УДК 004.056:519.2 **DOI:10.30837/rt.2023.4.215.03**

А.М. ОЛЕКСІЙЧУК, д-р техн. наук, О.С. ШЕВЧУК

МЕТОД ВІДНОВЛЕННЯ ЛІНІЙНИХ БЛОКОВИХ КОДІВ НАД ДОВІЛЬНИМ СКІНЧЕННИМ ПОЛЕМ ЗА НАБОРАМИ СПОТВОРЕНИХ КОДОВИХ СЛІВ

Вступ

Задача про відновлення невідомого двійкового лінійного блокового коду за набором його спотворених кодових слів привертає увагу, принаймні, останні 20 років, що пов'язано як з практичними потребами в ефективних методах вивідування конфіденційної інформації, так і з задачами криптоаналізу деяких симетричних криптосистем (див. публікації [1 – 10] та наведені у них посилання).

Майже усі відомі методи відновлення лінійних блокових кодів запропоновані для випадку двійкових кодів і базуються на такому спостереженні. Розглянемо $m\times n$-матрицю Y, яка складається з випадкових слів лінійного (n,k)-коду C, спотворених у двійковому симетричному каналі зв'язку з імовірністю помилки $p\in(0,1/2)$, а також довільний двійковий вектор x довжини n та ваги w. Тоді середня вага випадкового вектора Yx^T дорівнює $m/2\cdot(1-(1-2p)^w)$ або $m/2$ в залежності від того, чи належить вектор x коду $C^\perp$, дуальному до C, чи ні. Таким чином, для відновлення коду $C^\perp$ (а, отже, і коду C) достатньо знайти $n-k$ лінійно незалежних векторів x, для яких слова вигляду Yx^T, що належать коду, породженому стовпцями матриці Y, мають достатньо малу вагу. Для пошуку слів малої ваги у зазначеному коді застосовуються відомі швидкі алгоритми [11 – 13] (відзначимо роботу [7], де уніфіковано та оптимізовано раніше відомі методи відновлення двійкових лінійних блокових кодів шляхом пошуку слів малої ваги у (неспотворених) лінійних блокових кодах).

Аналіз публікацій [1 – 8] показує, що для оцінювання трудомісткості деяких відомих алгоритмів (зокрема, [5, 7]) потрібно мати певну додаткову інформацію про шуканий код (наприклад, знати частину вагового спектру дуального коду $C^\perp$), що не завжди можливо на практиці. Крім того, обґрунтування оцінок трудомісткості деяких алгоритмів, які застосовуються для пошуку слів малої ваги у двійкових кодах [11, 12], базується на евристичних припущеннях, тому для підтвердження цих оцінок потрібні обчислювальні експерименти. Окрім того, складність зазначених алгоритмів швидко зростає з ростом довжини n двійкового коду, що відновлюється (принаймні, як поліном від n, степінь якого залежить лінійно від w), що робить їх малопрактичними вже при $n\geq 2000$ та помірних значеннях w. Нарешті, кореляційні атаки на певні симетричні кодові шифросистеми [9, 10] приводять до задачі відновлення лінійних блокових кодів над довільними скінченними полями чи навіть скінченними абелевими групами.

Мета цієї статті – запропонувати метод відновлення лінійних блокових кодів над довільним скінченним полем, який відрізняється за сутністю від відомих [1 – 8] і полягає у відновленні шуканого коду шляхом розв'язання задачі LPN. Остання добре відома в теорії обчислювальних алгоритмів та криптоаналізі. Вона рівносильна задачі декодування випадкового лінійного блокового коду, а на її складності базується стійкість багатьох сучасних постквантових криптосистем (див., наприклад, [14 – 17]).

На відміну від [1 – 8], де з певною достовірністю відновлюються випадкові слова коду $C^\perp$, запропонований метод полягає у відновленні канонічної твірної матриці коду C, яка визначається за кодом однозначно. При цьому відновлення цієї матриці здійснюється шляхом розв'язання (не більше ніж) $n-1$ систем лінійних рівнянь зі спотвореними правими частинами від k (або меншої кількості) невідомих кожна.

