Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра математики факультету інформатики

Курсова робота на тему:
Схеми підписів, що базуються на решітках

Керівник курсової роботи
проф. Олійник Б. В
(прізвище та ініціали)

_____
(підпис)
“18” квітня 2020 р.

Виконала студентка
напряму підготовки 113
Прикладна математика
Тартасюк А. Г.
“18” квітня_ 2020 р.

Київ 2020

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра математики факультету інформатики

ЗАТВЕРДЖУЮ
Зав. кафедри математики,
проф. Олійник Б. В.

_____
(підпис)
“____” _____ 2020 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
на курсову роботу

студентці Тартасюк А.Г. факультету інформатики 4 курсу
ТЕМА «Схеми підписів, що базуються на решітках»

Вихідні дані:

- Vadim Lyubashevsky, Lattice Signatures Without Trapdoors

- Leixiao Cheng, Boru Gong, and Yunlei Zhao, Lattice-Based Signature from Key Consensus

Зміст ТЧ до курсової роботи:

Вступ

1. Огляд гомоморфних криптосистем
2. Решітки та базис решіток
3. Опис процедури створення підпису
4. Практична частина

Висновки

Список літератури

Дата видачі “___” _____ 2020 р. Керівник _____

Завдання отримала _____

Тема: «Схеми підписів, що базуються на решітках»
Календарний план виконання роботи:

№ п/п	Назва етапу курсового проекту (роботи)	Термін виконання етапу	Примітка
1.	Отримання завдання на курсову роботу.	05.11.2019	
2.	Огляд літератури за темою роботи.	16.11.2019	
3.	Вивчення діаграм Вороного	10.01.2020	
4.	Розгляд гомоморфних криптосистем	15.01.2020	
5.	Вивчення матеріалу з теми решіток та їх базису	02.02.2020	
6.	Опрацювання алгоритму створення підпису на основі решіток	21.02.2020	
7.	Дослідження стійкості до інсайдерських атак	18.03.2020	
8.	Написання пояснювальної роботи.	14.04.2020	
9.	Створення слайдів для доповіді та написання доповіді.	15.04.2020	

Студент Тартасюк А. Г.

Керівник Олійник Б. В.

“_____”_____2020 р.

Зміст

Анотація

Курсова робота присвячена дослідженню схеми цифрових підписів як таких, гомоморфніх криптосистем та решіток.
У вступі розповідається про актуальність обраної теми.

У першому розділі розглядаються основні відомості про гомоморфні криптосистеми та їх застосування. У другому розділі формулюється означення решіток та базису решіток. У третьому розділі розглянуті різні типи цифрових підписів та більш детально розглянутий варіант, запропонований Вадимом Любашевським. У четвертому розділі розглянуто на практиці стійкість цифрового підпису до інсайдерських атак. У списку використаної літератури наводяться джерела, які були використані під час дослідження.

Ключові слова: цифровий підпис, гомоморфна криптосистема, решітка, базис решітки.

Вступ

Одним із важливих засобів отримання послуг аутентифікації є цифровий підпис. Дослідження постквантових електронних підписів нині набувають актуальності через можливість виникнення квантового комп'ютера. Криптосистеми на решітках мають ряд переваг, серед яких основною є стійкість від квантового криптоаналізу. Тому питання безпеки підписів на решітках потребує детального вивчення.

Поява нових обчислювальних платформ, таких як хмарні обчислення, IoT, вимагає прийняття все більшої кількості стандартів безпеки, що, в свою чергу, вимагає впровадження різноманітного набору криптографічних примітивів, але це лише частина історії. На рівні обчислювальної платформи ми бачимо різноманітність можливостей обчислювальної техніки, починаючи від високоефективних (у режимі реального часу) віртуалізованих середовищ, таких як хмарні обчислювальні ресурси та програмно визначені мережі, до сильно обмежених ресурсів платформ IoT, щоб реалізувати бачення Інтернету речей .
Це створює труднощі при розробці та здійсненні нових стандартів криптографії в одному варіанті здійснення, оскільки обчислювальні платформи точно визначають цілі та обмеження.

