УДК 004.056.55 **DOI:10.30837/rt.2020.1.200.09**

І.Д. ГОРБЕНКО, д-р техн. наук, О.Г. КАЧКО, канд. техн. наук,
Ю.І. ГОРБЕНКО, канд. техн. наук, М.В. ЄСІНА, канд. техн. наук, С.О. КАНДІЙ,
Є.В. ОСТРЯНСЬКА, А.С. Д'ЯЧЕНКО

МОЖЛИВОСТІ ЗАСТОСУВАННЯ МЕХАНІЗМІВ ПОВНІСТЮ ГОМОМОРФНОГО ШИФРУВАННЯ В СИСТЕМАХ ЕЛЕКТРОННОГО ГОЛОСУВАННЯ

Вступ

Наразі багато аспектів повсякденного життя все більше пов'язані з інформаційно-комунікаційними системами та сервісами, побудованими на їх основі. Важливою є ідея проводити такі важливі соціальні заходи, як голосування, в електронному форматі. Вона з'явилася досить давно, проте досі не існує надійних систем електронного голосування, які б задовольняли усім вимогам. Основними причинами цього є обмеженість існуючих інструментів та можливостей їх застосування [23].

Однією із проблемних вимог, що висунута до систем електронного голосування, є забезпечення анонімності виборців. З однієї сторони, кожен виборець повинен бути ідентифікований, а з іншої – зміст його голосу має бути невідомим. Запропоновані нині методи та механізми, що використовуються у реальних системах голосування, не забезпечують реальної анонімності. Тому як в теоретичному, так і практичному сенсі актуальною та необхідною є проблема розроблення механізмів анонімного підрахунку голосів виборців з забезпеченням захищеності від їх викривлення [23].

Одним із складових механізмів вирішенням вказаної проблеми є використання механізму гомоморфного шифрування. Сутність гомоморфного шифрування полягає у тому, що існує деякий набір операцій, результат виконання яких над шифротекстами (з подальшим розшифруванням) співпадає з аналогічними діями над відкритими текстами [24]. Математично це можливо записати як

$$Dec(Enc(m_1) + Enc(m_2)) = m_1 + m_2 . \quad (1)$$

Тобто, гомоморфне шифрування дозволяє виконувати деякі обчислення над інформацією, при цьому не маючи доступу до самої інформації. Проте при спробі застосувати такі обчислення на практиці виникає ряд проблем. Основними з них є вибір методу асиметричного шифрування, що забезпечує необхідну криптографічну стійкість як від класичних, так і квантових атак, визначення можливих кандидатів асиметричних криптоперетворень при гомоморфному шифруванні, їх оцінка порівняння між собою, та, зрозуміло, вибір найбільш раціональних при заданій множині загроз та обмежень.

Метою цієї статі є обґрунтування можливостей, умов і обмежень щодо застосування стандартизованих асиметричних криптоперетворень при створенні сучасних гомоморфних перетворень типу шифрування, коли безумовно повинна бути забезпечена анонімність електронного голосування та практична реалізація анонімного голосування на основі доведення нульових знань.

1. Сутність та узагальнений попередній розгляд механізмів гомоморфного шифрування

Практично вперше на можливість гомоморфного шифрування у вигляді (1) увага була звернена після розробки алгоритму асиметричного RSA перетворення [24]. RSA є гомоморфним перетворенням відносно операції множення та задовольняє умові

$$\begin{aligned} &Dec(Enc(m_1) * Enc(m_2)) = Dec(m_1^e * m_2^e \bmod n) = \\ &= Dec((m_1 * m_2)^e \bmod n) = ((m_1 * m_2)^e)^d \bmod n = m_1 * m_2 \end{aligned} . \quad (2)$$

Згодом виявилося, що існує безліч інших перетворень, що задовольняють умовам (1) та (2). Якщо механізм має одне гомоморфне перетворення, то його називають частковим. Відносно нього виявилося, що вже за незначної кількості операцій, розрядна сітка переповнюється і механізм втрачає властивість гомоморфності. Така максимальна можлива кількість операцій називається максимальною глибиною гомоморфного перетворення.

Основними характеристиками гомоморфного перетворення є множина гомоморфних операцій та максимальна глибина обчислень. З теорії функціонального аналізу відомо, що у просторі обчислювальних функцій можливо виділити базис. Якщо криптосистема може обчислювати базисні функції, то може обчислювати будь-яку функцію, глибина обчислень якої не перевищує максимальну глибину, характерну для заданої системи. Традиційно, у якості базису обираються операції додавання та множення [24].

Перетворення, гомоморфні операції яких складають повний базис, а глибина обчислень яких може бути при правильному виборі параметрів скільки завгодно великою, отримали назву levelled homomorphic encryption (LHE). Якщо для механізму криптоперетворення існує такий набір загальносистемних параметрів, при якому можна обчислювати функції будь-якої складності, то вона має назву повністю гомоморфної системи криптоперетворення [3].

Аналіз показує, що в сучасній криптографії є значне число гомоморфних перетворень, які можуть бути реалізовані на базі різних алгебраїчних структур. Проте, досягти достатньої глибини для гомоморфних обчислень за адекватних вимог до обчислювальних ресурсів вдалося лише для систем на базі алгебраїчних решіток [3 – 7, 9 – 12]. Цьому сприяє багато факторів, одним з яких є те, що задачі теорії решіток можуть формулюватись мовою поліноміальних кілець, де точки, що належать решітці, представляються деякими поліномами. Таке поліноміальне представлення є достатньо зручним. Зазвичай повідомлення, що шифрується, входить до шифртексту лінійно, що створює природній гомоморфізм щодо операції складання. Схожа ситуація спостерігається і з гомоморфним множенням. Таким чином, природня підтримка базових гомоморфних операцій – додавання та множення, дозволяє реалізовувати оригінальні механізми гомоморфного шифрування.

Перша модель повністю гомоморфної криптографічної системи була запропонована Крейгом Джентрі в 2009 році. Після цього були запропоновані інші механізми гомоморфного перетворення, і на їх основі криптосистеми [9 – 12].

Механізм (схема) Джентрі має GGH-подібну конструкцію. У ньому в якості відкритого ключа виступає деякий "поганий" базис B_{pk} решітки J, разом з базисом B_I деякого ідеалу *I*. Зазвичай $I = (2)$. Безпосередньо шифротекст обчислюється як

$$c = 2r + m \bmod B_{pk}, \tag{3}$$

де m – повідомлення, закодоване у вигляді шуму, а $2r$ – деякий вектор, що належить решітці. При застосуванні (3) повідомлення кодується у вигляді шуму. Для декодування використовується редукований базис B_{sk}. Причому гомоморфність операцій досягається завдяки тому, що при складанні та множенні повідомлень внаслідок замкнутості сума та добуток векторів решітки переходитимуть у інший вектор на решітці, при цьому зберігаючи шум.

Розвитком ідей Джентрі є схема BGV [1]. Стійкість цієї схеми базується на складності вирішення проблеми навчання з помилками. Усі операції виконуються в кільці поліномів $\mathbf{Z}_q[X]/(X^n+1)$. При цьому секретний (особистий) ключ представляється як поліном, усі коефіцієнти якого належать до множини за деяким модулем, наприклад 2 ($\{0,1\}$). Відкритий ключ, обчислюється як пара поліномів

$$(p_0, p_1) = (-(a*s + t*e), a), \tag{4}$$

де a – деякий випадковий поліном, e – "поліном помилки", коефіцієнти якого розподілені за нормальним розподілом.