Показано, що запропонований метод надає можливість застосовувати для відновлення лінійних блокових кодів більш широкий клас алгоритмів в порівнянні з раніше відомими, зокрема, алгоритми типу BKW [18, 19], а також алгоритми, викладені в [20 – 22]. При цьому, на відміну від раніше відомих, складність запропонованого методу залежить лінійно від довжини n шуканого коду, проте зростає з ростом його вимірності k відповідно до того, який алгоритм розв'язання задачі LPN застосовується.

Означення основних понять та допоміжні твердження

Позначимо F скінченне поле з q елементів, $F_{m,n}$ – множину $m \times n$-матриць над цим полем, $wt(x) = |\{i \in \overline{1,n} : x_i \neq 0\}|$ – вагу вектора $x = (x_1, \dots, x_n) \in F^n$.

Для будь-якого $\varepsilon \in (0,1)$ назвемо ε-*нерівномірним* розподіл ймовірностей на полі F, який визначається за правилом

$$p(0) = q^{-1}(1 + (q-1)\varepsilon), \; p(a) = q^{-1}(1-\varepsilon), \; a \in F \setminus \{0\}. \tag{1}$$

Надалі закон розподілу (1) позначатимемо символом $U(\varepsilon)$.

Безпосередньо з наведеного означення та формули повної ймовірності випливає такий результат.

Лема 1 [10]**.** Нехай ξ та η є незалежними випадковими величинами на полі F, що мають ε_1-нерівномірний та ε_2-нерівномірний розподіли ймовірностей відповідно. Тоді для будь-якого $c \in F \setminus \{0\}$ випадкова величина $c\xi$ має ε_1-нерівномірний, а випадкова величина $\xi + \eta$ має $\varepsilon_1\varepsilon_2$-нерівномірний розподіл ймовірностей на полі F.

Задача LPN з параметрами (m, n, q, ε) відома у двох варіантах [14 – 17]. Перший з них називається задачею розпізнавання та полягає в наступному.

Спостерігається послідовність незалежних випадкових векторів

$$(A_1, b_1), \dots, (A_m, b_m), \tag{2}$$

де $A_1, \dots, A_m$ – випадкові рівноймовірні вектори довжини n над полем F, а $b_1, \dots, b_m$ або є випадковими рівноймовірними елементами цього поля, що не залежать від $A_1, \dots, A_m$ (гіпотеза H_0), або розподілені за законом

$$b_i = A_i x + \xi_i, \; i \in \overline{1,m}, \tag{3}$$

де $x = (x_1, \dots, x_n) \in F^n$ – невідомий вектор, $\xi_1, \dots, \xi_m$ є незалежними випадковими величинами, розподіленими за законом $U(\varepsilon)$ (гіпотеза H_1). Треба побудувати критерій для перевірки зазначених гіпотез.

Другий варіант задачі LPN полягає у розв'язанні системи рівнянь зі спотвореними правими частинами (3), яка формується зазначеним вище чином. По суті ця задача еквівалентна декодуванню випадкового $(n,k)_q$-коду (тобто коду довжини n та вимірності k над полем з q елементів) у q-му симетричному каналі зв'язку. При $q = 2$ вона зводиться до класичної задачі LPN, на якій базується стійкість багатьох постквантових криптосистем [15 – 17].

Зрозуміло, що вміння розв'язувати задачу LPN у другому варіанті її постановки надає можливість розв'язувати її розпізнавальну версію практично з такими ж трудомісткістю та достовірністю. Має місце також обернене твердження, доведення якого майже дослівно повторює доведення леми 4.2 в [23].

Лема 2. Нехай існує алгоритм A, який розрізняє зазначені вище гіпотези H_0, H_1 з максимальною ймовірністю помилки $\max\{\mathbf{P}(H_0 \mid H_1), \mathbf{P}(H_1 \mid H_0)\} \le \delta$, виконуючи T операцій.

Тоді існує алгоритм, який розв'язує систему рівнянь (3) з імовірністю помилки не вище ніж $qn\delta$ за $O(qn(T+m))$ операцій.