Робота складається з чотирьох розділів. У перших двох наведена ознайомлююча інформація, що стосується гомоморфних криптосистем, решіток та їх базису. У третьому розділі розглядаються різні типи цифрових підписів, та наведений більш детальний опис варіанту, запропонованим Вадимом Любашевським. У четвертому розділі розглядається стійксть схеми цифрового підпису, що базується на решітках, до інсайдерських атак.

Метою даної роботи є ознайомитися з темою цифрових підписів, що базуються на решітках, та її актуальністю, розглянути відомості щодо гомоморфних криптосистем та проаналізувати наскільки дані цифрові пидписи є стійкими до атак.

1. Огляд гомоморфних криптосистем

Гомоморфна криптосистема [1] відіграє важливу роль у збереженні наших даних під час обробки в публічному середовищі. Це форма шифрування, яка дозволяє виконувати конкретні типи обчислень на тексті шифру і генерувати зашифрований результат, який при розшифровці відповідає результатам операцій, виконаних на простому тексті. Більш точно, гомоморфне шифрування - це перетворення даних у шифротекст, який можна проаналізувати та працювати з ним, як ніби це все ще було у первісному вигляді. Гомоморфна криптосистема дозволяє виконувати складні математичні операції над зашифрованими даними без шкоди для шифрування. Це бажана особливість у сучасних архітектурах систем зв'язку.

Гомоморфні криптосистеми в основному поділяються на **дві категорії**, а саме – частково гомоморфні криптосистеми та повністю гомоморфні криптосистеми.

По-перше, **частково гомоморфна криптосистема**. Криптосистема вважається частково гомоморфною [2], якщо вона проявляє або адитивний, або мультиплікативний гомоморфізм, але не одночасно. Очевидно, що часткові схеми гомоморфного шифрування є корисними у певних програмах. Крім того, ефективність деяких часткових схем гомоморфного шифрування досить висока для практичного застосування.

По-друге, **повністю гомоморфна криптосистема**. Це криптосистема, яка підтримує довільне обчислення на шифротекстах, і вона є набагато потужнішою [3]. Така схема дозволяє будувати програми для будь-якої

бажаної функціональності, яку можна запустити на зашифрованих входах для отримання результату. Наявність ефективної та повністю гомоморфної криптосистеми мало б велике практичне значення для аутсорсингу приватних обчислень, наприклад, у контексті хмарних обчислень.

У 1978 році Рональд Рівест, Леонард Адлеман та Майкл Дертузос вперше запропонували концепцію гомоморфного шифрування. З тих пір за останні 40 років відбулося не так багато прогресу. Система шифрування Шафі Голдвассера та Сільвіо Мікалі була запропонована в 1982 році. Це була відома схема шифрування, яка досягла надзвичайного успіху через свій рівень безпеки; У цій же концепції в 1999 р. Паскаль Пейєльєр запропонував систему, яка захищає дані, що була також гомоморфним шифруванням над операцією додавання. Через кілька років, у 2005 році, Ден Боне, Еу-Джин Го і Кобі Ніссім винайшли систему шифрування, за допомогою якої ми можемо виконувати необмежену кількість додавань, але лише одне множення. У **2009 році Крейг Гентрі з IBM запропонував першу повністю гомоморфну систему шифрування**, яка оцінює довільну кількість додавань та множень і таким чином обчислює будь-який тип функції на зашифрованих даних.

Тепер, розглянемо де саме можуть застосовуватися гомоморфні криптосистеми:

1. **Виборчі схеми:** [4] У виборчих схемах гомоморфна властивість забезпечує інструмент для отримання підрахунку зашифрованих голосів без дешифрування окремих голосів.
2. **Протоколи лотереї:** [5] Зазвичай у криптографічній лотереї номер, який вказує на виграшний квиток, повинен бути спільно вибраним випадковим чином усіма учасниками. Використовуючи схему гомоморфного шифрування, це може бути реалізовано наступним

чином: кожен гравець вибирає випадкове число, яке вона шифрує. Тоді за допомогою гомоморфного властивості може бути ефективно обчислено шифрування суми випадкових значень. Поєднання цієї та порогової схеми дешифрування призводить до бажаної функціональності

