КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

Факультет комп'ютерних наук та кібернетики

Кафедра теоретичної кібернетики

Кваліфікаційна робота

На здобуття ступеня бакалавра

за спеціальністю 122 Комп'ютерні науки

на тему:

Гомоморфне шифрування для захисту даних в хмарних та туманних технологіях

Виконав студент 4-го курсу
Мальований Дмитро Борисович _____ (підпис)

Науковий керівник:
професор, доктор фіз-мат. наук
Пашко Анатолій Олексійович _____ (підпис)

Засвідчую, що в цій роботі немає запозичень праць інших авторів без відповідних посилань.

Студент _____ (підпис)

Роботу розглянуто й допущено до захисту
на засіданні кафедри
теоретичної кібернетики
"____"_____ 2023р
протокол №____
Завідувач кафедри
Крак Юрій Васильович _____ (підпис)

Київ - 2023

РЕФЕРАТ

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

FHE – Fully homomorphic encryption (Повне гомоморфне шифрування).

SHE – Somewhat homomorphic encryption scheme

RSA – Криптографічний алгоритм з відкритим ключем, який базується, розрахунковій складності великих полупростих чисел.

PKI – Public key infrastructure - набір інструментів які використовують пару (приватний, публічний) ключ, та в якій між користувачами передається тільки публічні ключі, залишаючи приватні анонімними.

Boolean Circuit – Булева схема - це математична модель, що використовується для представлення та обробки булевих функцій. Вона складається з логічних елементів, які з'єднані між собою для виконання логічних операцій над двійковими входами $\{0, 1\}$ та формування двійкових виходів. Схема складається з взаємопов'язаних логічних елементів, таких як (AND, NOT, OR).

ВСТУП

FHE або повне гомоморфне шифрування, це тип шифрування яке дозволяє виконувати розрахунки на зашифрованих даних, не вимагаючи, щоб вони були розшифровані для цього. Результатом розрахунків або ж гомоморфної операції над даними є зашифровані дані, які можуть бути розшифровані ключем з тої ж самої пари, з якої вони були зашифровані.

Завдяки особливості виконувати операції над зашифрованими даними без попереднього дешифрування, FHE стає гарним рішенням в задачах передачі даних в незахищених середовищах, в не авторизованих середовищах, або при передачі над чутливих даних, які не повинні бути видимі для сторони яка займається їх обробкою.

Оцінка сучасного стану об'єкта дослідження або розробки

Вперше технологія FHE була запропонована в 1978 в році, але майже 30 років не було авторитетних досліджень на цю тему і на той момент вже існувала система RSA, яка була краща за багатьма параметрам. Починаючи з 2009 року дослідження та розробки на тему гомоморфного шифрування дуже актуальні й розвиток цієї технології відбувається надзвичайно швидко, покращуючи швидкість виконання операцій, швидкість дешифрування, та розширюючи область застосування шляхом додавання більш комплексних гомоморфних операцій. На цей час існує багато рішень на типові проблеми з використовуванням FHE, які конкурують між собою в різних аспектах, та постійно розвиваються.

Актуальність роботи та підстави для її виконання

Безпечність передачі даних в незахищених середовищах та авторизація отримувача були завжди дуже важливими, рідко хто нехтує цим, оскільки не хоче, щоб їх данні були скомпрометовані або перехоплені. Окрім цього все частіше, за потребою складних обчислень, користувачі звертаються до віддалених машин, також відомі як хмари. Звісно кожен користувач хоче, щоб їх данні були захищенні під час передачі, та хоче бути впевнений, що він передає данні саме туди, куди планував.

Для забезпечення вище описаних вимог, користувач використовує чинні технології, такі як PKI. Єдина не вирішена проблема PKI або інших технологій, це вимога повного дешифрування даних, це означає що при отриманні зловмисником доступу до хмари або віддаленого сервера, у нього буде доступ до не зашифрованих даних. Хоча сучасні хмари дуже добре захищенні, розраховувати на те що зловмисник не зможе отримати до них доступ - не варто.

Для розв'язання проблеми, яку не вирішує PKI, чудово підходить FHE, оскільки сервер, зберігає і виконує операції над даними в зашифрованому вигляді, тому навіть якщо зловмисник отримає доступ до сервера або хмари, отримати дані в нього не вдасться.

Звісно є і деякі обмеження у використанні FHE: по-перше, операції над даними обов'язково повинні бути гомоморфні, по-друге, алгоритм застосування операції над зашифрованими даними дуже повільний. Якщо задача вимагає обробку великої кількості даних, або операція повинна бути не гомоморфна, то можливо краще подумати в сторону застосування інших криптосистем.

Мета й завдання роботи

Мета роботи дослідити інсуючі гомоморфні схеми, визначити їх криптографічну схему, обмеження та можливі області застосування. Також, необхідно показати імплементації існуючих повних та частково гомоморфних схем, описати їх вразливості, аналітично порівняти описані схеми.

Також метою роботи є засвідчення того що гомоморфне шифрування застосоване до задач безпечної передачі даних у хмарних та туманних технологіях. Завдання полягає в тому, щоб показати теоретично та практично, що

дані користувача можуть бути безпечно передані та оброблені хмарою, без розкриття цих даних для хмари.

Також необхідно перевірити результати практичного використання FHE на коректність, та порівняти накладні витрати, по часу та пам'яті, виконання операції над зашифрованими даними, та над не зашифрованими.

Об'єкт і методи дослідження

Для дослідження коректності та застосованості FHE і практичної реалізації системи з використанням технології FHE, було вибрано клієнт-серверний застосунок, де сервер буде виконувати роль хмари, та з'єднання клієнта з сервером відбувається в незахищеному середовищі.