Безпосередньо шифротекст обчислюється як пара поліномів

$$(c_0, c_1) = (m + p_0 u + te_1 \bmod q,\ p_1 u + te_2 \bmod q). \quad (5)$$

У (5) перший поліном фактично представляє з себе суму повідомлення, маски та вектора шуму. Другий поліном потрібен для розшифрування шифротекста. Якщо обчислимо $c_0 + s * c_1$, то отримаємо поліном, що є сумою повідомлення та деякого невеликого адитивного шуму. Після приведення за модулями t, q шум зникне за умови, що він не перевищував максимальної величини. Шум в схемі BGV представляється величиною виду $t * e$, де t – ціле число, а e – поліном, коефіцієнти якого розподілені за нормальним законом.

Наразі популярною модифікацією схеми BGV є схема BFV [21], у якій запропоновано ряд покращень, що дозволяють краще керувати рівнем шуму та швидкодією. Зокрема, шифротекст обчислюється у вигляді

$$(c_0, c_1) = (\lfloor q / t \rfloor m + p_0 u + e_1 \bmod q,\ p_1 u + e_2 \bmod q). \quad (6)$$

Схеми BFV та BGV і досі залишаються одними з найкращих схем і знаходять своє застосування на практиці.

Також важливим розвитком попередніх ідей є схема CKKS [6]. Справа в тому, що для багатьох перетворень, під час гомоморфних обчислень, виникає потреба апроксимувати певні величини і працювати вже з ними. Ідея такої схеми полягає у тому, щоб представити шум, отриманий під час апроксимації, як частину шуму, отриманого під час шифрування. Цим кодування повідомлень в CKKS дещо відрізняється від попередніх схем. В CKKS схемі повідомлення, як і в попередніх схемах, є поліномом в $\mathbf{Z}_q[X]/(X^n + 1)$, проте для відображення вихідного повідомлення на поліном застосовується канонічне вкладення (canonical embedding map), сутність якого полягає у тому, що між деякою адитивною підгрупою векторів в просторі $\mathbb{C}^{\phi(M)}$ та $\mathbf{Z}_q[X]/(X^M + 1)$ існує гомоморфізм. В цьому випадку повідомлення є вектором комплексних чисел $z = (z_i)_{i \in 0..\phi(M)}$, який відображається в циклотомічне кільце.

Загальна схема такого перетворення наведена на рис 1.

$$\begin{array}{ccccccc} \mathbb{C}^{\phi(M)/2} & \xrightarrow{\pi^{-1}} & \mathbb{H} & \xrightarrow{\lfloor\cdot\rceil_{\sigma(\mathcal{R})}} & \sigma(\mathcal{R}) & \xrightarrow{\sigma^{-1}} & \mathcal{R} \\ z = (z_i)_{i \in T} & \longmapsto & \pi^{-1}(z) & \longmapsto & \lfloor \pi^{-1}(z) \rceil_{\sigma(\mathcal{R})} & \longmapsto & \sigma^{-1}\left(\lfloor \pi^{-1}(z) \rceil_{\sigma(\mathcal{R})}\right) \end{array}$$

Рис. 1. Схема кодування повідомлень в CKKS

Аналіз показує, що в цілому структура шифротексту в CKKS схемі схожа на BGV. Детальний огляд схеми наведено в [6].

Іншим типом повністю гомоморфних систем є схеми на базі проблеми NTRU [5]. Секретним ключем в таких схемах є деякий поліном f, що має зворотній елемент в відповідному полі. Відкритим ключем є поліном $h = 2gf^{-1} \bmod q$. Для шифрування повідомлення m, шифротекст обчислюється як

$$c = h * s + 2e + m \bmod q, \quad (7)$$

де s, e – деякі малі поліноми.

Для розшифрування достатньо обчислити $m = fc \bmod q$, причому гомоморфність операції визначаються наступним чином:

$$\begin{cases} add(c_1, c_2) = c_1 + c_2 \bmod q \\ mult(c_1, c_2) = c_1 * c_2 \bmod q \end{cases}. \quad (8)$$

Важливою властивістю наведених схем є те, що вони дозволяють легко будувати мультиключові протоколи. Для попередніх схем вважалося, що обчислення відбуваються на одному й тому ж ключі, проте для схем на базі NTRU природнім чином зберігається можливість мультиключових обчислень. Для розшифрування достатньо обчислити $m = f_1 ... f_n c \bmod q$. Проте, як буде показано далі, існують деякі алгебраїчні аспекти, що потребують детальної уваги при їх реалізації.

2. Узагальнений аналіз рівня безпеки перспективних схем гомоморфного шифрування

Проведений аналіз дозволив визначити, що безпека найбільш перспективних схем зводиться до наступних проблем:

- NTRU проблема;
- LWE (Learning With Errors) проблема;
- R-LWE (Ring Learning With Errors) проблема.

Проблема NTRU у загальному вигляді полягає в знаходженні пари поліномів $f, g \in \mathbf{Z}_q[X]/(\tilde{f}(n))$, для яких виконується умова

$$h = f^{-1} * g \bmod q \in \mathbf{Z}_q[X]/(\tilde{f}(n)). \quad (9)$$

Причому, для заздалегідь заданого поліному h всі коефіцієнти поліномів f, g є меншими за певну величину, що визначається з вимог безпеки. Альтернативним визначенням, в термінах теорії решіток, є знаходження досить малого вектору (задача SVP_γ) на решітці:

$$\Lambda_h^q = \{(f, g) \in \mathbf{Z}_q^2[X]/(\tilde{f}(n)) \mid h * f - g = 0 \bmod q\}. \quad (10)$$

Аналіз показує, що для вирішення цієї задачі можливо використовувати як методи перебору, так і методи, що базуються на редукції решіток. Одним з найкращих відомих методів криптоаналізу проблеми NTRU є гібридна атака, яка поєднує методи редукції решіток з переборними методами [25].

Основна ідея методу гібридної атаки полягає у тому, щоб розділити шуканий вектор v на дві частини v_1, v_2. Друга частина знаходиться перебором усіх можливих варіантів, наприклад за допомогою метода “зустріч посередині” (meet-in-the-middle, MITM), який дозволяє значно покращити звичайний перебір. Після того, як друга частина v_2 знайдена, вектор $v' = (0, v_2)$ можливо розглядати як зашумлений вектор v. Маючи зашумлений вектор, за умови, що шум достатньо низький, за поліноміальний час можливо знайти вектор v за допомогою алгоритму Бабаї [25], якщо частина решітки, що відповідає вектору v_1, буде достатньо редукована.

У цілому гібридна атака для проблеми NTRU складається з наступних кроків:

1. Необхідно обрати межу, за якою шуканий вектор v буде розділений на дві частини v_1, v_2.
2. Редукувати частину решітки, що відповідає v_1.
3. За допомогою метода “Зустріч посередині” знайти v_2.
4. За допомогою алгоритму Бабаї знайти вектор.