3. **Приватне використання ненадійних веб-серверів:** [6] Користувачі є частими відвідувачами онлайн-сервісів, розміщених на віддалених веб-серверах. Як результат, зростає рівень занепокоєння щодо безпеки та конфіденційності даних, завантажених користувачами в такі віддалені сервіси, які перебувають під контролем потенційно недовірених сторін. Основна діяльність у цій сфері була зосереджена на запобіганні, виявленні та виправленні порушень безпеки веб-служб, з обмеженими зусиллями, витраченими на питання конфіденційності. Гомоморфне шифрування зберігає модульне множення. Побудова схем шифрування для різних типів та їх роботи має вирішальне значення для досягнення прозорої конфіденційності даних.

Отже, в епоху, коли зосередженість на приватному житті зростає, головним чином, через такі норми, як GDPR, концепція гомоморфного шифрування є великою перспективою для реальних програм у різних галузях. Можливості, що виникають внаслідок гомоморфного шифрування, майже нескінченні. І, мабуть, одним із найбільш хвилюючих аспектів є те, як він поєднує необхідність захисту конфіденційності з необхідністю надання більш детального аналізу.

2. Решітки, базис решіток

Останнім часом решітки стали темою активних досліджень у галузі інформатики. Алгоритмічні проблеми, засновані на ґратах (наприклад, найкоротші та найближчі векторні проблеми, ...) знайшли широке застосування; вони використовувались в алгоритмах оптимізації, при розробці безпровідних протоколів зв'язку та, можливо, найактивнішої галузі дослідження, при розробці захищених криптографічних примітивів (криптографії) та у встановленні незахищеності певних криптографічних схем (криптоаналіз).

Означення: Нехай $v_1 \ldots . v_n \in \mathbb{Z}^m, m \geq n$ - лінійно незалежні вектори. Решітка L, що охоплює $\{ v_1 \ldots . v_n\}$ - це сукупність усіх цілих лінійних комбінацій $v_1 \ldots . v_n$, так що: $L = \{v \in \mathbb{Z}^m \mid v = \sum_{i=1}^{n} a_i v_i \ , a_i \in \mathbb{Z} \}$

На рисунку 1 покажемо як виглядає решітка у двовимірному просторі:

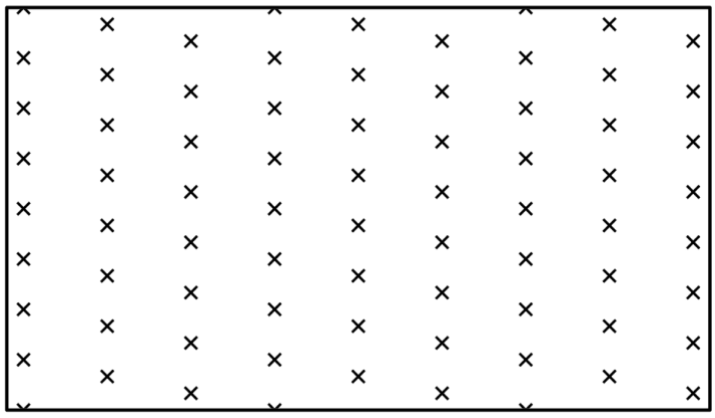

Рис. 1

Зауваження: лінійна комбінація означає, що всі a_i є цілими числами. Цілочисельна решітка означає, що всі v_{ij} (компоненти векторів v_i) є цілими числами, і тому всі точки мають цілі координати.

Означення: Сукупність векторів $B = \{v_1 \ldots . v_n\}$ називається базисом решітки L.

Простішими словами, базисом є невелика кількість векторів, які можна використовувати для відтворення будь-якої точки сітки, яка утворює решітку.

Визначаємо розмірність L як dim (L): = n.

Важливо зауважити, що будь-яка решітка не має лише однієї бази. Насправді їх багато.