Областю реалізації буде спрощена банківська система, де хмара буде виконувати роль банку, який дозволяє користувачу додавати, знімати, та передивлятись свій баланс віртуальних грошей. При цьому серверний застосунок повинен бути реалізований таким чином, що він не буде знати нічого, ні про користувача, а ні про то скільки умовного баланса у певного клієнта. Для цього він буде зберігати данні зашифровані FHE у внутрішній базі даних, та публічний ключ клієнта для виконання гомоморфних операцій над даними.

Ця система повинна чудово показати всю силу гомоморфного шифрування: тільки клієнт, який створив баланс за допомогою свого приватного ключа, буде мати можливість мати доступ до свого балансу, як переглядати його, так і виконувати над ним певні операції. Всі інші учасники та користувачі системи не матимуть доступу до даних, що забезпечує їх повну безпеку.

Більш детально про об'єкти та методи дослідження буде описано в другому розділі роботи, фрагменти реалізації будуть наведені в додатках до роботи.

Можливі сфери застосування

Гомоморфне шифрування може бути застосоване в будь-якій сфері де потрібна обробка даних, та для виконання цієї задачі використовується віддалений сервер, або хмара. Використання FHE, гарантує безпечну передачу та обробку без попереднього дешифрування даних, але при цьому накладає обмеження на операцію обробки, яка повинна бути гомоморфна, та значно

знижує час обробки.

Більш детально ця тема буде розкрита в відповідному розділі, де будуть описані як повноцінні області застосування FHE, так і використання FHE як інструмент для створення більш комплексних криптосистем, та інструментів.

Розділ 1
ОГЛЯД ТЕХНОЛОГІЇ FHE

1.1 Визначення

В цій секції описана термінологія, яка використовується в дослідженнях FHE. Деякі з визначень були взяті напряму з документів FHE, інші були перефразовані для того, щоб спростити формальність і зробити їх більш застосованими до обраної задачі.

Нехай є простір вхідного (чистого) тексту $\mathcal{P} = \{0, 1\}$, та сімейства функцій $F = f_1, f_2, ..., f_n$ де $f_n(x) = f(x_1, x_2, x_3, ..., x_k)$ це Булеві функції k аргументів: $f : P^n \to P$. Ми будемо називати F, сімейством Булевих схем (Boolean circuit) C, і використовувати звичайний запис функції $C(m_1, m_2, ..., m_n)$, для позначення оцінки Булевої схеми на кортежі $(m_1, m_2, ..., m_n)$.

Визначення 1.1.1 ($\mathcal{C}$-схема розрахунків, або ж просто $\mathcal{C}$-схема [4])**.** Нехай $\mathcal{C}$ це множина Булевих схем, тоді $\mathcal{C}$-схема розрахунків, для $\mathcal{C}$ це набір функцій (GEN, ENC, EVAL, DEC) які задовільняють наступним твердженням:

GEN$(1^\lambda, \alpha)$ - алгоритм генерації ключів, на вхід він приймає, параметр шифрування λ, та допоміжний параметр α. Результат виконання алгоритму це триплет ключів (pk, sk, evk), де ключ pk використовується для шифрування, sk для дешифрування, та evk для виконування розрахунків.

ENC(pk, m) - алгоритм шифрування, на вхід він приймає ключ шифрування pk та фрагмент не зашифрованого (чистого) тексту m. Результат виконання алгоритму це шифр c.

EVAL$(evk, C, c_1, c_2..., c_n)$ - алгоритм розрахунків. На вхід він отримує, ключ розрахунків evl та Булеву схему $C \in \mathcal{C}$, та вхідні аргументи, які можуть бути як шифром, так і результатом виконання минулих розрахунків. Результат виконання алгоритму це результат виконання розрахунків.

DEC(sk, c) - алгоритм дешифрування. На вхід приймає, ключ дешифрування sk, та шифр, або результат виконання розрахунків. Результат виконання алгоритму це не зашифрований (чистий) текст m.

Для подальшого опису властивостей, треба визначити простори даних, які є результатами, або вхідними параметрами описаних алгоритмів:

Нехай $\mathcal{X}$ буде описувати простір *чистого шифру*, $\mathcal{Y}$ - простір результатів виконання розрахунків, і $\mathcal{Z} = \mathcal{X} \cup \mathcal{Y}$. $\mathcal{Z}^*$ - містить кортежі довільної довжини, які складаються з елементів $\mathcal{Z}$. Простори ключів згенерованих **GEN**, позначимо як $\mathcal{K}_p, \mathcal{K}_s, \mathcal{K}_e$ для pk, sk, evk відповідно.Алгоритм **GEN** приймає на вхід параметр в унарній нотації 1^λ та опціональний допоміжний параметр λ з простору $\mathcal{A}$. Також, $\mathcal{C}$ містить простір *дозволених* булевих схем, а $\mathcal{P}$, як було зазначено раніше, область вхідного *чистого (незашифрованого)тексту.*

Тепер можна описати область роботи наведених вище алгоритмів:

$$\begin{aligned}
&\textbf{GEN} : \mathbb{N} \times \mathcal{A} \to \mathcal{K}_p \times \mathcal{K}_s \times \mathcal{K}_e \\
&\textbf{ENC} : \mathcal{K}_p \times \mathcal{P} \to \mathcal{X} \\
&\textbf{EVAL}: \mathcal{K}_e \times \mathcal{C} \times \mathcal{Z}^* \to \mathcal{Y} \\
&\textbf{DEC} : \mathcal{K}_s \times \mathcal{Z} \to \mathcal{P}
\end{aligned}$$

Тоді $\mathcal{X}$ та $\mathcal{Y}$ можна визначити наступним чином:

$$\begin{aligned}
\mathcal{X} &= \{c \mid \textbf{ENC}(pk, m) = c,\ m \in \mathcal{P}\} \\
\mathcal{Y} &= \{z \mid \textbf{EVAL}(evk, C, c_1, c_2, ..., c_n) = z,\ c_i \in \mathcal{Z},\ C \in \mathcal{C}\}
\end{aligned}$$

В деяких схемах, ключі розрахунків та шифрування однакові, але часто це і не так, тому в визначеннях було наведено більш спільний випадок.

