Сучасний стан наукових досліджень та технологій в промисловості. 2018. № 4 (6)
ISSN 2522-9818 (print)
ISSN 2524-2296 (online)

УДК 004.056.55:004.65
DOI: https://doi.org/10.30837/2522-9818.2018.6.005

О. І. Белей

ГОМОМОРФНЕ ШИФРУВАННЯ ДАНИХ У ХМАРНИХ СХОВИЩАХ МЕТОДОМ МАТРИЧНИХ ПОЛІНОМІВ

Предметом дослідження є шифрування інформації в хмарних обчисленнях і сховищах даних. Хмарні технології дозволяють значно скоротити витрати на IT-інфраструктуру і гнучко реагувати на зміни обчислювальних потреб. В такому випадку має бути забезпечено можливість проведення обчислень над зашифрованими даними без їх дешифрування. Таку властивість має повністю гомоморфне шифрування. **Метою** даної статті є підвищення ефективності повністю гомоморфного шифрування (ПГШ) на основі матричних поліномів за допомогою методу пакетного шифрування в один шифротекст декількох відкритих текстів з наступною комплексною обробкою зашифрованих даних. Пакетне шифрування зводиться до того, що при одній операції над двома шифротекстами відбувається одночасне виконання операцій покоординатно над усіма даними, що містяться в цих шифротекстах у вигляді відкритих текстів (SIMD). **Завданнями** визначено побудову алгоритмів повністю гомоморфного шифрування даних за допомогою матричних поліномів. У статті використано **методи** шифрування: з використанням китайської теореми про залишки; шляхом запису в одній матриці декількох різних власних значень при різних власних векторах; за допомогою інтерполяції матричних поліномів. В **результаті** описано та проаналізовано можливі підходи до побудови пакетних ПГШ на підставі матричних поліномів, а також представлено набір алгоритмів, що реалізують криптосхему ПГШ з інтерполяцією матричних поліномів. Наведені алгоритми і криптосхеми дозволяють передавати інформацію в повідомленнях і дані в запитах як відкритий текст, бо над шифрованими даними можна здійснювати необмежену кількість складних алгебраїчних операцій. Це, у свою чергу, ускладнює можливість дешифрування і зчитування даних без знання всього алгоритму. Було показано, що побудовані криптосхеми перевершують аналоги по ефективності, які розроблені дослідниками з IBM. Можна зробити наступний **висновок**: пакетне повністю е шифрування на основі матричних поліномів здатне виключити необхідність хоча б часткового дешифрування даних для несанкціонованих обчислень над зашифрованими масивами даних у хмарних сховищах.

Ключові слова: повністю гомоморфне шифрування; сховище даних; алгоритм; шифротекст; криптографічні методи; криптосхеми; матричні поліноми.

Вступ

Проблемою безпеки особливо часто нехтують щодо вбудованих систем. Як результат, вони можуть бути зламані і згодом використані з метою промислового шпигунства. Злочинці отримують можливість проникнути в корпоративну мережу підприємства, здійснити несанкціонований доступ до інтелектуальної власності та її комерційних таємниць, а також маніпулювати даними підприємства, завдаючи йому шкоди через систему автоматизованого управління. Що стосується користувачів домашніх інтелектуальних пристроїв, то через злом автоматизованих систем управління і несанкціоноване проникнення в системи безпеки потенційним зловмисникам може стати доступна інформація про наявність людей в будинку або, навіть, відкрити для проникнення в будинок двері і вікна. Автомобілі, завдяки наявності систем автономного водіння і бездротового оновлення програмно-апаратних засобів, теж схильні до практично необмеженої вразливості.

Вільний інтернет-простір надто часто використовується зловмисниками для реалізації своїх планів. Соціальні медіа-ресурси, особливо Facebook, зазнали невдачі в захисті прав конфіденційності своїх користувачів і не змогли захистити їх від злочинного впливу.

Зараз здійснюється важливий ключовий етап розвитку соціальних мереж. Виникає питання про те, чи буде регулювання доступу, відповідно до обіцянок Facebook і чи буде усунуто недоліки мережі? Чи дозволить цей новий, удосконалений інструмент спілкування здійснювати безпечну комунікацію, зближуючи світ?

Обидва методи надзвичайно складні. IBM і Microsoft разом з Агентством національної безпеки, багато років працюють над гомоморфним шифруванням (ГШ), проте технологія має значні проблеми з продуктивністю. Але прогрес все ж є. Зокрема, IBM отримала патент на конкретний метод ГШ – підтвердження того, що компанія займається відповідними розробками. Недавно в корпорації оголосили, що бібліотека шифрування ГШ тепер працює в 75 разів швидше.

Інструментарій ГШ допоміг би вберегти інформацію в Facebook від хакерів і незаконного використання, але при цьому дозволяв би витягти ту інформацію, яку вимагають рекламодавці. Гомоморфне шифрування перетворює зашифровані дані в рядки чисел, виконує аналіз цих рядків, а потім дешифрує отримані дані. В результаті виходить та же відповідь, як ніби дані і зовсім не були зашифровані.

Особливо багатообіцяючий поштовх в гомоморфному шифруванні стався в минулому році, коли Google відкрив новий інструмент маркетингових вимірювань. Він використовує цю технологію, щоб рекламодавці могли побачити, чи призводять їх онлайн-оголошення до покупок в магазині. Розшифровка цієї інформації вимагає аналізу масивів даних, що належать окремим підприємствам, незважаючи на те, що ці підприємства зобов'язуються захищати конфіденційність і особисту інформацію власників даних. Цю заборону можна обійти, створюючи агреговані, неспецифічні звіти про порівняння цих масивів даних [6–9].

В експериментальних тестах технологія ГШ допомогла Google успішно проаналізувати

ISSN 2522-9818 (print)
ISSN 2524-2296 (online) Innovative technologies and scientific solutions for industries. 2018. No. 4 (6)

зашифровані дані про те, хто натискає на рекламу. Маючи ці дані, компанія Google змогла надати звіти рекламодавцям, в яких підсумовується взаємозв'язок між двома базами даних, щоб зробити висновок, що 5% людей, які натиснули на рекламне оголошення, купили товари в магазині.

Залишається сподіватися, що компанія Facebook та інші соціальні мережі почнуть активно вивчати та впроваджувати технології гомоморфного шифрування якомога швидше. Також не можна недооцінювати інші потужні заходи, що спрямовані на захист прав конфіденційності. При цьому важливо, щоб всі можливості та інструменти були використані в своїй сукупності, а не окремо, адже це може вплинути життєздатність великих соціальних мереж в майбутньому [13–16].

Використання хмарних обчислень дає багато переваг, але для обробки даних в публічних "хмарах" в загальному випадку необхідно працювати з відкритими даними. Але для роботи з конфіденційними даними необхідна апаратура або хоча б організаційні заходи для зберігання ключів. Дуже рідко провайдери хмарних обчислень дотримуються таких вимог безпеки даних. Це несе в собі ризики, оскільки система не може вплинути жодним чином на те, що відбувається з третьої сторони (сторони зловмисного несанкціонованого доступу). Було б набагато безпечніше передавати дані в зашифрованому вигляді для того, щоб операції, які проводяться над цими даними, жодним чином не поширювало інформацію про ці дані [17, 18].

Гомоморфне шифрування як криптографічний примітив становить інтерес як з прикладної, так і з чисто математичної точки зору. Незважаючи на багаторічні дослідження в цій області, основні проблеми залишаються невирішеними.