Покажемо на рисунку 2 як виглядають решітки з різними базисами:

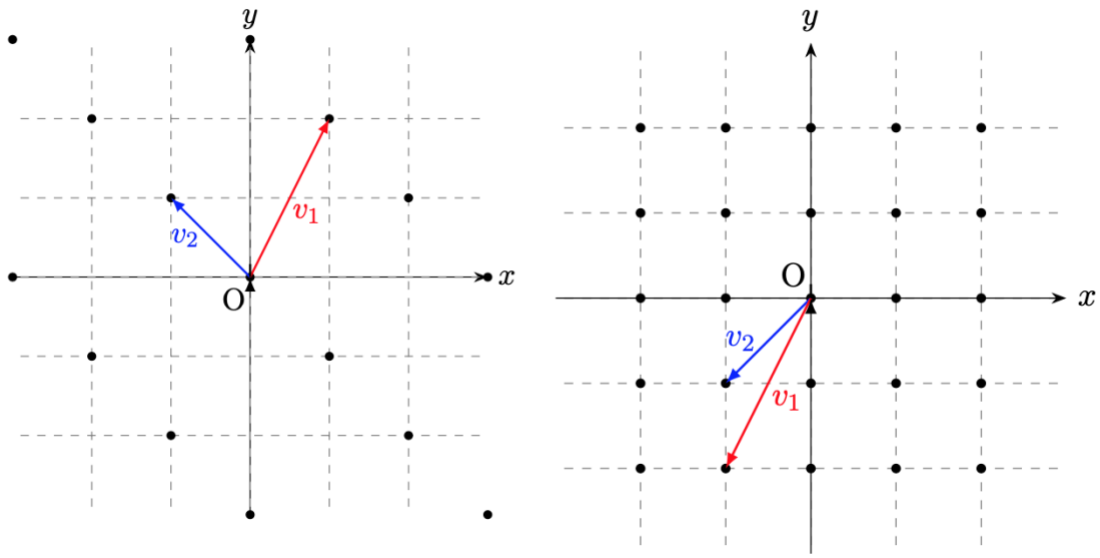

Рис. 2

Усі криптографічні системи, що ґрунтуються на теорії решіток, можна умовно поділити на **два типи**:

1) криптосистеми, які мають строго доведену криптостійкість, але вони неефективні за критерієм складності виконання, тобто швидкодії

прямого та зворотного криптографічних перетворень. До таких криптосистем шифрування відносяться криптографічні системи на SVP та uSVP - (Unique Shortest Vector Problem), SIVP-задачі (Shortest Independent Vector Problem);

2) ефективні за критерієм складності виконання, тобто швидкі відносно прямого та зворотного криптографічних перетворень, але відносно них не має строгого доведення криптографічної стійкості. До таких систем відносять криптографічні системи, в яких використовуються частинні параметри решіток або ті, що ґрунтуються на решітках з циклічністю утворюючого їх базису. До таких відноситься NTRU (Draft standard IEEE 1363.1) шифрування. [7]

Якщо задаватись питанням, **що ж такого цікавого у криптографії на основі решіток**, то на це є декілька причин.

По-перше, справедливо сказати, що це найбільш добре зрозуміла та широко вивчена сім'я важких математичних проблем. Решітки були вивчені математиками ще до початку 1800-х років, коли цим займався Йоганн Гаус, один з найбільших математиків усіх часів. Звичайно, Гуас та інші не мали поняття квантового комп'ютера, і тому навряд чи намагалися розробити квантовий алгоритм для вирішення проблем, які нас цікавлять сьогодні. Однак ці старі математичні роботи все ще служать джерелом глибокого розуміння та потужного розуміння того, що ми можемо, а що не можемо зробити, коли справа стосується роботи з решітками.

По-друге, криптографічні схеми на основі решіток складають левову частку наукових публікацій у галузі так званої «постквантної» криптографії. Якщо ви подивитесь на учасників міжнародного конкурсу, який проводить Національний інститут технологій США, який орієнтований на

стандартизацію нової постквантової захищеної криптографії, ви помітите, що найбільше сімейство матеріалів складається з тем, що стосуються решіток.

Також, щоб зрозуміти їх властивість, ми повинні думати про різні види припущень, найчастіше це і захоплює криптографів в роботі з решітками.