В оригінальних документах FHE [4] не було зазначено, що алгоритм розшифровування **DEC** повинен мати можливість працювати з результатом виконання алгоритму шифрування **ENC** - $\mathcal{X}$, і було зазначено, що данні можуть бути розшифровані після виконання розрахунків над ними **EVAL** - $\mathcal{Y}$. Для можливості розшифровування, зразу після зашифровування було запропоновано мати *чисту Булеву схему* або ж по суті функцію $f(x) = x$, для виконання розрахунків і отримання даних які вже можна буде розшифровувати. Більшість сучасних FHE схем, дозволяють проводити операції дешифрування даних, над якими не було проведено розрахунків, тому я не буду заглиблюватись в цю тему.

1.1.1 Атрибути та властивості

Тут представлені характеристики методів гомоморфного шифрування. Ми встановлюємо такі властивості, як компактність і конфіденційність схеми, які забороняють спрощені рішення задачі гомоморфного шифрування, з одного боку, і вимагають таких властивостей, як коректність, для того, щоб навіть називати це схемою шифрування.

Визначення 1.1.2 (Коректне розшифровування [1])**.** $\mathcal{C}$-схама має атрибут коректного розшифрування якщо виконується наступне твердження:

$$\mathbf{DEC}(sk, \mathbf{ENC}(pk, m)) = m,$$
$$\text{де } pk, sk, evk \leftarrow \mathbf{GEN}(1^\lambda, \alpha),\ \alpha \in \mathcal{A},\ m \in \mathcal{P}.$$

Це означає, що ми повинні мати можливість безпомилково розшифровувати зашифрований текст.

Визначення 1.1.3 (Коректні розрахунки [1])**.** $\mathcal{C}$-схема коректно розраховує всі Булеві схеми $C \in \mathcal{C}$, якщо виконується наступне твердження:

$$\mathbf{DEC}(sk, \mathbf{EVAL}(evk, C, c_1, c_2, ..., c_n)) \ = \ C(m_1, m_2, ...m_n),$$
$$pk, sk, evk \leftarrow \mathbf{GEN}(1^\lambda, \alpha),\ \alpha \in \mathcal{A},\ c_i \in \mathcal{X} \text{ та } m_i \leftarrow \mathbf{DEC}(sk, c_i)$$

Це визначення означає, що розрахунки над зашифрованими даними з подальшим розшифровуванням повинні бути однакові з результатом розрахунків над не зашифрованими даними.

Будемо називати $\mathcal{C}$-схему *коректною* якщо для неї будуть виконуватись 1.1.2 та 1.1.3 твердження.

Визначення 1.1.4 (Компактність $\mathcal{C}$-схеми)**.** $\mathcal{C}$-схема вважається компактною якщо існує поліном p, такий що, для будь-якого кортежу $(pk, sk, evk) \leftarrow \mathbf{GEN}(1^\lambda, \alpha)$, $\alpha \in \mathcal{A}$, будь-якої Булевої схеми $C \in \mathcal{C}$ та шифру $c_i \in \mathcal{X}$, розмір результату виконання $\mathbf{EVAL}(evk, C, c_1, c_2, ..., c_n)$ не більше від $p(\lambda)$ бітів, в не залежності від Булевої схеми.

Визначення 1.1.4, показує що під час гомоморфних операцій розмір результату не повинен збільшуватись, і залежить тільки від параметра безпеки λ.

Визначення 1.1.5 (Компактно розрахункова $\mathcal{C}$-схема [8])**.** $\mathcal{C}$-схема компактно розраховує всі Булеві схема $C \in \mathcal{C}$, якщо вона компактна 1.1.4 та *коректна.*

Безпека схеми

Далі, важливо зупинитись на безпеці та конфіденційності схеми. Безпеку схеми можна розділити на дві компоненти: семантична безпека, та обфускація схеми. Якщо обфускація використовується коли алгоритм шифрування секретний, і вразливий, то семантична безпека описує розподіл вихідних даних з **EVAL** та **ENC**.

Визначення 1.1.6 (Приватне гомоморфне шифрування схеми [8](2.16))**.** $\mathcal{C}$-схема вважається безпечною, якщо для будь-якого кортежу $(pk, sk, evk) \leftarrow \mathbf{GEN}(1^\lambda, \alpha)$, $\alpha \in \mathcal{A}$, будь-якої Булевої схеми $C \in \mathcal{C}$ та шифру $c_i \in \mathcal{X}$, такого що $m_i \leftarrow \mathbf{DEC}(sk, c_i)$ існує два розподіли:

$$Dist_1 = \mathbf{EVAL}(evk, C, c_1, c_2, ..., c_n)$$
$$Dist_2 = \mathbf{ENC}(pk, C(c_1, c_2, ..., c_n)$$

які повинні бути статистично або обчислювально нерозрізнені. Ці вимоги показують, що розподіл виконання обчислень Булевої схеми над шифром $Dist_1$ повинен бути однаковий (статистично, обчислювально) з розподілом, отриманим шляхом зашифровування *чистого* тексту, який насамперед являється результатом виконання Булевої операції над незашифрованими даними $Dist_2$.