Гомоморфне шифрування як криптографічний примітив може знайти широке застосування в криптографії і, в більш широкому сенсі, в розробці математичних методів захисту інформації. Тут, перш-за-все, слід виділити таке, цікаве з прикладної точки зору, завдання як обчислення над зашифрованими даними. Конфіденційні дані зберігаються в зашифрованому вигляді. Для виконання обчислень над ними дані можна розшифрувати, зробивши необхідні операції, і потім результати знову зашифрувати. Але для цього потрібні захищена апаратура організаційні заходи по зберіганню секретних ключів. Обчислення над зашифрованими даними, якщо вони можливі, допомагають уникнути всіх цих проблем.

Аналіз проблеми та існуючих методів

Говорячи про завдання в області інформаційної безпеки, розглянемо один з перспективних напрямків розвитку інформаційних технологій – хмарні обчислення і сховища даних. Дана технологія дозволяє значно зменшити витрати на IT-інфраструктуру і гнучко реагувати на зміни обчислювальних потреб. З іншого боку, подібне зберігання та обробка конфіденційних даних не є безпечною з огляду на можливості неконтрольованого доступу до цих даних провайдера хмарної інфраструктури, а також ризику шкідливих вторгнень в хмару. Для організації безпечної передачі інформації в хмару дані шифруються. Для збільшення продуктивності, зручності та реалізацію безпеки хмарних обчислень необхідно забезпечити можливість здійснення довільних обчислень над зашифрованими даними без їх попереднього розшифрування. Даною властивістю володіє повністю гомоморфне шифрування [12, 17].

Варіант організації безпечних хмарних обчислень полягає в наступному. Є дві сторони – клієнт і сервер. Клієнт генерує ключову пару (sk, pk) з заданим параметром криптостійкості λ. За допомогою секретного ключа sk, який відомий тільки клієнту, проводиться шифрування дешифрування даних. Ключ pk використовується при формуванні запиту клієнтом і відомий серверу. Клієнт виробляє шифрування даних і відправляє їх на сервер, де вони завжди зберігаються в зашифрованому вигляді φ. При виникненні потреби в отриманні даних, що відповідають певним умовам, клієнт посилає серверу запит з функцією f, яку сервер буде обчислювати для зберігання зашифрованих на ньому даних. Сервер, отримавши запит, зчитує зашифровані дані і проводить над ними необхідні обчислення $f(\varphi)$ без попереднього дешифрування. Отриманий результат φ клієнт розшифровує і отримує необхідні йому дані, що задовольняють умовам функції f.

Основною областю застосування повністю гомоморфної криптосистеми є хмарні обчислення та інфраструктура аутсорсингу (рис. 1). Повністю гомоморфне шифрування дозволяє зберігати файли на сервері в зашифрованому вигляді і отримувати від сервера тільки файли, що задовольняють запиту зашифрованому вигляді без необхідності їх розшифрування.

Існує багато протоколів запитів до баз даних [11–15], що включають не тільки конфіденційне отримання записів бази даних, але і конфіденційне отримання індексів певних записів бази даних, які відповідають встановленим умовам. Поряд з протоколами, що використовують повністю гомоморфне шифрування для можливості проведення будь-яких обчислень над збережуваними зашифрованими даними, існують й інші варіанти використання повністю гомоморфного шифрування для вирішення завдань криптографії. Одним із таких завдань є перевірка делегованих обчислень, а також побудова короткого неінтерактивного доведення з нульовим розголошенням [16].

В національній практиці і теоретичних розробках гомоморфному шифруванню приділено мало уваги. Це в основному стосується або часткового гомоморфного шифрування на основі методу заміни числа [19], або – гомоморфного шифрування на основі еліптичних кривих [20, 21].

ISSN 2522-9818 (print)
Сучасний стан наукових досліджень та технологій в промисловості. 2018. № 4 (6) ISSN 2524-2296 (online)

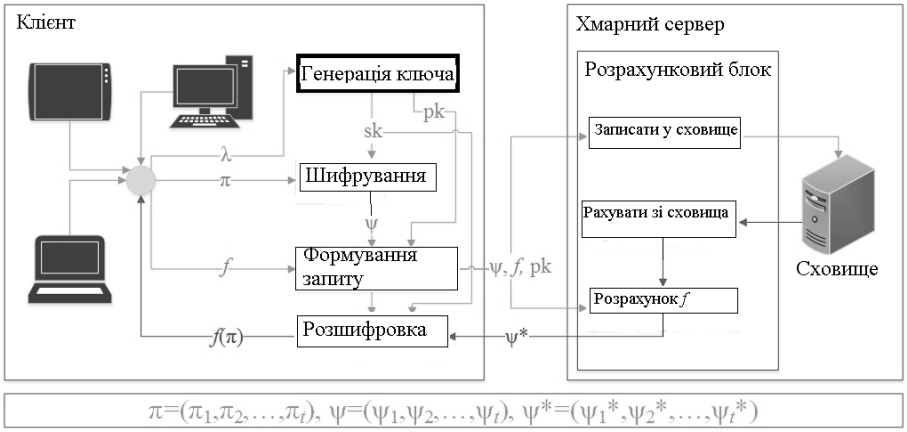

Рис. 1. Алгоритм безпечних хмарних обчислень над зашифрованими даними

Гомоморфне шифрування також знаходить застосування в пошукових системах, де воно використовується для забезпечення "приватного пошуку". В даному випадку пошуковий сервер отримує від користувачів зашифровані запити, які він не може розшифрувати, не знаючи ключа користувача. Результати пошуку також надаються користувачеві в зашифрованому вигляді. Крім того, гомоморфне шифрування використовується в системах електронного голосування, зокрема, при застосуванні підписів наосліп.

Метою даної статті є підвищення ефективності повністю гомоморфного шифрування (ПГШ) на основі матричних поліномів за допомогою методу упаковки в один шифротекст декількох відкритих текстів з подальшою "пакетною" обробкою зашифрованих даних. Даний метод вперше був введений в роботах Джентрі для прискорення роботи його конструкцій [3]. І незважаючи на те, що навіть з використанням цього методу криптосистеми типу Джентрі не стали придатними для практики, їх ефективність за рахунок нього суттєво зросла. Метод "упаковки шифротексту" є дуже перспективним. Оскільки пакетна зводиться до того, що при одній операції над двома шифротекстами відбувається одночасне виконання операцій покоординатно над усіма даними, які містяться в цих шифротекстах з відкритими текстами (організація обробки типу SIMD).

Вирішення завдання

Вперше ідея проведення векторних операцій (SIMD) над зашифрованими даними була запропонована в роботі [16]. Ними також було помічено, що зі застосуванням китайської теореми про залишки, простір відкритих текстів деяких відомих на той час криптосхем ПГШ може бути розширено за рахунок введення векторів, компоненти яких є клітинки для окремих відкритих текстів (plaintext slots). При цьому одне гомоморфне додавання (Add) або множення (Mult) пари шифротекстів неявно додає або множить вектори покомпонентно цілком відкритих текстів.

Кожна клітинка для відкритого тексту призначається для реального зберігання елемента з якогось кінцевого поля $K_n^l = F_p$ (n – кількість елементів кожного шифрованого повідомлення, l – кількість клітинок кінцевого зашифрованого відкритого тексту), і, абстрактно, якщо є два шифротексту, які зберігають зашифровані повідомлення $m_0,...,m_{l-1} \in K_{n1}^l$ і $m_0^{`},...,m_{l-1}^{`} \in K_{n2}^l$ відповідно в клітинках $0,..., l-1$ відкритого тексту. В результаті застосування l-арного додавання до двох шифротекстів виходить в результаті новий шифротекст, який зберігає $m_0 + m_0^{`},...,m_{l-1} + m_{l-1}^{`} \in K_n^l$, а при l-арному множенні - $m_0 \times m_0^{`},...,m_{l-1} \times m_{l-1}^{`} \in K_n^l$. Смарт і Веркотерен використовували це спостереження для створення пакетної (або SIMD) системи гомоморфного шифрування [16].