Отже, розв’язок задач теорії решіток надає широкі можливості реалізації криптографічних задач, а також охоплює багато питань, починаючи від задач лінійного програмування до узагальнених теоретико-числових проблем.

3. Схеми цифрових підписів, що базуються на решітках

Цифрові підписи, виходячи з складності проблем, як правило, поділяються на три категорії:

- **Підписи GGH / NTRUSign**

Криптосистеми GGH та NTRUEncrypt одними з перших вирішували клас задач решітки, зокрема на основі вирішення наближеної векторної задачі. Криптосистема GGH включала DSS (Digital Signature Schemes), в свою чергу лягаючи в основу NTRUSign, який поєднував майже всю конструкцію GGH, але використовує ґрати NTRU, використані в NTRUEncrypt. В ході досліджень були виявлені проблеми з безпекою даного типу цифрових підписів.

- **Hash-and-sign підписи**

Концепція відповідає критерію, згідно з яким повідомлення, μ, має бути хешоване перед підписанням. Тобто, щоб підписати повідомлення, спочатку хеш μ до деякої точки h = H (μ), яка повинна знаходитись в діапазоні функції "F", а тоді відома RSA - така функція. Після того, як повідомлення хеширується, воно підписується σ = f-1 (h) і алгоритм верифікації перевіряє, чи f (σ) = H (μ), щоб підтвердити, чи (σ, μ) є дійсною парою повідомлення / пара підписів. Центральне місце в схемі - побудова функцій трап-дору з необхідною властивістю, що кожне вихідне значення має кілька зображень, алгоритм вибірки Гаусса, а також використання модульних решіток.

- **Підписи Фіат-Шамір**

Альтернативний спосіб побудови DSS - спочатку побудувати ідентифікаційну схему певної форми, а потім перетворити її в DSS за допомогою перетворення Фіат-Шамір. Схеми підписів на основі решіток, які використовують перетворення Фіат-Шамір, в основному пояснюються

дослідженнями Любашевського та ін. Процедури в першій публікації Любашевського ґрунтуються на задачі короткого цілого рішення (SIS), тобто якщо рішення знайдено для DSS, то рішення знайдеться і для SIS.

В своїй роботі я хочу більш детально розглянути **схему цифрового підпису, запропонованого Любашевським** [8].

Любашевський пропонує, що підписант має як (секретний) ключ підпису $\hat{s}$, так і (загальнодоступний) ключ підтвердження (h, S), такий, що $h(\hat{s}) = S$. Домени, до яких належать ці ключі, стануть явними пізніше.
Роблячи підпис деякого повідомлення μ, підписант підтверджує свою можливість:

- вибрати випадковий вектор многочленів $\hat{y}$ (який він не розкриває)
- вивести вектор многочленів, різниця яких з $\hat{y}$ - $\hat{s}$ (його секретний ключ)

Для цього підписант вибере деякий випадковий $\hat{y}$, обчислить $e = H(h(\hat{y}), \mu)$ і виведе $(\hat{z}, e)$ з $\hat{z} = \hat{s}e + \hat{y}$. Верифікатор тестує, що $e = H(h(\hat{z}) - Se, \mu)$. Це справедливо для правильного підпису завдяки лінійності h, так як $h(\hat{z}) - Se = h(\hat{s}e + \hat{y}) - Se = h(\hat{y}) + Se - Se = h(\hat{y})$.
Для отримання кільця з цієї схеми ми робимо дві основні модифікації. Перша - переконатися, що кожен користувач має у своєму відкритому ключі функцію h, яка задовольняє $h(\hat{s}) = S$ де $\hat{s}$ - секретний ключ і
S - фіксований стандартний многочлен (не нульовий). Розглянемо кільце $R = \{h_i\}_{i \in [\ell]}$. Друга модифікація залишає справжнього підписувача анонімним, коли він підписує повідомлення. Ми робимо це шляхом простого додавання $\ell - 1$ випадкових змінних, які відповідають членам кільця, за винятком реального підписанта. Наприклад, припустимо, що справжній підписант індексується $j \in [\ell]$, підписант надсилає підпис

