УДК 004.056.55

Туржанська І. Д.

ПОРІВНЯЛЬНИЙ АНАЛІЗ СУЧАСНИХ АЛГОРИТМІВ ГОМОМОРФНОГО ШИФРУВАННЯ ДЛЯ ЗАБЕЗПЕЧЕННЯ КОНФІДЕНЦІЙНОСТІ ДАНИХ

Вінницький національний технічний університет

Анотація

Проведено порівняльний аналіз сучасних методів гомоморфного шифрування з метою покращення захисту даних, розглянуто характеристики, переваги та недоліки алгоритмів гомоморфного шифрування. Сформовано рекомендації щодо застосування конкретних алгоритмів.

Ключові слова: гомоморфне шифрування, алгоритми гомоморфного шифрування, порівняльний аналіз, рекомендації щодо застосування.

Abstract

A comparative analysis of modern homomorphic encryption methods has been conducted to enhance data protection. The characteristics, advantages, and disadvantages of various homomorphic encryption algorithms were examined. Based on this study, recommendations for the application of specific algorithms have been formulated.

Keywords: homomorphic encryption, homomorphic encryption algorithms, comparative analysis, recommendations for application.

Вступ

Наразі більшість аспектів повсякденного життя повинні забезпечувати надійність обробки, зберігання та передачі даних. Натомість дані, які використовуються, не мають властивість збереження значень, якими володіли на момент шифрування. Переважна більшість математичних операцій над шифротекстом призводить до зміни значення відповідного відкритого тексту. Можливість виконувати математичні операції над зашифрованими даними означає, що між відкритим текстом та шифротекстом має існувати певний зв'язок. Але цей зв'язок повинен залишатися непомітним для стороннього спостерігача. Можливість отримання інформації про відкритий текст шляхом аналізу операцій над шифротекстом призводить до зламаного шифрування. Вирішити одночасно ці два завдання – захисту даних і можливості виконувати обчислення над зашифрованою інформацією з отриманням справжнього результату – досить складно. Одним із сучасних шляхів одночасного вирішення вказаних проблем є використання гомоморфного шифрування. Метою роботи є покращення захисту даних шляхом розроблення на основі порівняльного аналізу рекомендації щодо застосування методів гомоморфного шифрування.

Результати дослідження

Гомоморфне шифрування — така модель шифрування, яка дозволяє виконувати певні математичні дії з зашифрованим текстом і отримувати зашифрований результат, який відповідає результату аналогічної операції, що проводиться з відкритим текстом. [1]

Процес розробки алгоритму гомоморфного шифрування є надто складним. Наразі існують деякі типи гомоморфного шифрування**,** які характеризують, наскільки близьким є певний алгоритм для досягнення зазначеної мети [3]:

1**.** Частково гомоморфне шифрування (PHE). У частково гомоморфних алгоритмах шифрування можна необмежену кількість разів виконувати лише одну певну операцію**.** Частково гомоморфні алгоритми шифрування є відносно простими у розробці. Деякі поширені алгоритми шифрування випадково мають властивість часткової гомоморфності.

2**.** Обмежено гомоморфне шифрування (SHE). Такий алгоритм дозволяє виконувати скінченну кількість довільних операцій**,** а не нескінченну кількість однієї конкретної операції. Наприклад, обмежено гомоморфний алгоритм виконує будь-яку комбінацію з п’яти операцій додавання або

множення. Однак виконання шостої операції будь-якого з цих типів призведе до отримання некоректного результату.

3. Повністю гомоморфне шифрування (FHE). Цей тип шифрування є ключовим етапом у розвитку гомоморфного шифрування. Повністю гомоморфний алгоритм шифрування дозволяє виконувати необмежену кількість операцій додавання та множення над шифротекстами і при цьому завжди отримувати коректний результат.

Більшість сучасних схем повністю (FHE) та обмежено (SHE) гомоморфного шифрування, що базуються на задачах навчання з помилками (LWE/RLWE), використовують поняття «шуму». Кожен шифротекст у таких схемах містить певну похибку, яка зростає з кожною виконаною математичною операцією. Шифротексти розшифровуються коректно лише доти, доки рівень шуму не перевищує критичну межу. Для подолання цього обмеження та переходу від обмежено гомоморфних схем до повністю гомоморфних використовується процедура bootstrapping (самовідновлення). Вона полягає у гомоморфному обчисленні процедури розшифрування, що дозволяє «очистити» шум і виконувати необмежену кількість операцій. Схеми, що не підтримують або не використовують bootstrapping, зазвичай є обмежено гомоморфними (SHE), тоді як його впровадження дозволяє досягти повної гомоморфності (FHE). На відміну від них, класичні частково гомоморфні схеми (PHE), такі як RSA чи Paillier, не використовують шум і не потребують процедури bootstrapping для виконання відповідних операцій

Основні характеристики сучасних алгоритмів гомоморфного шифрування представлено у таблиці 1.