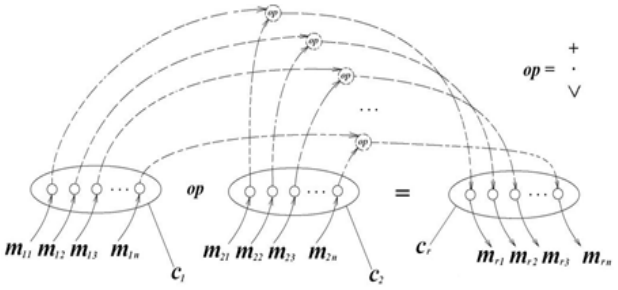

Рис. 2. Виконання векторних операцій над пакетними шифротекстами гомоморфного шифрування

Говорячи про пакетне шифрування і SIMD в криптосистемах зручно говорити про об'єднання (Agregate) відкритих текстів і ключів в просторі. Це пов'язано з тим, що над усіма відкритими даними, які містяться в шифротексті, виконуються паралельно одні й ті ж операції. Тому можна розглядати такі набори відкритих даних як єдині елементи простору масивів відкритих даних.

Ідея пакетного шифрування отримала розвиток і використання в роботах [9-16] завдяки можливості переставляти відкриті тексти всередині одного шифротексту без дешифрування. Це відкриває великі перспективи гомоморфної обробки даних, зокрема уможливлює проведення над зашифрованими

ISSN 2522-9818 (print)
ISSN 2524-2296 (online) Innovative technologies and scientific solutions for industries. 2018. No. 4 (6)

даними в бітовому поданні стандартних машинних операцій: Add, Mult, Xor (додавання, множення, ділення в бітовому поданні). Конкретно, перестановка бітів даних між клітинками (slots) одного шифротексту може бути реалізована по-різному, зокрема, для цієї мети використовується автоморфізм Фробеніуса [4], а в роботі [17] описується використання для цих цілей мереж Бенеша.

Пакетне гомоморфне шифрування має настільки важливе практичне значення, що процедури для його реалізації були включені в нову програмну бібліотеку HELib компанії IBM [10]. Розглянемо нижче структуру цієї повністю гомоморфної криптосхеми, що побудована на використанні булевих матричних поліномів. Відкритими текстами є елементи кільця класів розрахунків Z_p по модулю простого числа p, а секретний ключ складається з матриці $K \in Z_p^{N\times N}$ і вектора $\vec{k} \in Z_p^N$.

Відкритий текст $m \in Z_p$ спочатку кодується в матрицю $M \in Z_p^{N\times N}$, де $M \times \vec{k} = m \times \vec{k}$ і $M \in K$, а потім в матричний поліном $C(X) = R(X) \times (X - K) + M$, де $R(X)$ – випадковий матричний поліном. Після множення двох таких шифротекстів результат наводиться по модулю матричного полінома вигляду $R(X) \times (X - K)$, який називається ключем перешифрування. Семантична криптостійкість такого шифру пов'язана з завданням знаходження коренів булевих матричних поліномів [15].

Якщо сукупність (N, d, p) розглядати як завдання знаходження коренів булевого матричного полінома, то по заданому матричному поліному $F(X)$ ступеня d з коефіцієнтами матричного кільця $Z_p^{N\times N}$ (N – розмірність матриці, p – елемент матриці, просте число) можна відповісти на питання чи є корінь у матричного полінома і як знайти цей корінь. При застосуванні до матричних поліномів концепції SIMD, шифрування може бути реалізовано як мінімум трьома способами: з використанням китайської теореми про залишки; шляхом запису в одній матриці декількох різних власних значень при різних власних векторах; за допомогою інтерполяції матричних поліномів.

Використання китайської теореми про залишки є найбільш перспективним шляхом, однак, вона вимагає застосування великої алгебраїчної методики [9]. Використання декількох власних числових матриць є простим, але не дуже ефективним шляхом. Розглянемо далі реалізацію пакетного шифрування за допомогою інтерполяції матричних поліномів.

Інтерполяція матричних поліномів зводиться до наступного. Для заданих m пар матриць $(X_i, Y_i), i-1,...,m$ існує матричний поліном $A(X) = A_m X^m + A_{m-1} X^{m-1} + ... + A_1 X + A_0$ з вектором $A(X_i), i-1,...,m$ у випадку, коли блочно-матрична система лінійних рівнянь

$$(A_1, A_2, ..., A_m) \times \begin{pmatrix} I_1 & I_2 & ... & I_m \\ X_1 & X_2 & ... & X_m \\ ... & ... & ... & ... \\ X_1^{m-1} & X_2^{m-2} & ... & \end{pmatrix} = (Y_1, Y_2, ..., Y_m) \qquad (1)$$

має розв'язок.

В описаній далі криптосхемі складовим простором відкритих текстів є Z_p^l (простір l-мірних векторів з елементами p), простором шифротексту є кільце матричних поліномів $Z_p^{N\times N}[X]$, складовим простором ключів є вектор пар "матриця $Z_p^{N\times N}$ і вектор Z_p^N" разом з деякою оберненою матрицею $Z_p^{N\times N}$. Опишемо нижче алгоритми криптосхеми.

Алгоритм 1. Для генерації ключа (KeyGen) використовуємо наступні вхідні дані: параметр рівня криптостійкості λ, модуль простору відкритих текстів p, кількість клітинок l. Результатом буде секретний ключ *sk*, ключ перешифрування *rk*. Порядок виконання алгоритму 1 наступний:

1. Визначити $N \leftarrow \lambda, d \leftarrow \omega(\lambda)$.

2. Вибрати довільну перетворювану матрицю $K_0 \in GL(N, Z_p)$.

3. Вибрати l матриць $K_i, i = 1,...,l$, таких, що $K_i \neq K_i^2 \neq ... \neq K_i^{l-1}, I \in S(K_i)$.

4. Для кожної матриці K_i необхідно вибрати випадковий власний вектор k_i.

5. $sk \leftarrow \{(K_i, k_i), i = 1,...,l\}, K_0$.

6. Згенерувати випадковий приведений матричний поліном $\hat{R}(X), \deg(\hat{R}(X)) \le d$.

7. Згенерувати приведений матричний поліном $S(X), \deg(S(X)) = l + 1$ такий, що $S(K_i) = 0, i = 1,...,l$ (тобто всі K_i – його корені).

8. $R(X) \leftarrow \hat{R}(X) - S(X)$.

9. $rk \leftarrow R(X - K_0)$.

Матриці, які задовольняють умові $K_i \neq K_i^2 \neq ... \neq K_i^{p-1}$ потрібні для ефективної генерації матриці з $Comm(K_i)$, бо відомо, що лінійні комбінації ступенів матриці гарантовано знаходяться в її комутанті. В найбільш важливому випадку $p = 2$ ця умова зводиться до $K_i \neq K_i^2$, а такі матриці будуть неідемпотентними.

Наявність одиниці в масиві матриці разом з попередньою умовою є гарантією можливості вибору в $Comm(K_i)$ нетривіальних матриць з довільними власними числами.