$(\hat{z}_i; i \in l, e)$, де $e = H\left(\sum_{i\in[l]} h_i(\hat{y}_i),\ \mu\right), \hat{z}_j = \hat{s}_j e + \hat{y}_j$ та $\hat{z}_i = \hat{y}_i$ для $i \in [l] \setminus \{j\}$. Отже, остаточний підпис буде містити $[l]$ елементів по одному для кожного члена в кільці. Тепер ми переходимо до першої модифікації і покажемо її корисність у правильності схеми. На етапі перевірки верифікатор перевіряє, чи значення хеша $\left(\sum_{i\in[l]} h_i(\hat{z}_i) - Se,\ \mu\right)$, дорівнює e.

Варто зауважити, що це буде справедливо лише тоді, коли $\sum_{i\in[l]} h_i(\hat{z}_i) - Se$ дорівнює $\sum_{i\in[l]} h_i(\hat{y}_i)$,. Фактично, використовуючи лінійність h_j, ми отримуємо $h_j(\hat{z}_j) = h_j(\hat{s}_j e + \hat{y}_j) = h_i(\hat{y}_i) + Se$. Відколи усі кільця мають пари ключів $(h_i, \hat{s}_i)$, такі що $h_i(\hat{s}_i) = S$, верифікатор завжди прийме, коли один з них видає кільцевий підпис.

Для того, щоб протистояти обраним атакам субрингу проти нездатності, ми повинні змінити схему, щоб включити до випадкового виклику Oracle опис кільця, для якого підпис дійсний (якщо ні, підпис легко використовувати повторно для створення підробленого підпису на кільце більшого розміру). Щоб протистояти атакам на змагально обрані параметри, алгоритм кільцевого підпису починається з початкового кроку, під час якого вхідні дані повинні пройти прості тести (обмеження на кількість координат, скалярні розміри тощо). Докази безпеки, запропоновані Любашевським, також повинні бути змінені, щоб врахувати наявність декількох хеш-функцій та елементів підпису. Більше того, використання модифікованих хеш-функцій також вимагає ввести кілька нових лем і пропозицій для завершення доказів. [9]

Також, наведемо більш формалізований опис:

gen−params(1^k):
Given an integer k define the common public parameters.

1. Set n as a power of two larger than k
2. Set $m_u = 3\log n$, and p as a prime larg such that $p = 3 \mod 8$

— Note: these parameters define the sets $\mathcal{D}_y, D_{s,c}$ and the family $\mathcal{H}$.

3. Set $S \leftarrow \mathcal{D}$, $S \neq 0$
4. Output $\mathcal{P} = (k, n, m_u, p, S)$

Рис. 3

gen−keys$(\mathcal{P})$:
Generate a keypair.

1. Set $\hat{s} = (s_1, s_2, \ldots, s_{m_u}) \leftarrow D_{s,c}^{m_u}$
2. If none of the s_i is invertible, go to 1.
3. Let $i_0 \in \{1, \ldots, m\}$ such that s_{i_0} is invertible.
4. $(a_1, a_2, \ldots, a_{i_0-1}, a_{i_0+1}, \ldots, a_{m_u}) \leftarrow \mathcal{D}^{m_u-1}$.
5. Let $a_{i_0} = s_{i_0}^{-1}(S - \sum_{i \neq i_0} a_i s_i)$ and note $\hat{a} = (a_1, \ldots, a_{m_u})$
6. Output $(pk, sk) = (h, \hat{s})$, h being the hash function in $\mathcal{H}$ defined by $\hat{a}$

Рис. 4

sign$(\mathcal{P}, sk, \mu, R)$:
Given a message $\mu \in \mathcal{M}$, a ring of ℓ members with public keys $R = \{h_i\}_{i \in [\ell]} \subset \mathcal{H}(\mathcal{D}, D_h, m_u)$, and a private key $sk = \hat{s}_j$ associated to one of the public keys h_j in R, generate a ring signature for the message.