Слід відмітити, що кроки 2 і 3 можливо виконувати паралельно, що дозволяє значно зменшити складність (час) здійснення атаки. Причому, мінімальний результат, очевидно, досягається тоді, коли час виконання кроків 2 і 3 буде рівним. Час виконання кроку 3 можливо оцінити порівняно легко. Але для кроку 2 ця задача складніша.

Найкращими відомими методами редукції решіток є методи, що базуються на алгоритмі блочної редукції Коркіна – Золотарьова (BKZ) [26]. Суть метода полягає у тому, щоб реду-

кувати не всю решітку одночасно, а лише її частини в відповідних підпросторах. Причому, для редукції в підпросторі використовується деякий «Вирішувач», який є загальносистемним параметром. Зазвичай використовуються модифікації метода LLL, оскільки вони себе гарно показали на решітках малої розмірності [26]. Проблема оцінки часу редукції за допомогою методу BKZ є досить складною. Це тому, що BKZ виконує поліноміальну кількість викликів до Вирішувача, проте точної оцінки кількості викликів та точної оцінки часу роботи досі не знайдено. Більш того, за останні роки були знайдені нові більш ефективні вирішувачі. У тому числі був запропонований симулятор BKZ [26], який дозволяє робити більш-менш точні оцінки, проте з ростом розмірності решітки падає і точність її оцінки. Такий симулятор на основі розмірності решітки n, довжини блока β, кількості ітерацій m та фактору Ерміта δ дозволяє визначити час редукції решітки.

Наразі оцінки часу роботи BKZ є емпіричною та базуються на результатах обчислювальних експериментів. Останні оцінки зведені в табл. 1.

Таблиця 1

Емпіричні оцінки часу роботи алгоритму BKZ

Модель	Значення
Sieve [27]	$d^3 * B^2 + 2^{0.292*\beta+16.4+\log_2(8d)}$
Qsieve [28, 29]	$d^3 B^2 + 2^{0.265*\beta+16.4+\log_2(8d)}$
Lp [30]	$d^3 B^2 + 2^{1.8/\log_2 \delta_0 - 110 + \log_2 2.3^9}$
enum [26]	$d^3 B^2 + 8d * 2^{0.270188776350190*\beta*\log(\beta) - 1.0192050451318417\beta + 16.10253135200765 + \log_2 100}$

Позначення: d – розмірність підрешітки, на якій виконувалася редукція. $B = \log_2 q$ кількість бітів, δ_0 – фактор Ерміта, β – оптимальний розмір блока

У більшості NTRU-подібних використовуються циклотомічні поля виду $\mathbf{Z}_q[X]/(X^n - 1)$ та $\mathbf{Z}_q[X]/(X^{2^n} + 1)$ [25]. Такий вибір дозволяє створювати ефективні реалізації, оскільки для циклотомічних полів існують гомоморфізми, що дозволяють максимально ефективно реалізувати NTT та FFT. Проте, нажаль, складна структура поля робить систему вразливою при досить великих q відносно n.

Ідея атак такого типу базується на теорії Галуа, яка стверджує, що підполя нормального сепарабельного розширення деякого поля мають взаємно однозначне відношення з нормальними підгрупами групи автоморфізмів, що залишають незмінними елементи базового поля. Для аналізу таких конструкцій на полі визначається та використовується норма.

Позначимо циклотомічне поле як $K = \mathbf{Z}_q[X]/(X^n + 1)$. Припустимо, що існує деяке підполе $L \supset K$. Тоді група Галуа $G = Gal(K/L)$ складається з множини перестановок елементів, приєднанням яких до L отримується K. Для циклотомічних полів це відображення виду $\sigma_i(\zeta) = \zeta^i$. Нормою елемента $a \in K$ відносно під поля L є

$$N_{K/L}(a) = \prod_{\sigma \in G} \sigma(a). \tag{11}$$

Вказана норма має декілька цікавих властивостей, які застосовуються для побудови атаки. По-перше, вона є мультиплікативною $N_{K/L}(a*b) = N_{K/L}(a) * N_{K/L}(b)$. По-друге, норма, відносно підполя, завжди лежить у цьому підполі. Ідея полягає у тому, що замість пошуку вектору (g, f) на решітці Λ_h^q можна шукати вектор $(N_{K/L}(g), N_{K/L}(f))$ на решітці $\Lambda_{N_{K/L}(h)}^q$. Оскільки поліном f є дільником $N_{K/L}(f)$ (нормальна підгрупа завжди містить одиницю), то маючи норму, можна швидко знайти поліном, що відповідає цій нормі, якщо його коефіцієнти є достатньо малими. Решітка $\Lambda_{N_{K/L}(h)}^q$ має значно меншу розмірність, ніж Λ_h^q. Це відбува-

ється тому, що норма є розрядженим поліномом (більшість коефіцієнтів тотожно рівні 0). Завдяки цьому, деяка частина базисних векторів решітки обнуляється.

Для вдалої атаки потрібно, щоб модуль q значно перевищував n. Для більшості не гомоморфних схем це виконується, але гомоморфні обчислення часто потребують досить великих модулів, які експоненційно залежать від n, що дозволяє в окремих випадках виконати атаку навіть за поліноміальний час. Це значно обмежує застосування NTRU-подібних систем з циклотомічними полями для вирішення реальних задач. Можливим рішенням може стати використання поля $\mathbf{Z}_q[X]/(X^n - X - 1)$, як в стандарті ДСТУ 8961:2019. Проте, при використанні цього поля, неможливо ефективно використовувати NTT і виникає потреба в створенні ефективних алгоритмів множення.

2.1. Аналіз та порівняння безпеки LWE-подібних асиметричних криптосистем

Проблема навчання з помилками (LWE) визначається наступним чином [17]. Нехай n, q є деякими натуральними числами, χ – деякий ймовірнісний розподіл над $\mathbf{Z}$ та s – секретний вектор у $\mathbf{Z}_q^n$. Ймовірнісний розподіл $L_{s,\chi}$ над $\mathbf{Z}_q^n \times \mathbf{Z}_q$ отримується обчисленням

$$(a, c) = (a, \langle a, s \rangle + e) \in \mathbf{Z}_q^n \times \mathbf{Z}_q, \tag{12}$$

де $a \in \mathbf{Z}_q^n$ отримується з рівномірного розподілу та $e \in \mathbf{Z}$ з розподілу χ. Decision-LWE полягає у тому, щоб визначити, чи отримана пара $(a, c) \in \mathbf{Z}_q^n \times \mathbf{Z}_q$ з розподілу $L_{s,\chi}$ або рівномірного розподілу. Search-LWE полягає у знаходженні s з пари $(a, c) \in \mathbf{Z}_q^n \times \mathbf{Z}_q$. Проблеми Decision-LWE та Search-LWE є еквівалентними з точки зору теорії складності та можуть бути зведені одне до одного за поліноміальний час і фактично є різними поглядами на одну і ту ж задачу. Розподіл χ зазвичай є дискретним нормальним розподілом над кінцевим полем з математичним очікуванням рівним 0 та дисперсією, що характеризується параметром α. Більшість атак на LWE полягають у знаходженні деякого вектору v з певною нормою на решітці L з фіксованим об'ємом $vol(L)$, але з різною розмірністю m, яка фактично характеризує оптимальну кількість пар $(a_i, c_i) \in \mathbf{Z}_q^n \times \mathbf{Z}_q$ необхідних для атаки.