Часто термін безпечної системи можна зустріти як *Сильно гомоморфна система*[7].

1.1.2 Класифікація

Оскільки не всі схеми FHE мають однакові властивості, цей розділ показує як схеми класифікуються, в залежності від того, які схеми вони можуть обчислювати.

Визначення 1.1.7 (Частково Гомоморфна схема або $\mathcal{C}$-Гомоморфізм [4])**.** $\mathcal{C}$-схема називається, частково гомоморфною (SHE), якщо вона має коректне шифрування 1.1.2, та коректне обчислення 1.1.3.

Для частково гомоморфних $\mathcal{C}$-схем нема вимог до компактності, тому з кожним гомоморфним розрахунком розмір вихідного шифру може збільшуватись. Також нема ніяких вимог до множини Булевих операцій які можуть бути використовувані для розрахунків.

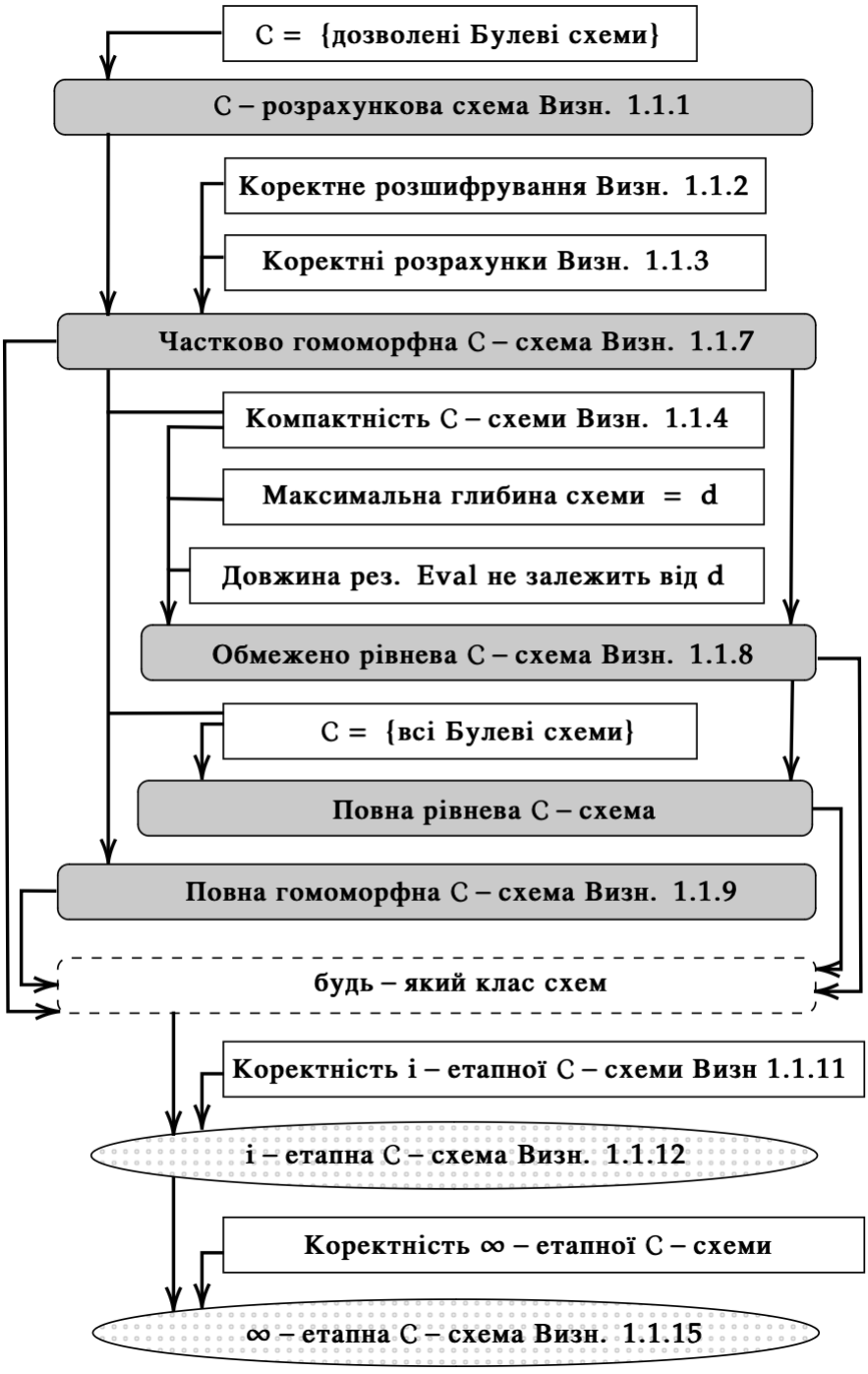

Рис. 1.1: Дерево класифікацій $\mathcal{C}$-схем. Прямокутниками позначені визначення, закруглені затемнені прямокутники, позначають класи $\mathcal{C}$-схеми, а еліпси позначають розширення для етапних розрахунків. Стрілки показуть залежність одного твердження від іншого.

Визначення 1.1.8 (Обмежено-рівнева Гомоморфна схема)**.** $\mathcal{C}$-схема називається Обмежено-рівневою, якщо алгоритм генерації ключів **GEN** приймає додатковий параметр $\alpha = d$, який означає максимальну глибину Булевої схеми, яка може бути обчислена. Також застосовані вимоги до компактності, коректності, і те що розмір вихідних даних розрахунків не повинен залежати від d.

Визначення 1.1.9 (Повна Гомоморфна схема)**.** Повною гомоморфною схемою, називають $\mathcal{C}$-схему, до якої застосовані вимоги, коректності, компактності, та вона може обчислювати Булеву схему з множини усіх схем, або ж будь-яку схему.