0. Verify that: the public parameters respect the constraints of steps 1−3 in Ring−gen−params ; sk is in $D_{s,c}^{m_u}$; R is of size bounded by k^c ; one of the public keys in R is associated to sk. If the verification fails output `failed`.
1. For all $i \in [\ell]$; $i \neq j$;$\hat{y}_i \leftarrow D_z^{m_u}$
2. For $i = j$; $\hat{y}_j \leftarrow D_y^{m_u}$
3. Set $\boldsymbol{e} \leftarrow H(\sum_{i \in [\ell]} h_i(\hat{y}_i), R, \mu)$ (e is therefore in $D_{s,c}$)
4. For $i = j$, $\hat{z}_j \leftarrow \hat{s}_j \cdot e + \hat{y}_j$
5. If $\hat{\boldsymbol{z}}_j \notin D_z^{m_u}$ then go to Step 2
6. For $i \neq j$, $\hat{z}_i = \hat{y}_i$
7. Output $\sigma = (\hat{\boldsymbol{z}}_i; i \in [\ell], e)$

Рис. 5

verify$(\mathcal{P}, \mu, R, \sigma)$:
Given a message μ, a ring $R = \{h_i\}_{i \in [\ell]}$ and a ring signature $\sigma = (\hat{\boldsymbol{z}}_i; i \in [\ell], e)$, the verifier accepts the signature only if both of the following conditions satisfied:

1. $\hat{z}_i \in D_z^{m_u}$ for all $i \in [\ell]$
2. $e = H(\sum_{i \in \{1,\ldots,\ell\}} h_i(\hat{z}_i) - S \cdot e, R, \mu)$

Otherwise, the verifier rejects.

Рис. 6

4. Стійкість до інсайдерських атак

Для того, щоб вважатись безпечною, схема підпис, що базується на решітках, повинна задовольняти деяким властивостям анонімності та нездатності до підробок.

У своїй роботі я хочу дослідити **стійкість проти інсайдерських атак**. Хакери, особливо "хакери терористи" або "кібер хакери", отримують багато уваги преси. Вони і справді становлять серйозну проблему. Однак загроза, яку вони становлять є майже непомітною в порівнянні загрози, що ставлять найближчі до нас: інсайдери.

Кіберзагроза, яку створюють інсайдери, не нова. У книзі 1978 року “Злочин за комп’ютером” Донн Паркер підрахував, що 95% комп'ютерних атак було скоєно авторизованими користувачами системи.Хоча це і було в до-інтернетну епоху, коли дуже мало інсайдерів взагалі мали доступ до системи; але все ж основне питання, що співробітники не завжди надійні – залишається і сьогодні. Безумовно, це завжди було правдою - злодійські або корумповані працівники, безсумнівно, існували і самі по собі, але сила комп’ютерів (і наша нездатність забезпечити їх безпеку найкращим чином) робить проблему набагато серйознішою. [10]

Припустимо, що у нас є алгоритм А, здатний зруйнувати стійкість нашої схеми з атаками з боку інсайдерів. Я хотіла б довести, що існує алгоритм В, який за допомогою А може порушити стійкість основної схеми підпису. Основне питання полягає в тому, що коли В отримує челендж від схеми підпису, він може розділити поліноми, але він не знає ключів підписання, пов'язаних з цими розділеними кортежами, і, таким чином, він не може відповісти на відповідні корупційні запити.

Для вирішення цього питання змінимо процес генерації ключів. Кожен користувач генерує k ключів підтвердження, k є параметром захисту. Серед цих ключів k / 2 генерується за допомогою оригінального алгоритму генерації ключів gen −keys, і користувач зберігає пов'язані з ними ключі підписання. Інші k / 2 верифікаційні ключі - це просто рівномірно вибрані хеш-функції. Ключі підтвердження k пронумеровані, порядок вибирається навмання (змішування обох типів ключів).

Наведемо формалізований вигляд:

$\mathsf{Ring{-}gen{-}keys{-}ic}(\mathcal{P})$**:**

1. Choose randomly a subset $T \subset [k]$ of size $k/2$.
2. For $i \in T$, set $pk_i \leftarrow \mathcal{H}(\mathcal{D}, D_h, m)$ and $sk_i = 0$.
3. For $i \in [k] \backslash T$, set $(pk_i, sk_i) \leftarrow \mathsf{Ring{-}gen{-}keys}(\mathcal{P})$.
4. Output $(pk, sk) = (\{pk_i\}_{i\in[k]}, \{sk_i\}_{i\in[k]})$.