Складність проблеми навчання з помилками точно знайдена лише асимптотично. Доведено, що за певних умов складність вирішення LWE в просторі розмірності n становить щонайменше $2^{O(n)}$ [17]. Цей результат зручно використовувати для оцінки загальносистемних параметрів, проте конкретні оцінки складності криптостійкості досі не відомі. Це пов'язано з тим, що атаки на LWE, в кінцевому випадку, зводяться до редукції решіток. В останні 10 років є суттєвий прогрес у цьому напрямку, що призводить до постійної зміни оцінок. В більшості сучасних криптосистем використовуються варіанти LWE над поліноміальними кільцями (PRLWE), тобто, розподіл не над $\mathbf{Z}_q$, а над $\mathbf{Z}_q[X]/(f(x))$. Часто використовується поліном $f(x) = x^n + 1$ і відповідне поле $R_q = \mathbf{Z}_q[X]/(x^n + 1)$. Коли $(a_i, c_i) \in R_q \times R_q$, то задача має назву RLWE. Коли $(a_i, c_i) \in R_q^d \times R_q$ – MLWE відповідно.

Поліном $f(x) = x^n + 1$ має цікаві властивості, які використовуються для доказу криптобезпеки. Також, його властивості дозволяють використати NTT для створення ефективних реалізацій. Однак, з теорії Галуа відомо, що $R_q = \mathbf{Z}_q[X]/(x^n + 1)$ має складну структуру підполів, що може бути використано для криптоаналізу. Для гомоморфних систем шифрування це питання є особливо важливим. Фактично, сучасними криптологами проблеми R-LWE та M-LWE розглядаються як LWE, оскільки для полінома $f(x) = x^n + 1$ доведено, що R-LWE та M-LWE є складнішими за LWE.

При криптоаналізі більшість дослідників розглядають тільки атаки на решітках. Фактично, атаки такого роду полягають у знаходженні деякого вектору v, що лежить на решітці L та має норму $\|v\|$ не більшу за певну величину. Для атаки на LWE відбувається зведення до інших задач у теорії решіток, які у свою чергу вирішуються вже відомими алгоритмами.

2.1.1. Аналіз захищеності від атаки LWE->BDD

Припустимо, що дано m пар $(a_i, c_i) = (a_i, \langle a_i, s_i \rangle + e_i) \in \mathbf{Z}_q^n \times \mathbf{Z}_q$ [17, 31]. Це можливо записати у більш зручному вигляді:

$$(A, c) = (A, A*s + e) \in \mathbf{Z}_q^{m \times n} \times \mathbf{Z}_q^{m \times 1}. \tag{13}$$

Тоді, можливо побудувати решітку $L = \{Ax \bmod q : x \in \mathbf{Z}_q^m\}$. Очевидно, що s вектор на решітці є найближчим до вектору $As + e$. Задача знаходження найближчого вектору на решітці до деякого довільного вектору має назву BDD та вирішується за допомогою алгоритму Бабаї [25]. Алгоритм працює за поліноміальний час, проте знаходить рішення тільки з деякою ймовірністю. Для LWE цю ймовірність можливо оцінити як:

$$\prod_{i=0}^{m-1} erf\left(\frac{\|b_i^*\|\sqrt{\pi}}{2\alpha q}\right), \tag{14}$$

де $\|b_i^*\|$ – норми ортогоналізованих за Граммом – Шмідтом векторів базису решітки (тобто стовбців матриці A). Для того щоб ймовірність вирішення BDD була близька до одиниці, потрібно зменшити $\|b_i^*\|$, тобто редукувати базис.

2.1.2. Аналіз захищеності від атаки Dual Attack (LWE->SIS)

Існують атаки на дуальній до Λ_h^q решітки [31]. Побудуємо наступну решітку $L = \{x \in \mathbf{Z}_q^m \mid A*x = 0 \bmod q\}$. Задача SIS полягає у знаходженні такого найменшого $x \in \mathbf{Z}^n$, щоб $A*x = 0$. Припустимо, що такий вектор знайдений. Тоді можна вирішити задачу Decision-LWE. Нехай дано m пар $(A, c) = (A, A*s + e) \in \mathbf{Z}_q^{m \times n} \times \mathbf{Z}_q^{m \times 1}$. Обчислимо скалярний добуток $\langle x, c \rangle$: $\langle x, c \rangle = x*a*s + x*e = 0*s + x*e = x*e = \langle x, e \rangle$.

Оскільки вектор $x \in \mathbf{Z}^n$ відомий, то з цієї рівності можна знайти значення вектору помилок e, хоча простір помилок і залишається досить великим. Доведено [31], що, якщо вектор x має норму

$$\|x\|_2 = \frac{1}{\alpha} * \sqrt{\frac{\ln(\frac{1}{\varepsilon})}{\pi}}, \tag{15}$$

то з ймовірністю близькою до 1 можливо вирішити задачу і при цьому знадобиться $\frac{1}{\varepsilon^2}$ запусків Вирішувача SIS. Вирішувач фактично знаходить достатньо малий вектор на решітці, тобто вирішує задачу SVP. Фактор Ерміта δ_0 при цьому повинен бути [31] не більше

$$\log \delta_0 = \frac{\log^2(\frac{1}{\alpha}\sqrt{\frac{\ln(\frac{1}{\varepsilon})}{\pi}})}{4*n\log q}. \tag{16}$$

Атаки такого типу називаються Dual Attack. Точна оцінка атаки потребує вибрати певний Вирішувач. У якості Вирішувача можливо взяти BKZ 2.0 і виконати оцінку як для атаки на BDD.

2.1.3. Аналіз захищеності від Primal Attack (LWE->uSVP)

У атаці, що описана в п. 2.1.1, решітка містить вектор s. Ідея Primal Attack полягає у тому, щоб побудувати таку решітку, на якій буде лежати вектор $(s,e,1)$ і він буде найменшим унікальним вектором (задача uSVP) [31]. Такою решіткою буде

$$\Lambda = \{x \in \mathbf{Z}^{m+n+1} : (A \mid I_m \mid -c) * x = 0 \bmod q\}. \quad (17)$$

Відповідно, для пошуку вектору можливо скористатися BKZ 2.0 і редукувати решітку. Тоді b_0 буде шуканим рішенням. Оцінити фактор Ерміта для вдалої редукції можливо як

$$\log \delta_0 = \frac{1}{4n^2 \ln^2 q}\left(W\left((-2n\ln q) * (\sqrt{n \log q}) * \frac{(\tau\alpha)^2}{2\pi}\right)\right)^2. \quad (18)$$

3. Порівняння асиметричних схем гомоморфного шифрування

Попередній аналіз показав, що розробка необхідного програмного забезпечення є дуже складним процесом. По суті воно розробляється вже десятки років та доступне в вигляді відкритих бібліотек. Розглянуті вище асиметричні схеми, стосовно яких наведені оцінки стійкості, реалізовані в декількох існуючих основних бібліотеках [32 – 34].

Для тестування та порівняння було обрано такі бібліотеки як SEAL, HeLib, cuHe. У якості параметрів було обрано $n = 4096$ та $\log(q) = 109$. Для порівняння використовувався метод аналізу ієрархій [35]. Характеристики обраних схем наведено в табл. 2, а також відображені на діаграмах (рис. 2 – 6). Шум та стійкість було оцінено за 10-бальною шкалою у порівнянні один з одним.