Таблиця 1

Основні характеристики сучасних алгоритмів гомоморфного шифрування

Характеристика	**BGV** (Brakerski-Gentry-Vaikuntanathan)	**BFV** (Brakerski-Fan-Vercauteren)	**FHEW** (Fast Homomorphic Encryption over the Torus)	**TFHE** (Torus Fully Homomorphic Encryption)	**CKKS** (Cheon–Kim–Kim–Song)	**RSA** (Rivest–Shamir–Adleman)	**Paillier** (Paillier probabilistic)
Тип	без bootstrapping є SHE, з bootstrapping є FHE	без bootstrapping є SHE, з bootstrapping є FHE	FHE	FHE	FHE	PHE	PHE
Рік розроблення	2011	2012	2014	2016	2017	1978	1999
Тип операцій	Add, Mult, MultConst, Eval $(f,c_1,..,c_n)$	Add, Mult, MultConst, Eval $(f,c_1,..,c_n)$	бітові: AND, OR, XOR, NOT	бітові: AND, OR, XOR, NOT	Add, Mult, MultConst, Eval $(f,c_1,..,c_n)$	Mult	Add, MultConst,
Довжина ключа	1024–8192 біт	1024–8192 біт	8192–16384 біт	8192–16384 біт	2048–16384 біт	1024–4096 біт	1024–4096 біт

З таблиці 1 можна зробити висновки, що BGV, BFV призначені для цілих чисел та є найбільш використовувані схеми для практичного FHE; FHEW, TFHE є дуже швидкими, які краще використовувати для бітових обчислень, CKKS використовують для приблизних обчислень з дійсними числами, RSA і Paillier мають обмежені операції.

Всі сучасні схеми гомоморфного шифрування (BGV, BFV, FHEW, TFHE, CKKS) є стійкими до атак із застосуванням квантових комп'ютерів. В той час, коли класичні криптосистеми (RSA, Paillier) є вразливими до квантових алгоритмів і забезпечують захист лише в межах класичних атак.

Аналіз практичних застосувань [4] показав, що вибір конкретного алгоритму гомоморфного шифрування істотно залежить від прикладної області. В першу чергу це визначається операціями над даними, які можуть бути виконані без потреби розшифрування даних. Так, для задач обчислень доз медичних препаратів, розглянутих в роботі [4], було достатньо методу Paillier, який належить до типу PHE і на перший погляд поступається більш пізнім схемам. Проте можливостей, які надає цей метод, виявилось достатньо для того, щоб врахувати особливості предметної області. Крім того, при виборі практичного застосування методу гомоморфного шифрування важливо зважати на очікувану стійкість та вимоги до швидкості обробки даних.

Ще одним фактором, який потрібно брати до уваги під час планування впровадження методу гомоморфного шифрування є наявність його реалізацій в програмних бібліотеках і фреймворках, які ви-

користовуються для розроблення відповідних програмних продуктів. Це пов'язано з тим, що розроблення криптографічного модуля "з нуля" та його відповідне тестування і зневадження можуть суттєво вплинути на загальну швидкість реалізації всього проєкту.

У цьому дослідженні внаслідок порівняльного аналізу визначено переваги та недоліки сучасних алгоритмів гомоморфного шифрування, на основі яких розроблено рекомендації їх застосування (таблиця 2).

Таблиця 2

Переваги, недоліки та рекомендації щодо застосування алгоритмів гомоморфного шифрування