1.1.3 Композиція розрахунків

Часто, задача потребує декілька послідовних розрахунків, тобто результат певної Булевої схеми повинен слугувати вхідними даними для наступної схеми, або ж простими словами можна це назвати - композиція. Кожну операцію розрахунків над шифром **EVAL** будемо називати *етапом розрахунків.*

З визначення коректних розрахунків 1.1.3 видно що вхідні дані для алгоритму обчислення **EVAL** повинні належати множині $\mathcal{X}$ - або ж множині *чистого шифру*, який є результатом алгоритму **ENC**. Цей розділ описує вимоги, виконуючи які алгоритм розрахунку схеми **EVAL**, може приймати на вхід як результат виконання інших розрахунків $\mathcal{Z}$, так і *чистий шифр* $\mathcal{X}$:

EVAL$(evk, C, c_1, c_2, ..., c_n)$, де $(pk, sk, evk) \leftarrow$ **GEN**$(1^\lambda, \alpha)$, $\alpha \in \mathcal{A}$, $C \in \mathcal{C}$ та $c_i \in \mathcal{X} \cup \mathcal{Z}$

В літературі *розрахунки з етапами* називають ***гомоморфним шифруванням з i-етапами***(i-hop homomorphic encryption [20], [10])

Визначення 1.1.10 (Розрахунки з етапами)**.** Обчислення $\mathbf{C}_{i,n}$ в i етапів, та шириною n, визначається множиною Булевих схем $\{C_{kl}\}$, де $1 \leq k \leq i, 1 \leq l \leq$ n, та C_{kl} має kn вхідних даних. За вхідними даними $m_{01}, m_{02}, ..., m_{0n}$ ми обчислюємо:

$$m_{kl} = C_{kl}(m_{01}, m_{02}, ..., m_{0n}, ..., m_{k-1,1}, ..., m_{k-1,n}), \text{ де } 1 \leq k \leq i, 1 \leq l \leq n.$$

Результат розрахунків з етапом після **EVAL** та **DEC** буде *чистий текст* $m_{i1}, m_{i2}, ..., m_{in}$. Визначимо початковий *чистий текст* як $\overrightarrow{m_0}$, та вихідний *чистий текст* як $\overrightarrow{m_i}$, тоді можна записати співвідношення $\overrightarrow{m_i} = C_{i,n}(\overrightarrow{m_0})$.

Нехай $(pk, sk, evk) \leftarrow \mathbf{GEN}(1^\lambda, \alpha)$, $\alpha \in \mathcal{A}$, та $c_{i1}, c_{i2}, ..., c_{in} \in \mathcal{X}$, тоді шифр $\{c_{kl}\}$, $1 \leq k \leq i, 1 \leq l \leq n$ обчислюється рекурсивно наступним чином:

$$c_{kl} = \mathbf{EVAL}(evk, C_{kl}, c_{01}, ..., c_{0n}, ..., c_{k-1,1}, ..., c_{k-1,n})$$

Результат розрахунків з етапом над зашифрованими даними, буде шифр $c_{i1}, c_{i2}, ..., c_{in}$. Позначивши початковий (вхідний) шифр як $\overrightarrow{c_0}$ та результівний шифр як $\overrightarrow{c_i}$ можна описати співвідношення яке описує нотацію алгоритму **EVAL** з декількома виходами: $\overrightarrow{c_i} = \mathbf{EVAL}(evk, C_{1,n}, \overrightarrow{c_0})$

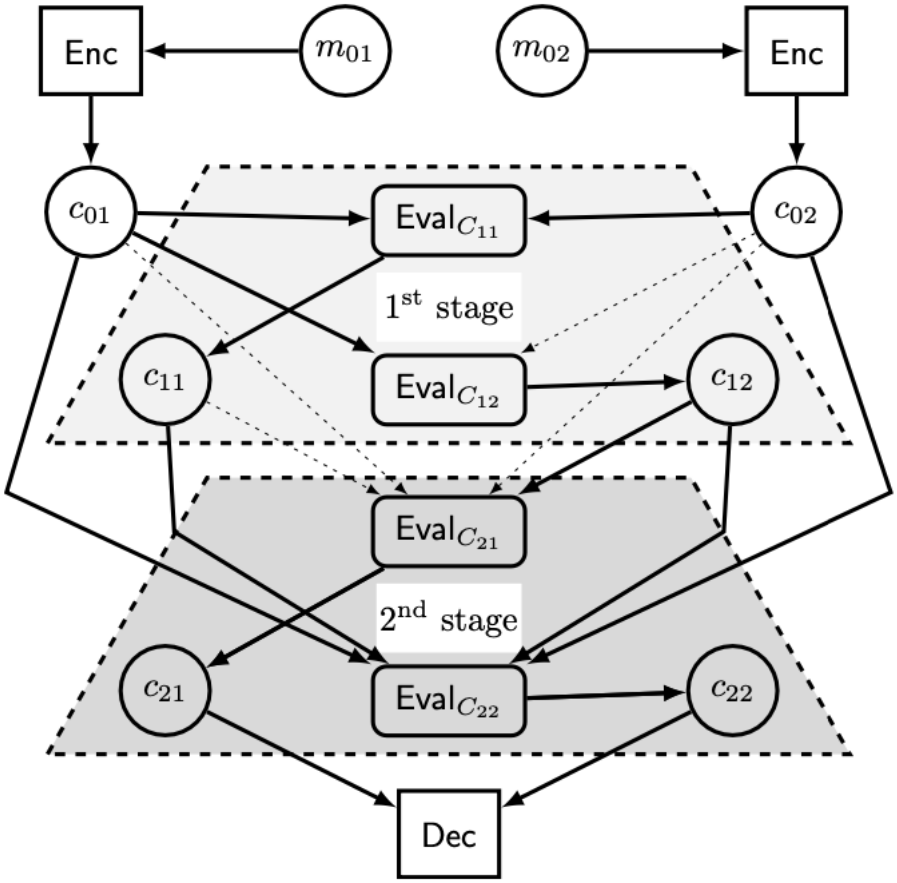

Рис. 1.2: Приклад [1] гомоморфного шифрування з і-етапами, де $i = 2, n = 2$

З вище описаного визначення 1.1.10 можна зробити висновок що вхідними даними для будь-якого етапу, окрім першого, може бути ТІЛЬКИ результат попереднього етапу.