Таблиця 2

Характеристики схем гомоморфного шифрування

	Показник	Схема			
		BGV	BFV	CKKS	LTV
1	Час розгортання ключів (ms)	319658	305937	5754892	4258114
2	Швидкість зашифрування (ms)	4538	8465	126751	3089
3	Швидкість розшифрування (ms)	916	1256	1573	341
4	Шум	6	5	5	7
5	Стійкість	6	7	7	5

Рис. 2. Діаграма часу розгортання ключів

Рис. 3. Діаграма часу зашифрування

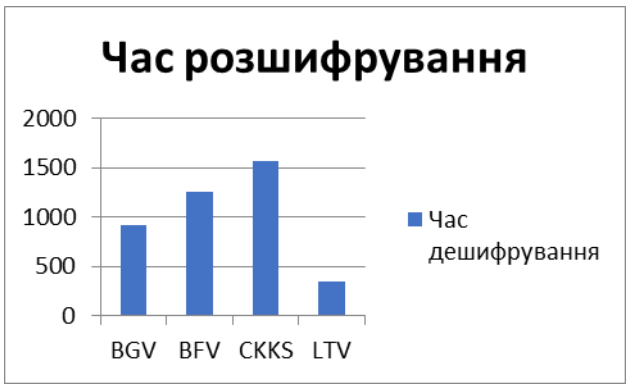

Рис. 4. Діаграма часу розшифрування

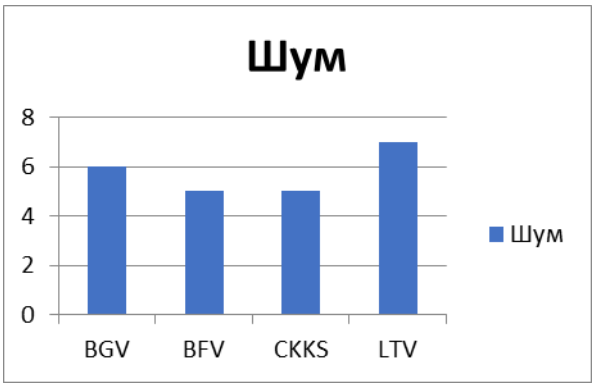

Рис. 5. Діаграма показників шуму

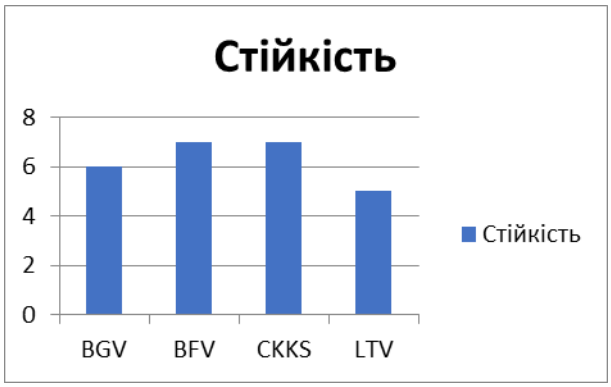

Рис. 6. Діаграма показників стійкості

В подальшому будемо розглядати задачу вибору перспективного шифру у вигляді певної цілі X_0. На даному етапі поки що незрозуміло, яким чином можна досягнути головної мети. Тому здійснимо для головної цілі процедуру декомпозиції та побудуємо дерево цілей [35].

Оскільки змістовна модель визначена у якості об'єкта (схема), то необхідно для першого кроку декомпозиції вибрати модель як основу, що дозволить отримати необхідну сукупність ознак розбиття головної цілі на її складові – підцілі. У якості такої моделі обираємо модель-діяльність [35].

З цією метою у якості підцілей першого рівня, можна використати різні показники якості функціонування шифру:

- технічні показники X_1^1;
- показники цільового призначення X_2^1;

Таким чином, маємо піддерево цілей №1 (рис. 7).

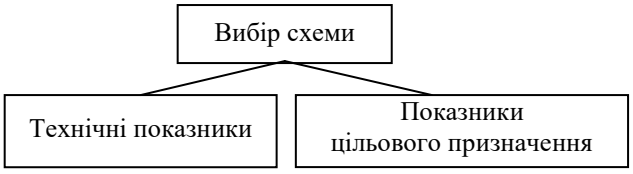

Рис. 7. Піддерево №1

З метою спрощення складності рішення задачі виберемо лише декілька показників для кожної із зазначених вище груп та визначимо піддерева цілей для кожної з них (рис. 8, 9).

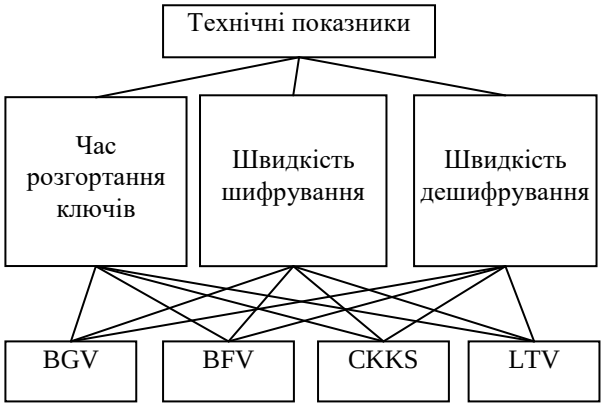

Рис. 8. Піддерево №2

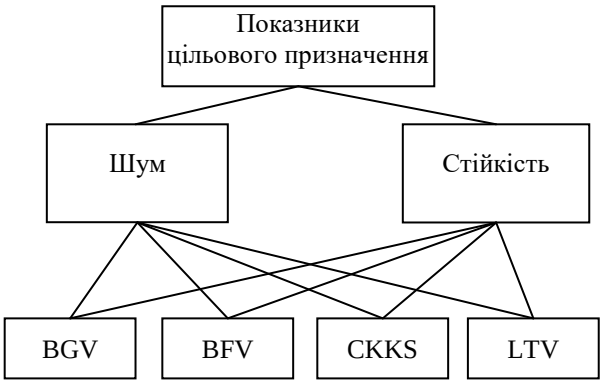

Рис. 9. Піддерево №3

У подальшому будемо використовувати наступні позначення: X_1^2 – час розгортання ключів, X_2^2 – швидкість зашифрування, X_3^2 – швидкість розшифрування, X_4^2 – шум, X_5^2 – стійкість.

Таким чином, дерево цілей буде мати наступний вигляд (рис. 10).