Алгоритми гомоморфного шифрування	**BGV**	**BFV**	**FHEW**	**TFHE**	**CKKS**	**RSA**	**Paillier**
Переваги	цілочисельна арифметика; швидке скалярне множення; швидкі лінійні функції, ефективне багаторівневе проектування	побітові операції, ефективні булеві схеми, швидке відновлення, швидке порівняння чисел	найшвидша процедура bootstrapping (менше 0.1 сек); ефективність для побітових логічних операцій	арифметика дійсних чисел, швидке наближення поліномів, швидкий обернений множник, ефективна логістична регресія, багаторівневе проектування	арифметика дійсних чисел, швидке наближення поліномів, ефективна логістична регресія, багаторівневе проектування	математична простота та висока швидкість операцій шифрування; мінімальні витрати на розмір шифротексту	ефективна адитивна гомоморфність (додавання); підтримка необмеженої кількості операцій додавання; компактність ключів
недоліки	обчислювальна складність процедури bootstrapping, значні часові витрати на реалізацію нелінійних функцій	повільніше за BGV при зростанні глибини обчислень, швидке зростання шуму при множенні	низька ефективність для виконання арифметичних операцій над цілими числами, великий обсяг оперативної пам'яті для зберігання ключів перемикання	обмежена ефективність для складних арифметичних операцій	повільне bootstrapping, наближений характер обчислень	підтримка лише мультиплікативної гомоморфності (неможливість додавання)	підтримка лише адитивної гомоморфності (неможливість множення двох шифротекстів)
рекомендації щодо застосування	обчислення над цілими числами (статистика), векторні обчислення, лінійні алгоритми машинного навчання	обчислення над цілими числами (статистика, приватні фінансові), обробка великих масивів чисел	логічні обчислення, побітові операції в приватних алгоритмах з високою частотою оновлення шуму	логічні обчислення та цифрові схеми, побітові операції в приватих алгоритмах	обчислення з дійсними числами та машинне навчання, обробка великих масивів даних, аналіз даних з наближеннями	мультиплікативні обчислення над великими цілими числами, перевірка цілісності даних	адитивні обчислення над цілими числами, агрегація статистичних даних
найпоширеніші фреймворки	Helib, Seal, Palisade, OpenFHe	Seal, Palisade, Lattigo, FV-NFLib	Palisade, OpenFHe, FHEW	TFHE Library, Concrete, Palisade, HElib	Helib, Seal, Palisade, Lattigo, Neaan, OpenFHe	OpenSSL, Crypto++, PyCrypto, Bouncy Castle	Helib, Pyfhel, JavaPaillier, Palisade
мова	C++, C#	C++, C#	C++	C, C++, Python	C++, C#	C, C++, Python, Java	C, C++, Python, Java, JavaScript

За допомогою гомоморфного шифрування можна вирішити певні бізнес, медичні та економічні проблеми, але існують певні ризики та недоліки, серед яких:

1. Висока обчислювальна складність. Наприклад, обчислення, що займає 1 секунду на відкритому тексті, може тривати 12 днів на зашифрованих даних [3].
2. Велике споживання пам'яті. Зашифровані дані займають у десятки разів більше місця, ніж оригінальні.
3. Складність використання. Алгоритми складні для налаштування та потребують спеціалізованих знань.

Висновки

Повністю гомоморфне шифрування — це сучасна технологія, яка на сьогодні є занадто повільною і ресурсозатратною. Проте здатність виконувати обробку інформації в зашифрованому вигляді дозволяє

розв’язувати низку специфічних задач, в яких висуваються підвищені вимоги до захисту конфіденційності інформації. Так розглянуті методи шифрування можуть сприяти підсиленню розмежування прав доступу — персони, які оновлюють дані, можуть це робити без отримання доступу до самих даних у відкритому вигляді.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Гніденко М. П., Жебеленко М. Г. Криптографічні методи захисту інформації : навч. посіб. Київ : Державний університет телекомунікацій, 2017. - 132 с.
2. Kupcova E., Pleva M., Khavan V., Drutarovsky M. A Comparative Study of Partially, Somewhat, and Fully Homomorphic Encryption in Modern Cryptographic Libraries. Electronics. 2025. Vol. 14, No. 23. URL: https://doi.org/10.3390/electronics14234753 (last accessed: 03.03.2026)
3. Gentry, C. Fully homomorphic encryption using ideal lattices. Proceedings of the 41st Annual ACM Symposium on Theory of Computing. – 2009. - P. 169–178. URL: https://doi.org/10.1145/1536414.1536440. (last accessed: 03.03.2026)
4. Baryshev Y., Lanova V. Method of Patients’ Data Protection on the Instance of Chemotherapy Dosing Data for Ewing’s Sarcoma Treatment. Proceedings of the 7th International Conference on Informatics & Data-Driven Medicine. Birmingham, United Kingdom, November 14-16, 2024. Pp. 81-91. URL: https://ceur-ws.org/Vol-3892/short2.pdf (last accessed: 03.03.2026)

Туржанська Ірина Дмитрівна - студентка групи 2БС-22Б, факультет інформаційних технологій і комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: turzhanskayaryna@gmail.com

Науковий керівник***: Баришев Юрій Володимирович*** – к.т.н., доцент кафедри захисту інформації, Вінництский національний технічний університет, м. Вінниця, email: yuriy.baryshev@vntu.edu.ua

Iryna Turzhanska - student of group 2BS-22B, Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: turzhanskayaryna@gmail.com

Scientific supervisor: ***Yurii Baryshev*** – PhD (Eng), Associated Professor of Information Protection Department, Vinnytsia National Technical University, Ukraine, yuriy.baryshev@vntu.edu.ua