УДК 004.056.55

Луценко Владислав Ігорович,

магістр Харківського національного університету радіоелектроніки

Олешко Інна Вікторівна,

к.т.н., доцент кафедри безпеки інформаційних технологій Харківського національного університету радіоелектроніки

ПОРІВНЯЛЬНА ОЦІНКА ОПТИМІЗОВАНОГО ПІДХОДУ ГОМОМОРФНОГО ШИФРУВАННЯ ДЛЯ КОНФІДЕНЦІЙНОГО АНАЛІЗУ ТЕЛЕМЕТРІЇ МЕТОДАМИ МАШИННОГО НАВЧАННЯ

Анотація. У роботі обґрунтовано оптимізований підхід до конфіденційного аналізу системної телеметрії методами машинного навчання на основі гомоморфно зашифрованих даних. Досліджено непрактичність базового гомоморфного шифрування для потокового моніторингу через тривалу затримку обчислень і великий обсяг шифротекстів. Доведено, що використання стовпчикового SIMD пакування та попереднього згортання нормалізації у ваги моделі прискорює аналіз з 19000 до 0,19–1,54 мікросекунд на запис і стискає шифротексти до 0,8 мегабайт без втрати точності виявлення аномалій із показником NAB Score на рівні 44,79. Запропонований підхід дозволяє виконувати моніторинг інфраструктури без розшифрування телеметрії на сервері, зменшує навантаження на мережевий трафік та зберігає конфіденційність метрик під час їх обробки зовнішніми сервісами машинного навчання.

Ключові слова: гомоморфне шифрування, системна телеметрія, машинне навчання, виявлення аномалій.

Зростання обсягів системної телеметрії хмарних і розподілених інфраструктур зумовлює потребу в автоматизованому аналізі даних методами машинного навчання, що дає змогу виявляти аномалії та своєчасно реагувати на інциденти інформаційної безпеки. Водночас передавання телеметрії зовнішнім сервісам машинного навчання створює ризик розкриття конфіденційної інформації, оскільки метрики навантаження відображають архітектуру інфраструктури та моменти пікових навантажень. Гомоморфне шифрування дає змогу виконувати обчислення над зашифрованими даними без їх розкриття на сервері. Для задач машинного навчання підходить схема CKKS [1], яку реалізовано в бібліотеці TenSEAL [2].

Пряме застосування гомоморфного шифрування до аналізу телеметрії великої кількості вузлів має практичні обмеження. Якщо кожен запис шифрувати окремо, ємність схеми використовується неефективно, що збільшує затримку обробки та обсяг даних, що передаються мережею. Нелінійні функції моделі додатково ускладнюють обчислення, оскільки CKKS підтримує лише додавання та множення. Альтернативна схема TFHE [3], реалізована в ConcreteML, має інші компроміси між швидкістю та підтримкою пакетної обробки.

Для подолання обчислювальних обмежень в роботі запропоновано оптимізацію лінійного класифікатора логістичної регресії. Оскільки сирі метрики телеметрії мають різний масштаб, під час навчання моделі застосовуються Z-нормалізація вхідних ознак із x_j із середнім значенням μ_j та стандартним відхиленням σ_j:

$$z = \sum_{j=1}^{d} w_j * \left(\frac{x_j - \mu_j}{\sigma_j} \right) + b \qquad (1)$$

У разі прямого обчислення за формулою (1) масштабування виконується безпосередньо над шифротекстами, що витрачає один рівень мультиплікативної глибини, через що сумарна розрядність ланцюжка модулів зростає і перевищує ліміт у 218 біт для кільця розмірності N = 8192 [4], змушуючи переходити до поліноміального модуля N = 16384. Для усунення гомоморфного ділення під час обробки потоку, параметри нормалізації згортаються безпосередньо у ваги та зсув класифікатора ще на етапі його підготовки:

$$w_j' = \frac{w_j}{\sigma_j};\ b' = b - \sum_{j=1}^{d} 1 \frac{w_j \mu_j}{\sigma_j} \qquad (2)$$

З використанням перетворення (2) сервер обчислює лінійне значення

z виключно множенням шифротекстів на відкриті скаляри, які за властивістю CKKS не втрачають мультиплікативних рівнів[1], що дозволяє працювати в кільці розмірності N = 8192, скорочуючи удвічі розмір поліномів.

Зі зменшенням глибини схеми також вирішується неефективність використання криптографічної ємності при пакуванні телеметрії. У разі шифрування окремих записів вектор із чотирьох метрик займає лише 4 із 8192 доступних слотів і вимагає близько 70 мегабайт ключів Галуа для міжслотових зсувів під час скалярного добутку.

Натомість застосовано стовпчикове SIMD пакування. Пакет із 4096 зрізів телеметрії шифрується у чотири окремі шифротексти для кожної з метрик $C_1 \dots C_4$, завдяки чому шифротекст вміщує рівно $\frac{N}{2} = 4096$ слотів та досягає повного заповнення ємності схеми.