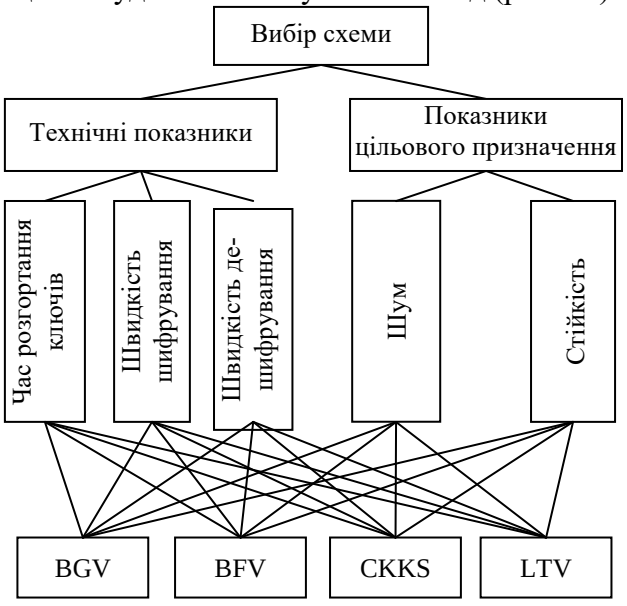

Рис. 10. Дерево цілей

Оцінки значущості вкладу підцілей у досягнення цілі вищого рівня, згідно методу аналізу ієрархій, здійснюються зверху вниз парним порівнянням. Сутність парного порівняння,

наприклад X_i^1 та X_j^1 відносно X^0 до цілі полягає у оцінці (суджень) того, у якій мірі X_i^1 більш важлива (більш вагома) для досягнення цілі X^0, ніж підціль X_j^1. Позначимо цю оцінку через $a_{ij}^{(1)}$. Подібні оцінки надаються експертами та носять суб'єктивний характер. При нашому порівнянні було вживано наступну шкалу оцінок (табл. 3).

Таблиця 3

Шкала оцінок

Перевага X_i^1 над X_j^1	Відсутня	Помірна	Значна	Велика	Дуже велика	Проміжні оцінки
$a_{ij}^{(1)}$	1	3	5	7	9	2, 4, 6, 8

Результати оцінок заносяться у таблиці (матриці) для підцілей r-го рівня.

У лівому стовпці та першому (верхньому) рядку записуються цілі, що порівнюються. У верхній лівій клітинці записується ціль, по відношенню до якої оцінюються підцілі нижчого рівня.

Отримані експертні оцінки підлягають обробці наступним чином:

- обчислюється середнє геометричне для кожного рядка:

$$q_j^{(r-1)} = \sqrt[t_r]{a_{j1}^{(r)} \ldots \times a_{jj}^{(r)} \times a_{jt_r}^{(r)}}; \tag{19}$$

- обчислюються нормовані значення:

$$\gamma_j^{(r-1)} = \frac{q_j^{(r-1)}}{\sum_{i=1}^{t_r} q_i^{(r-1)}}, \tag{20}$$

де $\gamma_j^{(r-1)}$ характеризує значущість цілі $X_j^{(r)}$ для цілі $X^{(r-1)}$. А сукупність усіх $\gamma_j^{(r-1)}$ складає вектор-стовпчик.

Оскільки ми розбили дерево на ряд піддерев, використаємо приведений вище алгоритм послідовно для всіх дерев починаючи з дерева №1 (рис. 1).

Оцінимо важливість показників, що розглядаються (табл. 4).

Таблиця 4

Шкала оцінок X^0

X^0	X_1^1	X_2^1	$q_j^{(0)}$	$\gamma_j^{(0)}$
X_1^1	1	1/2	0.7	0.33
X_2^1	2	1	1.4	0.66

$$\left\| Y_1^{10} \right\| = \begin{pmatrix} 0.33 \\ 0.66 \end{pmatrix}$$

Бачимо, що показники цільового призначення є більш вагомими при порівнянні.

Далі проведемо порівняння важливості технічних показників та показників цільового рівня окремо (табл. 5, 6).

Таблиця 5

Шкала оцінок X_1^1

X_1^1	X_1^2	X_2^2	X_3^2	$q_j^{(1)}$	$\gamma_j^{(1)}$
X_1^2	1	3	1/7	0.754199	0.247
X_2^2	1/3	1	5	1.185615	0.388
X_3^2	7	1/5	1	1.118689	0.366

$$\left\| Y_1^{21} \right\| = \begin{pmatrix} 0.247 \\ 0.388 \\ 0.366 \end{pmatrix}$$

Таблиця 6

Шкала оцінок X_2^1

X_2^1	X_4^2	X_5^2	$q_j^{(1)}$	$\gamma_j^{(1)}$
X_4^2	1	1/6	0.408248	0.143
X_5^2	6	1	2.449489	0.857

$$\left\| Y_2^{21} \right\| = \begin{pmatrix} 0.143 \\ 0.857 \end{pmatrix}$$

Тобто, серед технічних показників найбільш вагомою є швидкість шифрування, а серед показників цільового призначення – стійкість.

Далі, аналогічно попереднім порівнянням, проведемо порівняння технічних показників для кожної схеми окремо та зазначимо отримані результати:

$$\left\| Y_{1-3}^{32} \right\| = \begin{pmatrix} 0.273 & 0.152 & 0.298 \\ 0.053 & 0.633 & 0.087 \\ 0.063 & 0.072 & 0.061 \\ 0.611 & 0.143 & 0.553 \end{pmatrix}$$

Аналогічно зазначимо результати порівняння показників цільового призначення для кожної схеми окремо:

$$\left\| Y_{4-5}^{32} \right\| = \begin{pmatrix} 0.193 & 0.182 \\ 0.368 & 0.364 \\ 0.368 & 0.364 \\ 0.070 & 0.091 \end{pmatrix}$$

Розраховуємо вклад цілей третього рівня для кожного з піддерев:

$$\left\| Y_1^{31} \right\| = \left\| Y_{1-3}^{32} \right\| \times \left\| Y_1^{21} \right\| = \begin{pmatrix} 0.273 & 0.152 & 0.298 \\ 0.053 & 0.633 & 0.087 \\ 0.063 & 0.072 & 0.061 \\ 0.611 & 0.143 & 0.553 \end{pmatrix} \times \begin{pmatrix} 0.247 \\ 0.388 \\ 0.366 \end{pmatrix} = \begin{pmatrix} 0.235475 \\ 0.290537 \\ 0.065823 \\ 0.408799 \end{pmatrix}$$

$$\left\| Y_2^{31} \right\| = \left\| Y_{4-5}^{32} \right\| \times \left\| Y_2^{21} \right\| = \begin{pmatrix} 0.193 & 0.182 \\ 0.368 & 0.364 \\ 0.368 & 0.364 \\ 0.070 & 0.091 \end{pmatrix} \times \begin{pmatrix} 0.143 \\ 0.857 \end{pmatrix} = \begin{pmatrix} 0.053625 \\ 0.104676 \\ 0.104676 \\ 0.023023 \end{pmatrix}.$$

Таким чином, для піддерева, що відображає оцінки технічних показників, найкращою є схема LTV, а для піддерева, що відображає оцінки показників цільового призначення, найкращими є дві схеми BFV та CKKS.