Запис $\omega(\lambda)$ означає деяку функцію, лінійну від λ (тобто $\omega(\lambda) = O(\lambda)$), а її конкретизація істотна для

ISSN 2522-9818 (print)
Сучасний стан наукових досліджень та технологій в промисловості. 2018. № 4 (6) ISSN 2524-2296 (online)

аналізу криптостійкості, але несуттєва для аналізу асимптотичної складності розрахунків над шифротекстами.

Алгоритм 2. Для шифрування даних (Encrypt) використовуються наступні вхідні дані: вектор-повідомлення відкритих текстів $m_i, i = 1,...,l$, секретний ключ *sk*. Результатом буде матричний поліном шифротексту. Порядок виконання алгоритму визначено наступний:

1. Для кожного $m_i, i = 1,...,l$ вибрати таку випадкову матрицю M_i, де m_i буде її власним числом при власному векторі $\vec{k}_i$.

2. З допомогою алгоритму інтерполяції матричних багаточленів розрахувати такий $\hat{C}(X)$, в якому $deg(\hat{C}(X)) \le N + \omega(\lambda), \hat{C}(K_i) = M_i$.

3. Розрахувати $C(X) = \hat{C}(X - K_0)$.

4. Повернути як результат $C(X)$.

Алгоритм 3. Для дешифрування (Decrypt) використовуємо наступні вхідні дані: матричний поліном шифротексту $C(X)$, секретний ключ *sk*. В результаті отримаємо повідомлення відкритого тексту $m \in Z_p$. Порядок виконання запропонованого алгоритму буде наступний:

1. Розраховуємо $\hat{C}(X) = C(X - K_0^{-1})$.

2. Для кожного $i = 1,...,l$ виконуємо $M_i \leftarrow \hat{C}(K_i)$.

3. Для ненульової координати $(k_j^{-1})_i$ вектора k_i знаходимо $m_i = \left(k_j^{-1}\right)_i \left(M_i \times \vec{k}_i\right)$.

4. Повертаємо результат як $(m_1,...,m_l)$.

Запропонована криптосхема підтримує як адитивний, так і мультиплікативний гомоморфізм. Після множення двох шифротекстів для зниження ступеня результат потрібно приводити по модулю ключа перешифрування – матричного полінома.

При умові S – корінь матричного полінома $M(\lambda)$ коректність дешифрування базується на:

$$M(\lambda) = Q(\lambda) \times (I(\lambda) - S) \qquad (2)$$

де $Q(\lambda)$ – матричний поліном степені $m-1$, $I(\lambda)$ – приведений матричний поліном, λ – параметр рівня криптостійкості.

Твердження (2) справедливе як для матричних поліномів над комплексними числами, так і для матричних поліномів над кінцевими полями (вимога алгебраїчної замкненості поля не використовується).

Дешифрування вище описаної криптосхеми є коректним і гомоморфічним для всіх арифметичних схем, що складаються з операцій додавання та множення по модулю p. У нашому випадку підстановка полінома вказаного виду є ізоморфізмом кілець.

Вище описана криптосхема є компактною, бо в процесі проведення розрахунків над шифротекстами степінь матричних поліномів результату не перевищує заданий.

Найбільш значущою характеристикою ефективності ПГШ є аналіз складності алгоритму множення двох шифротекстів. Асимптотична складність цієї операції становить $\approx O(\lambda^{3,76\rightarrow})$. Важливе питання криптостійкості буде розглядатися далі, проте, потрібно сказати, що при виконанні криптоаналізу видно, як вище описана криптосхема може мати досить високу криптостійкість.

Запропонована Джентрі схема повністю гомоморфного шифрування може бути детально розглянута при розрахунках в Z_2, що є аналогом роботи з бітами.

Для початку вибирається будь-непарне число $p = 2k+1$, яке є секретним параметром. Нехай m приймає значення $\{0,1\}$. Побудуємо число $z \in Z_2$ за правилом $z = 2r + m$, де r – довільне число. Це означає, що $z = m(mod 2)$.

У розглядуваному випадку шифрування зводиться до того, що будь-якому m ставиться у відповідність число $c = z + pq$, де q – довільне число. Звідси випливає, що $c = 2r + m + (2k+1) * q$, а $c mod 2 = (m+q) mod 2$, відповідно. Злочинцем може бути визначена лишень парна частина виходу шифрування.

Алгоритм дешифрування зводиться до наступного. Припустимо є відоме зашифроване число C і відомий секрет P. Тоді процес дешифрування передбачає наступні дії:

1. Безпосереднє дешифрування з допомогою секретного параметра P: $r = c mod p = (z + pq) mod p = z mod p + (pq) mod p$, де $r = c mod p$ називається шумом і $z \in \left(-\frac{p}{2}, \frac{p}{2}\right)$.

2. Отримання вихідного зашифрованого біта: $m = r mod 2$.

Перевіримо, чи дійсно дана операція є гомоморфною. Візьмемо два біта $m_1, m_2 \in Z_2$ і поставимо їм у відповідність пару чисел $z_1 = 2r_1 + m_1$ і $z_2 = 2r_2 + m_2$. Беремо секретний параметр $p = 2k+1$ і виробляємо шифрування: $c_1 = z_1 + pq_1$ і $c_2 = z_2 + pq_2$. Сума цих чисел дорівнює:

$$c_1 + c_2 = z_1 + pq_1 + z_2 + pq_2 = z_1 + z_2 + p(q_1 + q_2) = 2r_1 + m_1 + 2r_2 + m_2 + (2k+1)(q_1 + q_2) \qquad (3)$$

Застосування операції дешифрування до цієї суми дає в результаті суму вихідних біт m_1, m_2:

$$((c_1 + c_2) \bmod p) \bmod 2 = (2(r_1 + r_2) + m_1 + m_2) \bmod 2 = m_1 + m_2.$$

Проте, не знаючи p, розшифрувати дані неможливо:

ISSN 2522-9818 (print)
ISSN 2524-2296 (online) Innovative technologies and scientific solutions for industries. 2018. No. 4 (6)

$$((c_1 + c_2) \bmod p) \bmod 2 = m_1 + m_2 + q_1 + q_2 .$$

Аналогічні дії можна провести з операцією множення:

$$c_1 c_2 = (z_1 + pq_1)(z_2 + pq_2) = z_1 z_2 + p(z_1 q_1 + z_2 q_2) + p^2 q_1 q_2 = (2r_1 + m_1)(2r_2 + m_2) + (2k+1)((2r_1 + m_1)q_2 + (2r_2 + m_2)q_1) = 4r_1 r_2 + (r_1 m_2 + r_2 m_1) + m_1 m_2 + 2k(2r_1 q_2 + m_1 q_2 + 2r_2 q_1 + m_2 q_1) + 2r_1 q_2 + m_1 q_2 + 2r_2 q_1 + m_2 q_1. \quad (4)$$

Застосовуючи до отриманого результату процедуру розшифрування отримаємо:

$$((c_1 c_2) \bmod p) \bmod 2 = (4r_1 r_2 + 2(r_1 m_2 + r_2 m_1) + m_1 m_2) \bmod 2 = m_1 m_2 .$$

Отже, розглянута вище схема дійсно є повністю гомоморфним шифруванням. Однак, незважаючи на отримані результати, дана схема має суттєві обмеження в практичному застосуванні, тому що в результаті подібних обчислень дуже швидко накопичується помилка *r*. Після того, як значення цієї помилки перевищить значення *p*, правильно розшифрувати повідомлення не вдасться. Для подолання цієї проблеми Джентрі був запропонований механізм самокорекції шифротексту. Однак такий спосіб призводить до стрімкого зростання обсягу шифротексту і також є непрактичним. Єдиним вирішенням цієї проблеми є обмеження кількості операцій над даними або розробка більш складних алгоритмів обчислення.