Оскільки CKKS підтримує лише додавання та множення, сигмоїдна функція $\sigma(z) = \frac{1}{1+e^{-z}}$ не може бути обчислена напряму. Для розв'язання цієї проблеми запропоновано два підходи.

За кубічної апроксимації SIMD Poly сервер оцінює зашифровані ймовірності за допомогою непарного полінома $\hat{\sigma}_3 = 0{,}5 + \mathrm{c}_1 z + \mathrm{c}_3 z^3$ [1]. Межі інтервалу наближення підібрано із запасом щодо максимальних значень навчальної вибірки, що мінімізує крайові спотворення полінома.

Підхід класифікації за знаком логіта SIMD logit усуває обчислення сигмоїд на сервері. Спираючись на еквівалентність $\sigma(z) \geq 0.5 \iff z \geq 0$, сервер повертає клієнту єдиний шифротекст лінійних логітів C_z, що виключає гомоморфне множення шифротекстів між собою та знижує мультиплікативну глибину до 1.

Експерименти проведено на сервері з 10-ядерним процесором та 16 ГБ ОЗП на телеметрії AWS CloudWatch з вибіркою по 4096 записів. Для порівняльного аналізу оцінено запропоновані конфігурації кубічної

апроксимації ймовірностей SIMD Poly, пряму класифікацію за знаком логіта SIMD logit, базове пооб'єктне шифрування та модель схеми TFHE ConcreteML.

Таблиця 1 - Порівняльна оцінка підходів гомоморфного шифрування для пакета з 4096 записів телеметрії

Метод	Схема	Час на 4096 (мс)	Час на 1 запис (мс)	Обсяг даних (МБ)	Ключі Оцінки (МБ)	NAB Score
Базовий текст	–	0,14	0,000037	0,06	0	44,79
Базове CKKS	CKKS	78149	19,08	3643,16	70,48	44,79
SIMD Poly	CKKS	6,29	0,0015	1,46	1,91	44,79
SIMD Logit	CKKS	0,81	0,00020	0,80	0	44,79
ConcreteML	TFHE	52451	12,81	1,41	0,00006	44,79

Результати в таблиці 1 показують, що схема TFHE добре стискає дані та ключі оцінки, проте через відсутності пакетної обробки, TFHE аналізує запити поодинці. Запропоноване стовпчике пакування в схемі CKKS обробляє всі 4096 запитів одночасно, через що випереджає TFHE за швидкістю. Показник NAB Score[5] для всіх підходів становить 44,79, що підтверджує збереження точності виявлення аномалій при шифруванні. На телеметрії з AWS CloudWatch запропоновані рішення скорочують час серверного аналізу пакета з 78 секунд до 6,29 та 0,81 мілісекунд відповідно. Мережевий трафік шифротекстів вдалося зменшити з 3643,16 мегабайт до 1,46 та 0,80 мегабайт для SIMD Poly та SIMD Logit методів відповідно.

Запропонований підхід дозволяє здійснювати конфіденційний аналіз телеметрії методами машинного навчання із мінімальним навантаженням на мережу та без зниження точності детектування інцидентів інформаційної безпеки. Порівняно з базовим гомоморфним шифруванням, метод SIMD Poly скорочує час серверної обробки приблизно у 12,4 тисячі разів, а обсяг мережевого трафіку у 2,5 тисячі разів, тоді як метод SIMD Logit скорочує їх

у 96,5 тисячі та 4,5 тисячі разів відповідно. Отримані результати підтверджують ефективність запропонованого підходу для конфіденційного аналізу телеметрії під час її обробки зовнішніми сервісами.

Список використаної літератури

1. Cheon J. H., Kim A., Kim M., Song Y. Homomorphic encryption for arithmetic of approximate numbers. Cryptology ePrint Archive, Report 2016/421. 2016. URL: https://eprint.iacr.org/2016/421.
2. TenSEAL: A Library for Encrypted Tensor Operations Using Homomorphic Encryption / A. Benaissa, B. Retiat, B. Cebere, A. E. Belfedhal. ICLR 2021 Workshop on Distributed and Private Machine Learning (DPML). 2021. 14 p. DOI: https://doi.org/10.48550/arXiv.2104.03152.
3. Meyre, A., Chevallier-Mames, B., Frery, J., Stoian, A., Bredehoft, R., Montero, L., Kherfallah, C.: Concrete-ml (2022). URL: https://github.com/zama-ai/concrete-ml
4. Homomorphic Encryption Standard / M. Albrecht, M. Chase, H. Chen et al. Toronto: HomomorphicEncryption.org, 2018. 37 p. URL: https://homomorphicencryption.org/standard
5. The Numenta Anomaly Benchmark (NAB): open-source benchmark for streaming anomaly detection: GitHub repository. URL: https://github.com/numenta/NAB