На перший погляд, може здатись, що якщо у нас є можливість обчислити довільну Булеву схему, не використовуючи i-етапне шифрування, то повинна бути можливість обчислювати багато схем послідовно. Проте це не так. Нема

гарантій того, що результат виконання **EVAL**, буде валідний для використання як вхідні дані для наступного **EVAL**. Наведемо приклад [1]: Нехай у нас є схема C яка приймає на вхід $c_1, c_0, ..., c_n$, та результатом якої є $c'_0, c'_1, ..., c'_v$, та схема C', яка приймає на вхід $c_0, c_1, ..., c_v$ та результат якої $c'_0, c'_1, ..., c'_w$. Тоді існує 2 можливих сценарії: 1) Якщо ми візьмемо композицію C та C' як схему для розрахунків $\mathbf{EVAL}(\text{evk}, C \circ C', c_1, c_2, ..., c_n)$ то ці розрахунки будуть коректними, оскільки виконується одна Булева схема, 2) проте, якщо ми спочатку розрахуємо $\mathbf{EVAL}(\text{evk}, C, c_1, c_2, ..., c_n) = c'_1, c'_2, ..., c'_v$, а потім $\mathbf{EVAL}(\text{evk}, C', c'_1, c'_2, ..., c'_v)$ то це не спрацює зі звичайною схемою повного гомоморфного шифрування, оскільки вона не гарантує коректність даних після розрахунків, для наступних операцій. Тому якщо стоїть задача виконання послідовних, незалежних обчислень, то варто використовувати схему з i-етапами.

Визначення 1.1.11 (Коректність гомоморфного шифрування з i-етапами)**.** Нехай $(pk, sk, evk) \leftarrow \mathbf{GEN}(1^\lambda, \alpha)$, $\alpha \in \mathcal{A}$, та $\mathbf{C}_{i,n} = \{C_{k,l}\}$ - довільне поетапне обчислення, де n це розмір полінома від λ та $\overrightarrow{c_0} = (c_{01}, c_{02}, ..., c_{0n}) \in \mathcal{X}^n$. Тоді $\mathcal{C}$-схему можна вважати коректною з i-етапами, якщо виконується наступне твердження:

$$\mathbf{DEC}(sk, \mathbf{EVAL}(evk, \mathbf{C}_{i,n}, \overrightarrow{c_0})) = \mathbf{C}_{i,n}(\mathbf{DEC}(sk, \overrightarrow{c_0}))$$

Хоча це визначення і дуже схоже на визначення коректності розрахунків 1.1.3, проте важливо розуміти, що наведене вище визначення застосоване до розрахунків з багатьма етапами, про що свідчить $\mathbf{C}_{i,n} = \{C_{k,l}\}$.

На Рис. 1.2 зображений приклад розрахунків з етапами, де $i = 2, n = 2$.

Тепер, маючи загальне визначення коректності гомоморфного шифрування, можна описати більш часткові випадки шифрування з i-етапами, а саме: i-етапне, мульті-етапне, полі-етапне та ∞-етапне.

Визначення 1.1.12 (i-етапна $\mathcal{C}$-схема [10])**.** Нехай $i \in \mathbb{N}$, тоді $\mathcal{C}$-схема i-етапна, якщо вона коректна для всіх j-етапних схем, де $1 \leq j \leq i$.

Замість того щоб параметризувати етапи числом, як в визначенні i-етапної схеми: $i \in \mathbb{N}$, етапи можуть залежати від полінома параметризовані λ.

Визначення 1.1.13 (мульті-етапна $\mathcal{C}$-схема [10])**.** Нехай p - деякий поліном, тоді $\mathcal{C}$-схема називається мульті-етапною, якщо вона коректна для всіх j-етапних схем, таких що: $1 \leq j \leq p(\lambda)$.

Визначення 1.1.14 (полі-етапна $\mathcal{C}$-схема [1])**.** Нехай p - деякий поліном, та $\alpha \in \mathcal{A}$, тоді $\mathcal{C}$-схема називається полі-етапною, якщо вона коректна для всіх j-етапних схема, таких що: $1 \leq j \leq p(\lambda, \alpha)$.

Визначення 1.1.15 (∞-етапна $\mathcal{C}$-схема)**.** $\mathcal{C}$-схема називається ∞-етапною, якщо вона коректна для всіх j-етапних схем для всіх j.

1.1.4 Зв'язок FHE та i-етапних схем

todo

1.2 Обмеження

Існує велика кількість програм які використовують FHE для вирішування поставленої задачі. Проте наразі існую обмеження у використання цієї технології, далі в цьому розділі буде розглянуто декілька з них.

- Перше обмеження FHE, це жорстка прив'язка пар ключів, що унеможливлює багатьом користувачам використовувати спільні дані. Уявимо ситуацію де багато користувачів використовують систему яка, в свою чергу покладається на внутрішню базу даних для обчислень. Дані, які були додані в базу даних, можуть бути використані в обчисленнях, тільки користувачем який їх туди додав. Тобто в ситуації коли багато користувачів повинні працювати над одними даними, щоб досягти спільної цілі - FHE обмежений. Проте існує гарний претендент на вирішення цього обмеження [17] Multikey Fully Homomorphic Encryption.