Розглянемо іншу гомоморфну криптосхему, яка побудована на основі матричних поліномів і має малі обчислювальні витрати та підтримує можливість розпаралелення. В даному випадку шифротексти є поліномами, коефіцієнтами яких є матриці (так звані матричні поліноми). Ідея побудови повністю гомоморфної криптосистеми полягає в наступному.

Секретним ключем є матричний поліном $K(X)$ і вектор $\vec{k}_i$. Нехай m_1 *і* m_2 – відкриті тексти, а відповідними для них шифротекстами будуть матричні поліноми $C_1(X) = R_1(X)K(X) + M_1$ і $C_2(X) = R_2(X)K(X) + M_2$, де $M_1\vec{k} = m_1\vec{k}$ і $M_2\vec{k} = m_2\vec{k}$. Щоб розшифрувати повідомлення, необхідно взяти залишок від ділення шифротексту на $K(X)$ і отримати з отриманої матриці відкритий текст за допомогою $\vec{k}_i$.

Нехай $Z_p^{N\times N}$ – кільце $N \times N$ матриць з елементами кільця цілих чисел Z_p. $F = \{A_0, A_1, A_2, ...\}, A_i \in Z_p^{N\times N}$ – множина послідовних матриць з $Z_p^{N\times N}$, причому всі, крім кінцевого їх числа, рівні нульовій матриці. Позначимо множину всіх таких послідовностей $Z_p^{N\times N}[X]$. У разі, якщо $F, G \in Z_p^{N\times N}[X]$, $G = \{B_0, B_1, B_2, ...\}, B_i \in Z_p^{N\times N}$, виконуються наступні операції:

$$F + G = \{A_0 + B_0, A_1 + B_1, A_2 + B_2, ...\},$$
$$FG = \{A_0B_0, A_0B_1 + A_1B_0, A_0B_2 + A_1B_1 + A_2B_0, ...\} = \{C_k\} \quad (5)$$
$$C_k = \sum_{i+j=k} A_i B_j, k = 0,1,2,...$$

При таких введених операціях додавання і множення множина $Z_p^{N\times N}$ є кільцем, елементами якого є матричні поліноми. Опис алгоритмів генерації ключів *KeyGen*, шифрування *Enc* і розшифрування *Dec*, а також реалізація гомоморфності обчислень наведено нижче. Тут ω, δ, φ – заздалегідь визначаються як константи.

Генерація ключа шифрування *sk* здійснюється наступним чином:

1. Генеруємо приведений поліном $K(X) \in Z_p^{N\times N}[X]$, тобто його коефіцієнти $K_i, i = 0, ..., \deg(K(X))^{-1}$ вибираємо з $Z_p^{N\times N}$ згідно рівномірного розподілу.

2. Генеруємо вектор $\vec{k}_i \in Z_p^N$ (де k_i вибираємо з Z_p згідно рівномірного закону) такий, щоб хоч одна координата перетворювалася в Z_p.

3. На виході алгоритму отримуємо секретний ключ $sk \leftarrow \{K(X), \vec{k}\}$.

При генерації секретного перешифратора *pk* порядок дій буде наступним:

1. Генеруємо матричний поліном $R`(X) \in Z_p^{N\times N}[X]$, тобто $\deg(R`(X)) \le \delta\lambda, R_i`, i = 0, ..., \deg(R`(X))^{-1}$ вибираємо згідно рівномірного розподілу.

2. Тоді відкритий ключ $pk \leftarrow \{R`(X), K(X)\}$.

В розглядуваному випадку алгоритм шифрування можна буде виконати наступним чином:

1. У відповідності до відкритого тексту $m \in Z_p$ ставиться випадкова матриця $M \in Z_p^{N\times N}$ $(M\vec{k} = m\vec{k})$ і $MK(X) = K(X)M$, тобто вона має власний вектор $\vec{k}$ при власному значенні m і комутує з матричним поліномом $K(X)$.

2. Генеруємо $R(X) \in Z_p^{N\times N}[X]$, $\deg(R(X)) \le \omega\lambda$ і вибираємо $\deg(R(X)) \le \deg(R`(X) - \deg(K(X))), R_i, i = 0, ..., \deg(R(X))$ з $Z_p^{N\times N}$ згідно рівномірного розподілу.

3. Шифротекст розраховуємо відповідно до формули $C(X) = R(X)K(X) + M$.

В розглядуваному випадку алгоритм дешифрування повідомлень буде виглядати наступним чином:

1. Розраховуємо $M = C(X) \bmod K(X)$.

2. Для зворотних координат k_i розраховуємо $m = k_i^{-1}(M\vec{k})$.

ISSN 2522-9818 (print)
Сучасний стан наукових досліджень та технологій в промисловості. 2018. № 4 (6) ISSN 2524-2296 (online)

Гомоморфні розрахунки при шифруванні матричним поліном наведено нижче. Для співставлення полінома $f^{\perp}(X_1,...,X_t)$ в шифротекстах $C_1,...,C_t$ з поліномом $f(x_1,...,x_t)$ над $m_1,...,m_t \in Z_p$ необхідно замінити операції з Z_p на операції додавання та множення поліномів в $Z_p^{N\times N}$.

Для шифротекстів $c \leftarrow R(X)K(X)+M$ операція дешифрування дає $m = k_i^{-1}((C(x) \bmod K(x)\vec{k})$. Тоді, $c_1 + c_2 = (R_1(X)+R_2(X)K(X)+M_1+M_2$, які є вірним шифротекстом для $(m_1+m_2) \bmod 2$, тобто $(M_1+M_2)\vec{k} = (m_1+m_2)\vec{k}$.

У випадку множення отримуємо:

$$c_1c_2 = R_1(X)K(X)R_2(X)K(X)+R_1(X)K(X)M_2+R_2(X)K(X)M_1+M_1M_2 = \\ = R_1(X)K(X)R_2(X)+R_1(X)M_2+R_2(X)M_1)K(X)+M_1M_2 \quad (6)$$

що є результатом дешифрування $(m_1m_2) \bmod 2$, тобто $(M_1M_2)\vec{k} = M_1(M_2\vec{k}) = M_1(m_2\vec{k}) = m_2(M_1\vec{k}) = m_1m_2\vec{k}$.

Описана вище криптосхема є коректною і компактною, а значить реалізує повністю гомоморфне шифрування. Функція шифрування $E(k,m)$, де m – відкритий текст, k – ключ шифрування, є гомоморфною щодо операції множення над відкритими текстами, якщо існує ефективний алгоритм M, який отримавши на вхід будь-яку пару криптограм виду $E(k,m_1), E(k,m_2)$, видає криптограму C таку, що при дешифрування C буде отримано відкритий текст $m_1 \times m_2$. На відміну від частково гомоморфного шифрування, яке дозволяє здійснювати гомоморфні обчислення тільки над однією операцією відкритого тексту, повністю гомоморфне шифрування забезпечує гомоморфізм обох операцій (як додавання, так і множення):

$$\begin{cases} Dec\left(Enc(m_1) \otimes Enc(m_2)\right) = m_1 \times m_2 \\ Dec\left(Enc(m_1) \oplus Enc(m_2)\right) = m_1 + m_2 \end{cases}. \quad (7)$$

Вище запропоновані алгоритми та криптосхеми реалізовані та протестовані на мові C#. Здійснено порівняння швидкості шифрування даних і швидкості операцій в рамках властивостей гомоморфності. Для кожної криптосистеми були реалізовані виділені окремі класи, які містили такі функції: генерація ключів (KeyGeneration); шифрування (Encrypt); дешифрування (Decrypt).