Формальна постановка задачі та основні результати

Нехай C – невідомий лінійний $(n,k)_q$-код з твірною матрицею G. Спостерігається послідовність випадкових векторів

$$Y^{(i)} = U^{(i)}G + \eta^{(i)},\ i \in \overline{1,m}, \tag{4}$$

де $U^{(1)},...,U^{(m)}$, $\eta^{(1)},...,\eta^{(m)}$ є незалежними в сукупності випадковими векторами, розподіленими за законами

$\mathbf{P}(U^{(i)} = a) = q^{-k}$, $a \in F^k$, $\mathbf{P}(\eta^{(i)} = x) = (q^{-1}(1-\varepsilon))^{wt(x)}(q^{-1}(1+(q-1)\varepsilon))^{n-wt(x)}$, $x \in F^n$.

Треба відновити код C за випадковою послідовністю (4).

Для викладення запропонованого методу розв'язання цієї задачі введемо додаткові позначення.

Для будь-якої множини $I \subseteq \overline{1,n}$ та довільної матриці M, стовпці якої занумеровані числами від 1 до n, позначимо M_I підматрицю матриці M, що складається з її стовпців з номерами із множини I. Якщо $I = \{i\}$ є одноелементною множиною, то замість $M_{\{i\}}$ використовуватимемо позначення M_i для i-го стовпця матриці M; i-й рядок матриці M позначатимемо символом $M^{(i)}$.

За означенням матриця $G \in F_{k\times n}$ рангу k називається *спеціальною ступеневою*, якщо вона має вигляд

$$G = \begin{pmatrix} 0\ldots0 & \overset{i_1}{1} * \ldots * & \overset{i_2}{0} * \ldots * & \overset{i_k}{0} * \ldots * \\ 0\ldots0 & 0\,0\ldots0 & 1 * \ldots * & 0 * \ldots * \\ \ldots & \ldots\ \ldots & \ldots & \ldots \\ 0\ldots0 & 0\,0\ldots0 & 0\,0\ldots0 & 1 * \ldots * \end{pmatrix}, \tag{5}$$

де $*$ позначає довільний елемент поля F, $1 \le i_1 < \ldots < i_k \le n$. З відомих теорем лінійної алгебри (див., наприклад, [24, розд. VII]) випливає, що кожен код C має єдину твірну матрицю, яка є спеціальною ступеневою і називається *канонічною твірною матрицею* коду C.

Зафіксуємо множину $I \subseteq \overline{1,n}$ потужності $l \in \overline{1,k}$ та елемент $j \in \overline{1,n}$ такі, що $\operatorname{rank}(G_I) = l$, $j \notin I$, і розглянемо матрицю A, яка складається з рядків $A^{(i)} = U^{(i)}G_I + (\eta^{(i)})_I$, $i \in \overline{1,m}$, а також вектор-стовпець b з координатами $b_i = U^{(i)}G_j + (\eta^{(i)})_j$, $i \in \overline{1,m}$.

Лема 3. Мають місце такі твердження:

1) рядки $A^{(1)},..., A^{(m)}$ є незалежними випадковими рівноймовірними двійковими векторами довжини l;

2) якщо вектор G_j лінійно не залежить від стовпців матриці G_I, то b є випадковим рівноймовірним вектором, що не залежить від матриці A;

3) якщо вектор G_j є лінійною комбінацією стовпців матриці G_I: $G_j = G_I x$, де $x \in F^l$, то x є розв'язком задачі LPN $Az = b = Ax + \xi$, де координати $\xi_1,...,\xi_m$ випадкового вектора ξ мають вигляд $\xi_i = (\eta^{(i)})_j - (\eta^{(i)})_I x$, $i \in \overline{1,m}$, та розподілені за законом $U(\varepsilon^{wt(x)+1})$.

Доведення. Справедливість твердження 1) випливає безпосередньо з наведених означень, а твердження 2) є наслідком твердження 1), оскільки лінійна незалежність вектора G_j від стовпців матриці G_I рівносильна умові $\operatorname{rank}(G_{I\cup\{j\}}) = l$.