- Друге обмеження це те що FHE потребує дуже великих витрат на обчислення. Розрахунки та проведення операцій над зашифрованими даними виконуються набагато довше ніж на чистих незашифрованих даних. Хоча сучасні алгоритми FHE показують достойні покращення в часі на розрахунки, однак ця проблема все ще залишається одним із головних аргументів не використовувати FHE. Запропоноване часткове розв'язання цієї проблеми, це використовувати Тьюрінг Машини замість Булевих схем [11].

- Третє обмеження полягає в тому що алгоритм розрахунків над зашифрованими, даними не може бути зашифрований сам по собі. Тому, наприклад, маючи складний алгоритм розрахунків акцій, який не повинен бути оприлюднений, складно використати в контексті FHE. Часткове рішення цієї проблеми було запропоноване в роботі Michael Naehrig [18], де він запропонував передавати функцію у зашифрованому вигляді. Проте шифрування алгоритму це не зовсім область відповідальності FHE, і повинна досягатись шляхом обфускації алгоритмів.

1.3 Відомі області застосування

В цьому розділі описані можливі області де може бути ефективно застосовані FHE. Будуть описані як і повноцінні області де може бути застосована технологія, тек і використання як допоміжного інструменту, для побудови більш комплексних систем.

Конфіденційність користувача у рекламних пропозиціях

В сучасному світі реклама може бути не тільки набридливою для користувача, а напроти дуже корисною, якщо алгоритми для підбору цієї реклами базуються на персональних даних, користувача, таких як: його вподобання, рік народження, перегляд певних ресурсів та джерел, локація користувача тощо. Більшість людей відносяться до персональної безпеки дуже відповідально, і не хочуть її розголошувати задля отримання персоналізованої реклами.

Для вирішення цієї проблеми чудово підходить FHE, оскільки він може виконувати алгоритми над зашифрованими даними користувача.

В документі [19] була описана одна з таких систем, де рекомендації для користувача основані на рекомендаціях його друзів. Система застосовує гомоморфне шифрування, щоб була можливість отримувати рекомендації друзів буз розголошення їх особистостей.

Інша реалізація задачі була описана в документі [2]. В реалізації користувач отримує рекомендації від системи, якій не важливо який контент їй був переданий, та від якого користувача. Для побудови такого алгоритму, була зроблена проста, але дуже ефективна FHE схема, яка дозволяє отримувати рекомендації для користувача, який залишається невидимим для системи.

Ще одна робота, яка варта згадки [18], забезпечує рекламні рекомендації на базі локації користувача, виконуючи алгоритми над зашифрованими даними, що не дозволяє зловмисникам отримати, де знаходиться користувач системи.

Конфіденційність даних пацієнта у медичних застосунках

В роботі [18], описане практичне використання FHE в медичних застосунках, де важлива конфіденційність даних пацієнта. В описаній системі пацієнт робить запит до системи зі своїми даними в зашифрованій формі, оскільки користувач системи це власник даних, то тільки він може їх розшифрувати. Застосунок який реалізує сервіс, в свою чергу, може рахувати, чи отримувати з баз даних, інформацію, таку як: група крові, тиск, серцебиття, хвороби та інше. за зашифрованими даними, результат роботи сервісу, також буде зашифровані дані.

В роботі [16] була реалізована подібна система, яка рахує вірогідність серцевого нападу, основуючись на зашифрованих даних пацієнта.

Інтелектуальний аналіз даних

Аналіз даних на великих обсягах інформації дає гарний результат, проте ціна цьому результату приватність даних користувачів.

Була зроблена чудова робота [23] яка реалізує логіку зашифрованого аналізу даних, без втрати точності, і забезпечує безпеку за допомогою FHE схеми.

Конфіденційність фінансових операцій

Хоча як було описано в розділі про обмеження, FHE і не забезпечує шифрування самого алгоритму, його можна використовувати в іншому сценарії:

Представимо, що існує дві компанії X та Y, у компанії X є приватні акції, а у компанії Y є секретний алгоритм який рахує прогноз по динаміці змін акцій. Тоді компанії X достатньо застосувати FHE для своїх даних, що дозволить рахувати над ними алгоритм компанії Y, без дешифрування.

Криміналістичне розпізнавання зображень

В роботі [3] була представлена ще одна чудова сфера застосування FHE. Поліція та інші правоохоронні органи використовують подібні інструменти для пошуку нелегальних фотографій на жорстких дисках, у мережевих потоках даних та інших наборах даних. Поліція використовує базу даних "поганих"хеш-значень зображень. Можливість того, що злочинці можуть отримати доступ до цієї бази даних, перевірити, чи будуть їхні фотографії розпізнані, і, якщо так, змінити їх, викликає серйозне занепокоєння.

Ця схема реалізує сценарій, коли база даних поліції зашифрована, але водночас законний мережевий трафік компанії залишається приватним завдяки використанню дещо гомоморфної стратегії шифрування, запропонованої в наступному документі [5]. Компанія протиставляє хешований і зашифрований потік фотографій зашифрованій базі даних поліції. Тимчасова змінна надається поліції через заздалегідь визначений проміжок часу або поріг, при цьому постачальник послуг нічого не дізнається про саму зашифровану базу даних.

Використання FHE як інструмент для більш комплексних криптосистем

Далі наведені приклади, як FHE може бути використаний як інструмент, для створення більш складних криптографічних інструментів, таких як: цифрові підписи, MACs, доведень з нульовим розголошенням, та інш.