В ході експерименту були отримані наступні результати:

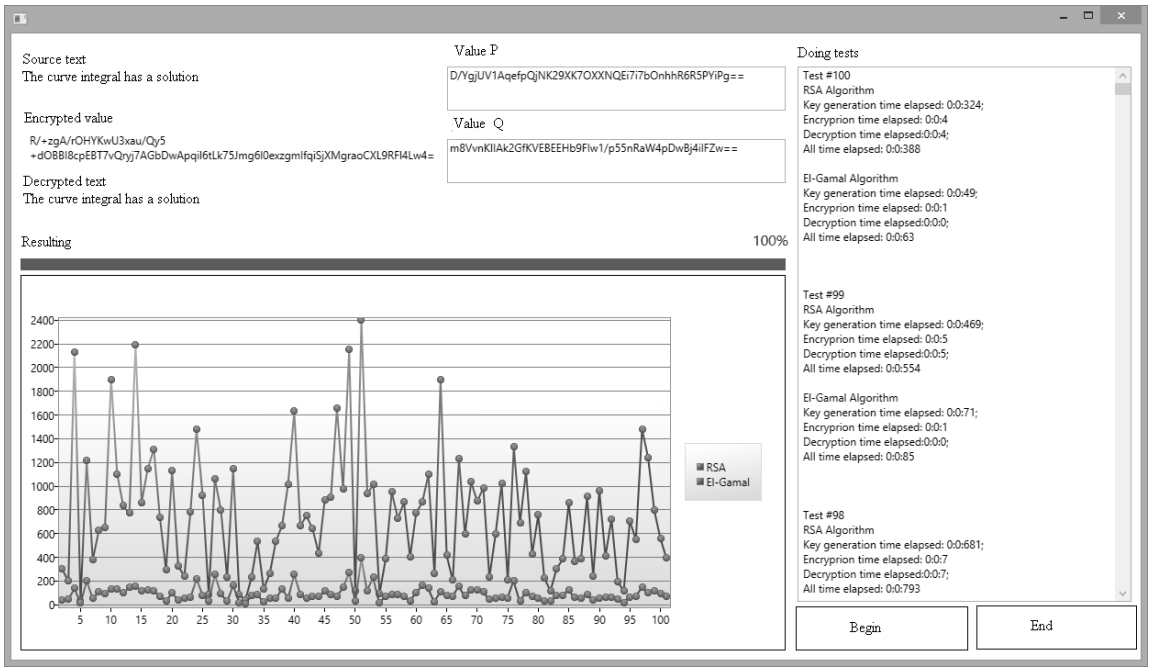

Рис. 3. Тестування процесів генерації ключів, шифрування та дешифрування тексту з використанням криптосистем RSA та Ель-Гамаля

З наведених результатів можна вивести певні дані по продуктивності застосування криптосистем RSA та Ель-Гамаля на основі описаних вище алгоритмів повного гомоморфного шифрування з використанням матричних поліномів:

Таблиця 1. *Продуктивність криптосистем RSA та Ель-Гамаля на основі алгоритмів повного гомоморфного шифрування*

Крипто-система	Генерація ключа	Шифру-вання	Дешифру-вання
RSA	0:0:212 мс	0:0:5 мс	0:0:4 мс
Ель Гамаля	0:0:143 мс	0:0:2 мс	0:0:1 мс

ISSN 2522-9818 (print)
ISSN 2524-2296 (online) Innovative technologies and scientific solutions for industries. 2018. No. 4 (6)

Проведене тестування криптосистем (рис. 4) підтвердило ефективність використання даного підходу для проведення безпечних обчислень. Вибір досить великої, порівняно зі значеннями використовуваних операндів, модуля дозволяє коректно вирішувати проблему однозначності обчисленні. Операції віднімання і ділення на ціле число, наявність яких необхідна для здійснення повноцінних обчислень, можуть бути реалізовані через операції додавання і множення на зворотне число (в цьому випадку модуль групи повинен бути простим числом).

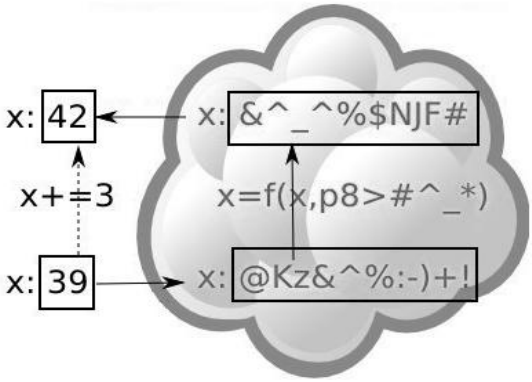

Рис. 4. Повністю гомоморфне шифрування в хмарних базах даних без знання вмістимого запиту повідомлення

Гомоморфне шифрування є потужним апаратом для захисту збережуваних у хмарних сховищах даних. І лише повністю гомоморфне шифрування здатне виключити необхідність хоча б часткового дешифрування даних для здійснення обчислень над ними. Втім, і воно не буде здатне витіснити будь-які інші види криптографічного захисту, оскільки будь-яке подібне шифрування принципово вразливе до атаки з підібраним текстом.

Висновки

В дослідженні описано та проаналізовано можливі підходи до побудови пакетного ПГШ на основі матричних поліномів, а також представлено набір алгоритмів, який реалізує один з цих підходів – криптосхему ПГШ з інтерполяцією матричних поліномів. Було показано, що по ефективності побудована криптосхема перевершує аналоги, розроблені дослідниками з IBM. Опис програмної реалізації криптосхеми з обґрунтуванням криптостійкості відповідних алгоритмів буде розглянуто в подальших дослідженнях.

Розглянуто гомоморфне шифрування, як найбільш перспективний напрямок в області захисту інформації при використанні хмарних обчислень. Здійснене тестування продуктивності криптосистем RSA, Пейе та Ель-Гамаля, кожна з яких володіє певною гомоморфною властивістю. В ході експериментів з'ясувалося, що на даний момент криптосистема Гентрі не завжди може застосовуватися для шифрування передачі повідомлень та запитів до сховищ даних. У той же час, криптосистеми RSA і Пейе можуть використовуватися в окремих випадках для шифрування даних у сховищі MS Azure чи Amazon.