Рис. 7

Для набору користувачів $S \subset \mathbb{Z}$(ми пов’язуємо користувачів з цілими числами) ми визначаємо $fulldesc\ (S)$ як опис повного набору ключів перевірки користувачів S.

Підписуючи повідомлення μ для набору користувачів S, користувач викликає перший випадковий оракул з введенням (μ, $fulldesc\ (S)$). Вихід цього випадкового оракула - $\{T_{\sigma,i}\}_{i\in S}$, набір підмножин [k], кожен з яких розміром k / 2. Для підписання користувач створить кільце верифікаційних ключів T, яке включає для кожного користувача iпідмножину ключів підтвердження, індексованих $T_{\sigma,i}$.

Наведемо формалізований вигляд:

Ring−sign−ic$(\mathcal{P}, sk, \mu, R)$:

0. Verify that: the public parameters respect the constraints of steps $1-3$ in Ring−gen−params ; each $sk_i \in sk$ is in $D_{s,c}^{m_u}$; R is of size bounded by k^c ; one of the public keys in R is associated to sk. If the verification fails output `failed`.
1. Set $\{T_{\sigma,i}\}_{i \in R} \leftarrow H_\sigma(\mu, \text{keys}(R))$
2. Define $\text{keys}(T) = (pk_{i,j})_{i \in R, j \in T_{\sigma,i}}$
3. Let i_0 denote the index of the signer in R. Choose randomly $sk_i \in sk$ with $i \in T_{\sigma,i_0}$ such that $sk_i \neq 0$. If none exists, abort.
4. Output Ring−sign$(\mathcal{P}, sk_i, \mu, T)$.

Рис. 8

Оскільки випадковий оракул вибирає k / 2 ключі перевірки підписувача навмання, ймовірність того, що всі вони є рівномірно вибраними випадковими хеш-функціями, експоненціально мала, і тому ймовірність переривання незначна.

Висновки

Курсову роботу було присвячено дослідженню схем цифрового підпису, що базується на решітках.

В перших двох розділах роботи були розглянуті основні теоретичні відомості з теми та буди надані відповіді на запитання що є гомоморфна криптосистема загалом та виклики, які перед нею постають; що є решітки та їх базис; криптографічні системи, що базуються на них, тощо.

У третьому розділі розглянуті різні види цифрових підписів, що базуються на решітках, та окремо розглянутий метод, запропонований Вадимом Любашевським.

У четвертому розділі розглянута стійкість такого цифрового підпису до можливої інсайдерської атаки.

Список літератури

1. R. L. Rivest, L. Adleman, and M. L. Dertouzos “On data banks and privacy homomorphisms- In Foundations of Secure Computation” 1978
2. Cryptanalysis of a Provable Secure Additive and Multiplicative Privacy Homomorphism. In: Proceedings of International Workshop on Coding and Cryptography, Versailles, France
3. Stehle, Damien; Steinfeld, Ron. "Faster Fully Homomorphic Encryption". Asiacrypt 2010
4. M. Hirt and K.Sako, Efficient receipt-free voting based on homomorphic encryption, Springer-Verlag, 2000
5. P-Alain Fouque, G Poupard, J Stern "Sharing Decryption in the Context of Voting or Lotteries" in Financial Cryptography 2000
6. M. Christodorescu. Private Use of Untrusted Web Servers via Opportunistic Encryption. In Web 2.0 Security and Privacy, 2008
7. Usatyuk, V. S. and Kuzmin, O. V. Systems of ciphering based on lattice theory problems, 2010
8. Vadim Lyubashevsky. Towards practical lattice-based cryptography. PhD thesis, University of California, San Diego, 2008
9. Carlos Aguilar-Melchor, Slim Bettaieb, Xavier Boyen, Laurent Fousse, and Philippe Gaborit Adapting Lyubashevsky’s Signature Schemes to the Ring Signature Setting
10. Steven M. Bellovin, The Insider Attack Problem Nature and Scope