residues during RSA-CRT execution. The obtained performance confirms that the proposed architecture achieves a favorable trade-off between fault tolerance and computational efficiency without significantly affecting the throughput of cryptographic operations.

The proposed architecture extends conventional RSA-CRT implementations by integrating two complementary protection mechanisms operating at different stages of information processing. Such an approach improves the robustness of cryptographic systems against both communication errors and computational fault attacks and may be applied in secure embedded systems, industrial controllers, cyber-physical systems, and other security-critical applications requiring high reliability of public-key cryptography.

Keywords: RSA-CRT algorithm, fault injection, redundant residue number system, Reed-Solomon codes, fault tolerance.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Vavruk, Ye. Ya., Popovych, B. R., & Popovych, R. B. (2021). Software model of Reed-Solomon codes. *Computer Systems and Networks, 3*(1), 1-6. https://doi.org/10.23939/csn2021.01.001
2. Davletova, A. (2024). Data encryption algorithm with error correction. *Herald of Khmelnytskyi National University. Technical Sciences, 341*(5), 168-176. https://doi.org/10.31891/2307-5732-2024-341-5-26

3. Bernstein, D. J. (2011). List decoding for binary Goppa codes. In Y. M. Chee et al. (Eds.), *Coding and cryptology: Third International Conference, IWCC 2011* (Lecture Notes in Computer Science, Vol. 6639, pp. 62-80). Springer. https://doi.org/10.1007/978-3-642-20901-7_4
4. Chakraborty, A., Bhattacharya, S., Saha, S., & Mukhopadhyay, D. (2020). Rowhammer induced intermittent fault attack on ECC-hardened memory. *IACR Transactions on Cryptographic Hardware and Embedded Systems, 2020*(1), 328-350. https://doi.org/10.46586/tches.v2020.i1.328-350
5. Cintas Canto, A., Kaur, J., Mozaffari Kermani, M., & Azarderakhsh, R. (2023). *Algorithmic security is insufficient: A comprehensive survey on implementation attacks haunting post-quantum security*. arXiv. https://arxiv.org/abs/2305.13544
6. Dottax, E., Giraud, C., Rivain, M., & Sierra, Y. (2009). On second-order fault analysis resistance for CRT-RSA implementations. In O. Markowitch et al. (Eds.), *Information security theory and practice* (Lecture Notes in Computer Science, Vol. 5746, pp. 68–83). Springer. https://doi.org/10.1007/978-3-642-03944-7_6
7. Imam, R., Areeb, Q. M., Alturki, A., & Anwer, F. (2021). Systematic and critical review of RSA based public key cryptographic schemes: Past and present status. *IEEE Access, 9*, 155949-155976. https://doi.org/10.1109/ACCESS.2021.3129224
8. Kim, C. H., & Quisquater, J.-J. (2007). Fault attacks for CRT based RSA: New attacks, new results, and new countermeasures. In D. Sauveron et al. (Eds.), *Information security theory and practice* (Lecture Notes in Computer Science, Vol. 4462, pp. 215-228). Springer. doi.org/10.1007/978-3-540-72354-7_18
9. Krasnobayev, V., Yanko, A., & Kovalchuk, D. (2023). Control, diagnostics and error correction in the modular number system. In S. Subbotin (Ed.), *Proceedings of the Sixth International Workshop on Computer Modeling and Intelligent Systems (CMIS-2023)* (CEUR Workshop Proceedings, Vol. 3392, pp. 219–233). CEUR-WS. https://ceur-ws.org/Vol-3392/paper17.pdf
10. Rauzy, P., & Guilley, S. (2014). Countermeasures against high-order fault-injection attacks on CRT-RSA. *Journal of Cryptographic Engineering, 4*(3), 173–185. https://doi.org/10.1007/s13389-013-0065-3
11. Yatskiv, V., Nyemkova, E., Kulyna, S., Kulyna, H., & Ivasiev, S. (2024). Data encryption method based on the redundant residue number system. In S. Caron et al. (Eds.), *Proceedings of the 5th International Workshop on Intelligent Information Technologies and Systems of Information Security (IntelITSIS 2024)* (CEUR Workshop Proceedings, Vol. 3675, pp. 195–204). CEUR-WS. https://ceur-ws.org/Vol-3675/paper16.pdf

Отримано редакцією журналу / Received: 02.02.26
Прорецензовано / Revised: 14.02.26
Схвалено до друку / Accepted: 24.09.26