Список літератури

1. Albrecht M. R., Farshim P., Faugere J. C., Perret L. Polly cracker, revisited. Advances in Cryptology. Springer Berlin Heidelberg. 2011. P. 179–196.
2. Armknecht F., Augot D., Perret L., Sadeghi A. R. On constructing homomorphic encryption schemes from coding theory. *Cryptography and Coding.* Springer Berlin Heidelberg. 2011. P. 23–40.
3. Boneh D., Gentry C., Halevi S., Wang F., Wu D. J. Private database queries using somewhat homomorphic encryption. *Applied Cryptography and Network Security.* Springer Berlin Heidelberg. 2013. P. 102–118. DOI: https://doi.org/ 10.1007/978-3-642-38980-1_7.
4. Cheon J. H., Coron J. S., Kim J., Lee M. S., Lepoint T., Tibouchi M., Yun A. Batch Fully Homomorphic Encryption over the Integers. *Advances in Cryptology. EUROCRYPT*. Vol. 7881. 2013. P. 315–335. DOI: https://doi.org/ 10.1007/978-3-642-38348-9_20.
5. Dennis, Jr J. E., Traub J. F., Weber R. P. Algorithms for solvents of matrix polynomials. *SIAM Journal on Numerical Analysis*. 1978. Vol. 15. No. 3. P. 523–533.
6. Domingo-Ferrer J. A Provably Secure Additive and Multiplicative Privacy Homomorphism. *Information Security.* Springer Berlin Heidelberg. 2002. P. 471–483.
7. Gavin G. An efficient FHE based on the hardness of solving systems of non-linear multivariate equations. *IACR Cryptology ePrint Archive*. 2013. No. 262.
8. Gentry S., Halevi N. P. Smart. Fully homomorphic encryption with polylog overhead. *Advances in Cryptology – EUROCRYPT Springer Berlin Heidelberg*. 2012. P. 465–482. DOI: https://doi.org/ 10.1007/978-3-642-29011-4_28.
9. Guellier Antoine. Can Homomorphic Cryptography ensure Privacy? *[Research Report] RR-8568*. 2014. P. 111. URL : https://hal.inria.fr/hal-01052509v1.
10. Halevi S., Shoup V. Algorithms in HElib. *IACR Cryptology ePrint Archive*. 2014. No. 106.
11. Herold G. Polly cracker, revisited, revisited. Public Key Cryptography. *PKC*. Springer Berlin Heidelberg. 2012. P. 17–33.
12. Hojsík M., Půlpánová V. A fully homomorphic cryptosystem with approximate perfect secrecy. Proceedings of the 13th international conference on Topics in Cryptology. Springer-Verlag. 2013. P. 375–388. DOI: https://doi.org/10.1007/978-3-642-36095-4_24.
13. Naehrig M., Lauter K., Vaikuntanathan V. Can homomorphic encryption be practical? Proceedings of the 3rd ACM workshop on Cloud computing security workshop. ACM. 2011. P. 113–124. DOI: https://doi.org/10.1145/2046660.2046682.
14. Poteya Manish M., Dhoteb C. A., Sharmac Deepak H. Homomorphic Encryption for Security of Cloud Data. *Procedia Computer Science 79*. 2016. P. 175 – 181. DOI: https://doi.org/10.1016/j.procs.2016.03.023.

ISSN 2522-9818 (print)
Сучасний стан наукових досліджень та технологій в промисловості. 2018. № 4 (6) ISSN 2524-2296 (online)

15. Rivest R. L., Adleman L., Dertouzos M. L. On data banks and privacy homomorphisms. *Foundations of secure computation*. 1978. Vol. 4. No. 11. P. 169–180.
16. Silverberg. Fully homomorphic encryption for mathematicians. Women in Numbers 2: Research Directions in Number Theory. 2013. T. 606. P. 111.
17. Smart Nigel P., Vercauteren F. Fully Homomorphic Encryption with Relatively Small Key and Ciphertext Sizes. Public Key Cryptography-PKC 2010: 13th International Conference on Practice and Theory in Public Key Cryptography, Paris, France, May 26-28, 2010, Proceedings. Springer, 2010. P. 420.
18. Wagner D. Cryptanalysis of an algebraic privacy homomorphism. Proc. of 6th Information Security Conference (ISC'03). 2003. DOI: https://doi.org/10.1.1.5.1420.
19. Yasuda M., Shimoyama T., Kogure J., Yokoyama K., Koshiba T. Packed homomorphic encryption based on ideal lattices and its application to biometrics. *Security Engineering and Intelligence Informatics*. Springer Berlin Heidelberg. 2013. P. 55–74.
20. Ступень П. В., Соколов С. О., Золкіна О. Ю. Застосування гомоморфного шифрування для захисту числових даних у хмарних сховищах. *Наукові праці Чорноморського державного університету імені Петра Могили комплексу "Києво-Могилянська академія". Серія: Комп'ютерні технології*. 2015. Т. 266, Вип. 254. С. 71–75. URL : http://nbuv.gov.ua/UJRN/Npchduct_2015_266_254_13 (дата звернення : 28.11.2018).
21. Квєтний Р. Н., Титарчук Є. О. Використання частково гомоморфного алгоритму шифрування на еліптичних кривих у хмарній системі електронного голосування. *Оптико-електронні інформаційноенергетичні технології*. 2016. № 32 (2). С. 14–22.
22. Квєтний Р. Н., Титарчук Є. О. Аналіз криптостійкості частково гомоморфного алгоритму шифрування на основі еліптичних кривих. *Інформаційні технології та комп'ютерна інженерія*. 2017. № 1 (38). С. 83–87.

References

1. Albrecht, M. R., Farshim, P., Faugere, J. C., Perret, L. (2011), "Polly cracker, revisited. Advances in Cryptology", Springer Berlin Heidelberg, P. 179-196.
2. Armknecht, F., Augot, D., Perret, L., Sadeghi, A. R. (2011) "On constructing homomorphic encryption schemes from coding theory", *Cryptography and Coding*, Springer Berlin Heidelberg, P. 23-40.
3. Boneh, D., Gentry, C., Halevi, S., Wang, F., Wu, D. J. (2013), "Private database queries using somewhat homomorphic encryption", *Applied Cryptography and Network Security*. Springer Berlin Heidelberg, P. 102–118. DOI: https://doi.org/10.1007/978-3-642-38980-1_7.
4. Cheon, J. H., Coron, J. S., Kim, J., Lee, M. S., Lepoint, T., Tibouchi, M., Yun, A. (2013), "Batch Fully Homomorphic Encryption over the Integers", *Advances in Cryptology, EUROCRYPT*, Vol. 7881, P. 315–335. DOI: https://doi.org/ 10.1007/978-3-642-38348-9_20.
5. Dennis, Jr J. E., Traub, J. F., Weber, R. P. (1978), "Algorithms for solvents of matrix polynomials", *SIAM Journal on Numerical Analysis*, Vol. 15, No. 3, P. 523–533.
6. Domingo-Ferrer, J. (2002), "A Provably Secure Additive and Multiplicative Privacy Homomorphism", *Information Security*, Springer Berlin Heidelberg, P. 471–483.
7. Gavin, G. (2013), "An efficient FHE based on the hardness of solving systems of non-linear multivariate equations", *IACR Cryptology ePrint Archive*, No. 262.
8. Gentry, S., Halevi, N. P. Smart (2012), "Fully homomorphic encryption with polylog overhead" Advances in Cryptology, EUROCRYPT, Springer Berlin Heidelberg, P. 465-482. DOI: https://doi.org/ 10.1007/978-3-642-29011-4_28.
9. Guellier, Antoine (2014), "Can Homomorphic Cryptography ensure Privacy?" *[Research Report], RR-8568*, P. 111, available at : URL : https://hal.inria.fr/hal-01052509v1 (last accessed 11.11.2018).
10. Halevi, S., Shoup, V. (2014), "Algorithms in HElib", *IACR Cryptology ePrint Archive*, No. 106.
11. Herold, G. (2012), "Polly cracker, revisited, revisited. Public Key Cryptography", *PKC*, Springer Berlin Heidelberg, P. 17–33.
12. Hojsík, M., Půlpánová, V. (2013), "A fully homomorphic cryptosystem with approximate perfect secrecy", *Proceedings of the 13th international conference on Topics in Cryptology, Springer-Verlag*, P. 375–388. DOI: https://doi.org/10.1007/978-3-642-36095-4_24.
13. Naehrig, M., Lauter, K., Vaikuntanathan, V. (2011), "Can homomorphic encryption be practical?", *Proceedings of the 3rd ACM workshop on Cloud computing security workshop, ACM*, P. 113–124. DOI: https://doi.org/10.1145/2046660.2046682.
14. Poteya, Manish, M., Dhoteb, C. A., Sharmac Deepak H. (2016), "Homomorphic Encryption for Security of Cloud Data", *Procedia Computer Science 79*, P. 175–181. DOI: https://doi.org/10.1016/j.procs.2016.03.023.
15. Rivest, R. L., Adleman, L., Dertouzos, M. L. (1978), "On data banks and privacy homomorphisms", *Foundations of secure computation*, Vol. 4, No. 11, P. 169–180.
16. Silverberg (2013), "Fully homomorphic encryption for mathematicians", *Women in Numbers 2: Research Directions in Number Theory*, Vol. 606, P. 111.
17. Smart, Nigel, P., Vercauteren, F. (2010), "Fully Homomorphic Encryption with Relatively Small Key and Ciphertext Sizes", *Public Key Cryptography-PKC 2010: 13th International Conference on Practice and Theory in Public Key Cryptography, Paris, France*, Proceedings, Springer, P. 420.
18. Wagner, D. (2003), "Cryptanalysis of an algebraic privacy homomorphism", *Proc. of 6th Information Security Conference (ISC'03)*. DOI: https://doi.org/10.1.1.5.1420.
19. Yasuda, M., Shimoyama, T., Kogure, J., Yokoyama, K., Koshiba, T. (2013), "Packed homomorphic encryption based on ideal lattices and its application to biometrics", *Security Engineering and Intelligence Informatics*, Springer Berlin Heidelberg, P. 55–74.
20. Stupen, P. V., Sokolov, S. O., Zolkina, O. Yu. (2015), "Application of homomorphic encryption for the protection of numerical data in cloud storage", *Scientific works of the Petro Mohyla Black Sea State University of the Kyiv-Mohyla Academy complex. Series: Computer Technology*, Vol. 266, No. 254, P. 71–75, available at : http://nbuv.gov.ua/UJRN/Npchduct_2015_266_254_13 (last accessed: 28.11.2018).
21. Kvyetnyy, R. N., Tytarchuk, Ye. O. (2016), "The use of a partially homomorphic encryption algorithm on elliptic curves in a cloud-based electronic voting system", *Optoelectronic information technology technologies*, No. 32 (2), P. 14–22.