Розрахуємо вклад цілей третього рівня в досягнення головної цілі та представимо отримані результати у вигляді діаграми (рис. 11):

$$\left\|Y_1^{30}\right\| = \left\|Y_{1-2}^{11}\right\| \times \left\|Y_1^{10}\right\| = \begin{pmatrix} 0.235475 & 0.053625 \\ 0.290537 & 0.104676 \\ 0.065823 & 0.104676 \\ 0.408799 & 0.023023 \end{pmatrix} \times \begin{pmatrix} 0.33 \\ 0.66 \end{pmatrix} = \begin{pmatrix} 0.11309925 \\ 0.16496337 \\ 0.09080775 \\ 0.15009885 \end{pmatrix}.$$

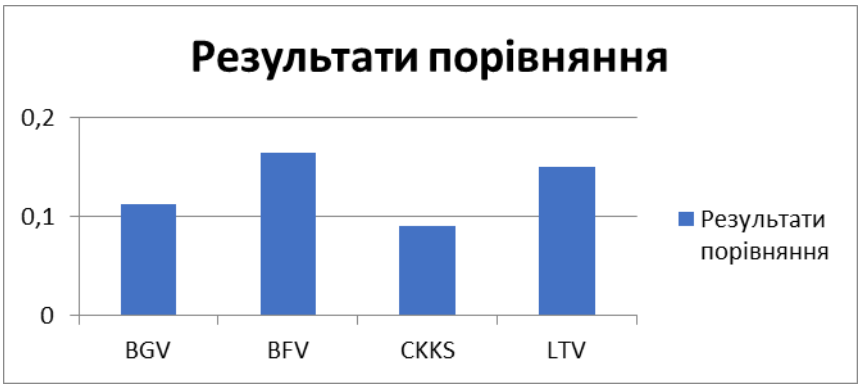

Рис. 11. Результати порівняння

Таким чином, для досягнення головної цілі, тобто вибору кращої схеми, перевагу має BFV (0.165), LTV (0.15), BGV (0.113), CKKS (0.091). Але слід зазначити, що схему необхідно обирати згідно потреб і цілей, для яких вона буде застосована.

4. Обґрунтування методу асиметричного шифрування з нульовими знаннями

Для створення механізму шифрування з верифікацією, зручно скористатися схемою, представленою в роботі [36]. Автори запропонували використовувати фреймворк Фіата – Шаміра для доказів з нульовим розголошенням. Фактично вони додають до криптограми доказ з нульовим розголошенням, який дозволяє перевірити, що деяке повідомлення, закодоване в поліномі m, задовольняє умові

$$B \cdot m \equiv u \bmod t, \tag{21}$$

де (B, r) є заздалегідь відомими параметрами, що однозначно задають множину валідних повідомлень.

Нехай в якості такого валідного значення задається як вектор у $R_q^{N_k}$, де N_k – кількість кандидатів. Разом з цим, у кожного полінома всі коефіцієнти дорівнюють 0, окрім одного молодшого коефіцієнта в одному поліномі. Припустимо, що для валідного значення (голосу) справедливою буде формула

$$f(a) = \sum_{i=1}^{N_k} \left(a_i^2 - a_i\right)^2 + \left(\sum_{i=1}^{N_k} a_i\right)^2 - 1 = 0, \tag{22}$$

де $a_i \in R_q$ – відповідні поліноми.

Для такого значення голосу $a = (a_1, \ldots, a_{N_k}) \in R_q^{N_k}$ обчислимо вектор

$$a' = \left(f(a) + g_1, f(a) + g_2, \ldots, f(a) + g_{N_k}\right), \tag{23}$$

де g_i – наперед задані поліноми (загальносистемні параметри).

За даних умов параметрами B, u та m з формули (8) є: динамічним чином згенерована для кожного голосуючого випадкова матриця $B \in R_q^{N_k \times N_k}$, вектор $u = B \cdot \left(g_1, g_2, \ldots, g_{N_k}\right)$ та вектор-голос m, які потім передаються голосуючому для формування доказу.

Доказ є парою поліномів (c, z). Поліном c формується таким чином, щоб його могла сформувати перевіряюча сторона, використовуючи поліном z, та порівняти з переданим у доказі. Поліном c належить до множини малих поліномів з умовою, що кількість ненульових елементів у ньому не більше, ніж деяке γ, тобто $c \in R_2$, $\|c\|_\infty = \gamma$.

Далі, нехай задана деяка криптографічна геш-функція $H : \{0,1\}^* \to \{u \mid u \in R_2, \|u\|_\infty = \gamma\}$, тоді шифрування з верифікацією для схеми BFV на відкритому ключі $pk = (p_0, p_1)$ для деякого повідомлення $m \in \square_t [X] / (x^n + 1)$, що належить до множини, яку можливо верифікувати за допомогою пари (B, r), буде виглядати наступним чином:

1. Сформувати поліноми u, e_1, e_2 для гомоморфного шифрування.

2. Отримати шифротекст (c_{0m}, c_{1m}) з використанням сформованих на кроці 1 поліномів.

3. Сформувати поліноми u_y, e_{1y}, e_{2y}, m_y.

4. Зашифрувати повідомлення m_y з використанням поліномів u_y, e_{1y}, e_{2y} та отримати шифротекст (c_{0y}, c_{1y}).

5. Обчислити поліном

$$c = H\left(p_0, p_1, B, r, c_0, c_1, c_{0y}, c_{1y}, B \cdot m_y \pmod t\right)$$

6. Обчислити $z = \left(u \cdot c + u_y, e_1 \cdot c + e_{1y}, e_2 \cdot c + e_{2y}, m \cdot c + m_y\right)$.

7. Перевірити, що z задовольняє умовам схеми Фіата-Шаміра.

8. Повернути шифротекст (c_0, c_1) та доказ (c, z).

Для того щоб верифікувати повідомлення, потрібно повторно обчислити поліном c за допомогою z та порівняти його з оригінальним поліномом. Повна процедура верифікації шифротексту (c_0, c_1) за доказом (c, z) виглядатиме наступним чином:

1. Перевірити, що z задовольняє умовам схеми Фіата-Шаміра.

2. Обчислити шифротекст від z : (c_{0z}, c_{1z}).

3. Обчислити поліном:

$$c' = H\left(p_0, p_1, B, r, c_0, c_1, c_{0z} - c \cdot c_0 \pmod q, c_{1z} - c \cdot c_1 \pmod q, B \cdot z_4 - c \cdot r \pmod t\right)$$

4. Якщо $c = c'$ то повернути true, інакше повернути false.

Очевидно, що верифікація пройде успішно, тільки, якщо буде виконуватись наступна система умов:

$$\begin{aligned} & c_{0z} - c \cdot c_0 \equiv c_{0y} \pmod q \\ & c_{1z} - c \cdot c_0 \equiv c_{1y} \pmod q \\ & B \cdot z_4 - c \cdot r \equiv B \cdot m_y \pmod t \end{aligned} \quad . \tag{24}$$

Перші дві умови будуть виконуватися тільки тоді, коли при формуванні z дійсно використовувалося повідомлення m. Третя умова виконується тільки тоді, коли виконується порівняння (21).

Таким чином, наведена вище схема верифікації шифротекстів BFV на основі лінійної залежності $B \cdot m \equiv u \bmod t$ дозволяє здійснити захищену верифікацію шифртексту. Розроблена схема може використовуватися зокрема для систем електронного голосування. В поєднанні з технікою батчінга для повністю гомоморфних схем, запропонована система може сильно

зменшити як складність обчислення, так і об'єм інформації в відповідних криптографічних протоколах.

Висновки

1. Концепція гомоморфного шифрування з'явилася майже одразу після винайдення асиметричних криптосистем, проте тривалий час вона залишалася цікавою можливістю. Побудувати повністю гомоморфні системи для застосування на практиці вдалося відносно нещодавно за допомогою перетворень поліноміальних кілець.

2. Побудова сучасних повністю гомоморфних систем базується на проблемах навчання з помилками (LWE) та NTRU. В цілому LWE-подібні системи вважаються стійкішими до атак, проте під час їх криптоаналізу використовується чимало евристик. Стійкість NTRU зводиться до проблеми знаходження найменшого вектору і є більш зрозумілою.