Доведення з нульовим розголошенням

todo

Делегування розрахунків

todo

Цифрові підписи

todo

Багатопарне обчислення

todo

1.4 Існуючі FHE схеми

Розділ 2
Використання FHE в хмарних технологіях

2.1 Бібліотека HeLib

Для реалізації поставленої задачі буде використовуватись бібліотека гомоморфного шифрування HeLib. HeLib була написана на C++ та реалізовує функціонал Brakerski-Gentry-Vaikuntanathan (BGV), та Cheon-Kim-Kim-Song (CKKS) схем.

2.1.1 Алгоритми над схемою

ВИСНОВКИ

Контент

ПЕРЕЛІК ДЖЕРЕЛ

1. Frederik Armknecht, Colin Boyd, Christopher Carr, Kristian Gjøsteen, Angela Jäschke, Christian A. Reuter, and Martin Strand. A guide to fully homomorphic encryption. Cryptology ePrint Archive, Paper 2015/1192, 2015. https://eprint.iacr.org/2015/1192.

2. Frederik Armknecht and Thorsten Strufe. An efficient distributed privacy-preserving recommendation system. pages 65 – 70, 07 2011.

3. Christoph Bosch, Andreas Peter, Pieter Hartel, and Willem Jonker. Sofir: Securely outsourced forensic image recognition. pages 2694–2698, 05 2014.

4. Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) lwe. Cryptology ePrint Archive, Paper 2011/344, 2011. https://eprint.iacr.org/2011/344.

5. Zvika Brakerski and Vinod Vaikuntanathan. Fully homomorphic encryption from ring-lwe and security for key dependent messages. In Phillip Rogaway, editor, *Advances in Cryptology – CRYPTO 2011*, pages 505–524, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

6. Andrei Bulatov. *Boolean Circuits*. duke.edu, https://users.cs.duke.edu/ reif/courses/complectures/Bulatov/32.pdf, Feb 2021.

7. Michael Clear, Arthur Hughes, and Hitesh Tewari. Homomorphic encryption with access policies: Characterization and new constructions. In *Progress in Cryptology – AFRICACRYPT 2013*, pages 61–87. Springer Berlin Heidelberg, 2013.

8. Craig Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009. crypto.stanford.edu/craig.

9. Craig Gentry, Shai Halevi, Chris Peikert, and Nigel P. Smart. Field switching in bgv-style homomorphic encryption. Cryptology ePrint Archive, Paper 2012/240, 2012. https://eprint.iacr.org/2012/240.

10. Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. i-hop homomorphic encryption and rerandomizable yao circuits. Cryptology ePrint Archive, Paper 2010/145, 2010. https://eprint.iacr.org/2010/145.

11. Shafi Goldwasser, Yael Tauman Kalai, Raluca Ada Popa, Vinod Vaikuntanathan, and Nickolai Zeldovich. How to run turing machines on encrypted data. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology – CRYPTO 2013*, pages 536–553, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

12. Shai Halevi and Victor Shoup. Algorithms in helib. Cryptology ePrint Archive, Paper 2014/106, 2014. https://eprint.iacr.org/2014/106.

13. Shai Halevi and Victor Shoup. Bootstrapping for helib. Cryptology ePrint Archive, Paper 2014/873, 2014. https://eprint.iacr.org/2014/873.

14. Shai Halevi and Victor Shoup. Design and implementation of helib: a homomorphic encryption library. Cryptology ePrint Archive, Paper 2020/1481, 2020. https://eprint.iacr.org/2020/1481.

15. Marc Joye. Sok: Fully homomorphic encryption over the [discretized] torus. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2022(4):661–692, Aug. 2022.

16. Kristin Lauter, Michael Naehrig, and Vinod Vaikuntanathan. Can homomorphic encryption be practical? Cryptology ePrint Archive, Paper 2011/405, 2011. https://eprint.iacr.org/2011/405.

17. Adriana Lopez-Alt, Eran Tromer, and Vinod Vaikuntanathan. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. Cryptology ePrint Archive, Paper 2013/094, 2013. https://eprint.iacr.org/2013/094.

18. Michael Naehrig, Kristin Lauter, and Vinod Vaikuntanathan. Can homomorphic encryption be practical? In *Proceedings of the 3rd ACM*

Workshop on Cloud Computing Security Workshop, CCSW '11, page 113–124, New York, NY, USA, 2011. Association for Computing Machinery.

19. Arjan Jeckmans Andreas Peter and Pieter Hartel. Efficient privacy-enhanced familiarity-based recommender system. Embedded Security Group, University of Twente, 2005. https://ofmas.ir/dlpaper/y1091.pdf.

20. Ron Rothblum. Homomorphic encryption: From private-key to public-key. In Yuval Ishai, editor, *Theory of Cryptography*, pages 219–234, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

21. Harsha Tirumala Rutgers University, Swastik Kopparty. *Boolean Circuits and Formulas*. rutgers.edu, https://sites.math.rutgers.edu/ sk1233/courses/topics-S20/lec1.1.pdf, 2020.

22. Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers. Cryptology ePrint Archive, Paper 2009/616, 2009. https://eprint.iacr.org/2009/616.

23. Zhiqiang Yang Sheng Zhong Rebecca N. Wright. Privacy-preserving classification of customer data without loss of accuracy. 1Computer Science Department, Stevens Institute of Technology, 2005. https://www.cs.columbia.edu/ rwright/Publications/sdm05.pdf.

24. Amit Sahai Yuval Ishai and David Wagner. Private circuits: Securing hardware against probing attacks, 2006. https://people.eecs.berkeley.edu/ daw/papers/privcirc-crypto03.pdf.

ДОДАТКИ

контент