ISSN 2522-9818 (print)
ISSN 2524-2296 (online) Innovative technologies and scientific solutions for industries. 2018. No. 4 (6)

22. Kvyetnyy, R. N., Tytarchuk, Ye. O. (2017), "Analysis of cryptostability of partially homomorphic encryption algorithm on the basis of elliptic curves", *Information Technology and Computer Engineering*, No. 1 (38), P. 83–87.

Надійшла (Received) 04.12.2018

Відомості про авторів / Сведения об авторах / About the Authors

Белей Олександр Ігорович – кандидат економічних наук, доцент, Національний університет "Львівська політехніка", доцент кафедри систем автоматизованого проектування, Львів, Україна; e-mail: tiger_oles@i.ua; ORCID ID: https://orcid.org/0000-0003-4150-7425.

Белей Александр Игоревич – кандидат экономических наук, доцент, Национальный университет "Львовская политехника", доцент кафедры систем автоматизированного проектирования, Львов, Украина.

Belej Oleksandr – PhD (Economics Sciences), Associate Professor, Lviv Polytechnic National University, Associate Professor at the Department of Computer-Aided Systems, Lviv, Ukraine.

ГОМОМОРФНОЕ ШИФРОВАНИЕ ДАННЫХ В ОБЛАЧНОМ ХРАНИЛИЩЕ МЕТОДОМ МАТРИЧНЫХ ПОЛИНОМОВ

Предметом исследования является шифрование информации в облачных вычислениях и хранилищах данных. Облачные технологии позволяют значительно сократить расходы на ИТ-инфраструктуру и гибко реагировать на изменения вычислительных потребностей. В таком случае должно быть обеспечено возможности проведения вычислений над зашифрованными данными без их дешифровки. Таким свойством обладает полностью гомоморфного шифрования. **Целью** данной статьи является повышение эффективности полностью гомоморфного шифрования (ПГШ) на основе матричных полиномов с помощью метода пакетного шифрования в один шифротекст нескольких открытых текстов с последующей комплексной обработкой зашифрованных данных. Пакетное шифрование сводится к тому, что при одной операции над двумя шифротекста происходит одновременное выполнение операций покоординатно над всеми данными, содержащимися в этих шифротекста в виде открытых текстов (SIMD). **Задачами** определено построение алгоритмов полностью гомоморфного шифрования данных с помощью матричных полиномов. В статье использованы методы шифрования: с использованием китайской теоремы об остатках; путем записи в одной матрицы нескольких различных собственных значений при различных собственных векторах; с помощью интерполяции матричных полиномов. В **результате** описано и проанализированы возможные подходы к построению пакетных ПГШ на основании матричных полиномов, а также представлены набор алгоритмов, реализующих криптосхему ПГШ с интерполяцией матричных полиномов. Приведенные алгоритмы и криптосхемы позволяют передавать информацию в сообщениях и данные в запросах в виде открытого текста, поскольку над шифрованными данными можно совершать неограниченное количество сложных алгебраических операций, что затрудняет возможность расшифровки и считывания данных без знания всего алгоритма. Было показано, что построенные криптосхемы превосходят аналоги по эффективности, разработанные исследователями из IBM. Можно сделать следующий **вывод**: пакетное полностью гомоморфного шифрования на основе матричных полиномов способно исключить необходимость хотя бы частичной расшифровки данных для несанкционированных вычислений над зашифрованными массивами данных в облачных хранилищах.

Ключевые слова: полностью гомоморфного шифрования; хранилище данных; алгоритм; шифротекст; криптографические методы; криптосхемы; матричные полиномы.

HOMOMORPHIC ENCRYPTION OF CLOUD DATA BY THE MATRIX POLYNOMIAL METHOD

The **subject matter** of the study is the encryption of information in cloud data computation and storage. Cloud technologies enable reducing the cost of IT infrastructure significantly and responding to changes in computing needs flexibly. In this case, the possibilities to perform calculations on the encrypted data without decrypting should be provided. Fully homomorphic encryption has this feature. The **goal** of this article is to increase the efficiency of fully homomorphic encryption (FHE) on the basis of matrix polynomials using the method of batch encryption to one ciphertext of several plaintexts with the subsequent complex processing of encrypted data. Batch encryption comes down to the fact that while conducting the operation on two ciphertexts, operations are simultaneously conducted coordinatewise on all the data contained in these ciphertexts in the form of plaintexts (SIMD). The **task** is the construction of algorithms of fully homomorphic data encryption using matrix polynomials. The following encryption methods are used in the article: the use of the Chinese remainder theorem; recording several different eigenvalues with different eigenvectors to the same matrix; the interpolation of matrix polynomials. The following **results** were obtained: possible approaches to constructing a batch EHE on the basis of matrix polynomials were described and analyzed, a set of algorithms that implement the FHE crypto scheme with interpolation of matrix polynomials was presented. The above algorithms and crypto schemes enable transmitting information in messages and data in queries as a plain text because an unlimited number of complex algebraic operations can be performed on the encrypted data, which makes it difficult to decrypt and read data without the knowledge of the entire algorithm. The constructed crypto schemes were shown as more efficient than analogues developed by IBM researchers. The following **conclusion** can be made: a batch fully homomorphic encryption using matrix polynomials can eliminate the need for at least partial decryption of data to carry out unauthorized computation on encrypted cloud data arrays.

Keywords: fully homomorphic encryption; databank; algorithm; ciphertext; cryptographic methods; crypto scheme; matrix polynomials.