3. NTRU-подібні повністю гомоморфні системи швидші за LWE-подібні, проте вразливі до алгебраїчних атак на циклотомічні кільця. Захиститися від атак можливо правильним вибором поля. При використанні поля $\mathbf{Z}_q[X]/(X^n - X - 1)$, наприклад, як у стандарті ДСТУ 8961:2019, забезпечується надійний захист від атак такого роду, оскільки група Галуа полінома $X^n - X - 1$ є симетричною групою S_n.

4. На основі порівняння визначено, що для застосування в системах електронного голосування кращі показники показала схема BFV, проте схема LTV поступається лише завдяки потенційній можливості існування алгебраїчних атак. Синтез схеми повністю гомоморфного шифрування на основі ДСТУ 8961:2019 дає змогу побудувати надійну систему гомоморфного шифрування, на основі якої можливо побудувати протоколи електронного голосування з високим рівнем анонімності.

5. Для виборців електронної верифікації голосів можливо використовувати схему доказу Фіата – Шаміра з перериваннями. В цьому випадку перевірку валідності голосу можливо звести до перевірки лінійного відношення, для якого генерується доказ валідності з нульовим розголошенням.

Список літератури:

1. Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. In Shafi Goldwasser, editor, ITCS 2012. P. 309–325. ACM, January 2012.

2. Homomorphic Encryption based on Hidden Subspace Membership / Uddipana Dowerah and Srinivasan Krishnaswamy // Indian Institute of Technology Guwahati.

3. Craig Gentry. A fully homomorphic encryption scheme. PhD thesis / Stanford University, 2009.

4. Craig Gentry Amit Sahai Brent Waters Homomorphic Encryption from Learning with Errors: Conceptually-Simpler, Asymptotically-Faster, Attribute-Based.

5. Adriana L´opez-Alt On-the-Fly Multiparty Computation on the Cloud via Multikey Fully Homomorphic Encryption.

6. Jung Hee Cheon1, Andrey Kim1, Miran Kim2, and Yongsoo Song1 Homomorphic Encryption for Arithmetic of Approximate Numbers.

7. Homomorphic Encryption Standardization. [Electronic resource]. Access mode: https://homomorphicencryption.org/.

8. Junfeng Fan and Frederik Vercauteren. Somewhat practical fully homomorphic encryption. Cryptology ePrint Archive, Report 2012/144, 2012. Access mode: http://eprint. iacr.org/2012/144.

9. Marten Van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers // Advances in cryptology–EUROCRYPT 2010. P. 24–43. Springer, 2010.

10. Nigel P Smart and Frederik Vercauteren. Fully homomorphic encryption with relatively small key and ciphertext sizes // International Workshop on Public Key Cryptography. P. 420-443. Springer, 2010.

11. Zvika Brakerski and Vinod Vaikuntanathan. Fully homomorphic encryption from ring-LWE and security for key dependent messages // Advances in Cryptology–CRYPTO 2011. P. 505-524. Springer, 2011.

12. Craig Gentry and Shai Halevi. Implementing gentry's fully-homomorphic encryption scheme // Advances in Cryptology–EUROCRYPT 2011. P. 129–148. Springer, 2011.

13. Zvika Brakerski. Fully homomorphic encryption without modulus switching from classical gapsvp // Annual Cryptology Conference. P. 868-886. Springer, 2012.

14. Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based // Advances in Cryptology–CRYPTO 2013. P. 75-92. Springer, 2013.

15. Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE // SIAM Journal on Computing, 43(2):831-871, 2014.

16. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography // Proceedings of the thirty-seventh annual ACM symposium on Theory of computing. P. 84-93. ACM, 2005.

17. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography // Journal of the ACM (JACM), 56(6):34, 2009.

18. Fully Homomorphic Encryption over the Integers Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan.

19. Danger of using fully homomorphic encryption: A look at Microsoft SEAL Zhiniang Peng Qihoo 360 June 18, 2019.

20. Gentry C. Fully homomorphic encryption using ideal lattices[C] // Stoc. 2009, 9(2009): 169-178.

21. Fan J, Vercauteren F. Somewhat Practical Fully Homomorphic Encryption[J]. IACR Cryptology ePrint Archive, 2012, 2012: 144.

22. Chen H, Huang Z, Laine K, et al. Labeled PSI from Fully Homomorphic Encryption with Malicious Security[C]//Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2018: P.1223-1237.

23. Mursi, Mona & Assassa, Ghazy Moh Rateb & Abdelhafez, Ahmed & Samra, Kareem. (2013). On the Development of Electronic Voting: A Survey // International Journal of Computer Applications. 61. 1-11. 10.5120/10009-4872.

24. Варновский, Н. П. Гомоморфное шифрование / Н. П. Варновский, А. В. Шокуров // Труды ин-та системного программирования РАН. 2006.

25. Hoffstein, Jeff & Pipher, Jill & Schanck, John & Silverman, Joseph & Whyte, William & Zhang, Zhenfei. (2017). Choosing parameters for NTRUEncrypt. 3-18. 10.1007/978-3-319-52153-4_1.

26. Yuanmi Chen Phong BKZ 2.0: Better Lattice Security Estimates / Yuanmi Chen Phong, Q. Nguyen // International Conference on the Theory and Application of Cryptology and Information Security ASIACRYPT 2011: Advances in Cryptology – ASIACRYPT 2011. P. 1-20.

27. Becker A., Ducas L., Gama N., Laarhoven T. (2016). New directions in nearest neighbor searching with applications to lattice sieving // SODA, 2016. P.10-24.

28. Thijs Laarhoven, Michele Mosca, & Joop van de Pol. Finding shortest lattice vectors faster using quantum search. Cryptology ePrint Archive, Report 2014/907, 2014. Access mode: https://eprint.iacr.org/2014/907.

29. Laarhoven T. (2015). Search problems in cryptography: from fingerprinting to lattice sieving (Doctoral dissertation). Eindhoven University ofTechnology. Access mode: http://repository.tue.nl/837539.

30. Lindner R., Peikert C. (2011). Better key sizes (and attacks) for LWE-based encryption // A. Kiayias, CT-RSA~2011. P. 319-339. Springer, Heidelberg.

31. Rachel Player.Parameter selection in lattice-based cryptography. PhD thesis, Royal Holloway, University of London, 2018.

32. Microsoft SEAL. [Electronic resource]. Access mode: https://www.microsoft.com/en-us/research/project/microsoft-seal/.

33. CuHe. [Electronic resource]. Access mode: https://github.com/vernamlab/cuHE.

34. HeLib [Electronic resource]. Access mode: https://github.com/shaih/HElib.

35. Саати Т. Принятие решений. Метод анализа иерархий // Радио и связь, 1993. 278 с.

36. Vadim Lyubashevsky. One-Shot Verifiable Encryption from Lattices / Vadim Lyubashevsky, Gregory Neven // Annual International Conference on the Theory and Applications of Cryptographic Techniques EUROCRYPT 2017: Advances in Cryptology – EUROCRYPT 2017. P. 293-323.

Харківський національний
університет імені В. Н. Каразіна;
АТ «Інститут інформаційних технологій»;
Харківський національний
університет радіоелектроніки

Надійшла до редколегії 09.02.2020