Для доведення твердження 3) помітимо, що на підставі наведених вище означень

$$b_i - A^{(i)}x = U^{(i)}G_j + (\eta^{(i)})_j - (U^{(i)}G_I + (\eta^{(i)})_I)x =$$

$$= U^{(i)}(G_j - G_I x) + (\eta^{(i)})_j - (\eta^{(i)})_I x = \xi_i,\ i \in \overline{1,m}.$$

При цьому вектор ξ не залежить від (випадкової рівноймовірної) матриці A, а його координати розподілені за законом $U(\varepsilon^{wt(x)+1})$ на підставі леми 1.

Отримана лема складає основу наступного методу відновлення канонічної твірної матриці G коду C за випадковою послідовністю (4).

Не обмежуючи загальності, припустимо, що кожен стовпець матриці G має вагу від 1 до w, де $w \in \overline{1,k}$. Якщо східці цієї матриці починаються у стовпцях з номерами $1 \le i_1 < \ldots < i_k \le n$ (див. формулу (5)), то $i_1 = 1$ і для кожного $l \in \overline{1,k}$ стовпець з номером i_l матриці G не залежить лінійно від її попередніх стовпців, а стовпці цієї матриці з номерами $i \in \{i_l + 1, \ldots, i_{l+1} - 1\}$ є лінійними комбінаціями її стовпців з номерами $i_1, \ldots, i_l$ (тут і далі вважаємо, що $i_{k+1} = n+1$).

Алгоритм відновлення матриці G (рис. 1) використовує, як допоміжний, довільний фіксований алгоритм $\mathbf{A}$ розв'язання задачі LPN. Останній отримує на вхід впорядкований набір (A, b, θ, δ), де A – випадкова рівноймовірна $m \times l$-матриця над полем F, b – вектор вигляду $b = Ax + \zeta$, де $x \in F^l$, $\zeta = (\zeta_1, \ldots, \zeta_m)$ – випадковий вектор з незалежними координатами, розподіленими за законом $U(\theta)$, $\theta, \delta \in (0,1)$. При цьому вважається, що існує функція $N_{\mathbf{A}} = N_{\mathbf{A}}(l, \theta, \delta)$ така, що за умови $m \ge N_{\mathbf{A}}(l, \theta, \delta)$ виконується нерівність $\mathbf{P}(\mathbf{A}(A, b, \theta, \delta) \ne x) \le \delta$, тобто ймовірність помилки алгоритму $\mathbf{A}$ (відносно сумісного розподілу матриці A та вектора ζ) не перевищує δ, якщо число рівнянь у системі зі спотвореними правими частинами $Ax + \zeta = b$ є не менше ніж $N_{\mathbf{A}}(l, \theta, \delta)$. Трудомісткість алгоритму $\mathbf{A}$ визначається як його часова складність у найгіршому випадку та позначається $T_{\mathbf{A}}(l, \theta, \delta)$.

Зафіксуємо число $m_1 \in \overline{1,m}$ та позначимо Y' і Y'' підматриці, які містяться відповідно в перших m_1 та останніх $m_2 = m - m_1$ рядках матриці Y. Алгоритм на рис. 1 складається з $n-1$ кроків і починає роботу зі стовпця $G_1 = (1, 0, \ldots., 0)^T$ довжини k та множини $I = \{i_1 = 1\}$.

Вхідні дані:

– $m \times n$-матриця $Y = \begin{pmatrix} Y' \\ Y'' \end{pmatrix}$, що складається з рядків (4).

– алгоритм $\mathbf{A}$ розв'язання задачі LPN;

– числа $m_1 \in \overline{1,m}$, $w \in \overline{1,k}$, $\varepsilon, \delta \in (0,1)$;

1. Покласти $G_1 = (1, 0, \ldots., 0)^T$, $I = \{1\}$, $j = 2$.
2. Поки $j \le n$:
   - 2.1. Покласти
     $$A' = Y'_I,\ b' = Y'_j,\ l = |I|,\ l_* = \min\{l, w\},\ p_{l_*} = q^{-1}(q-1)(1 - \varepsilon^{l_*+1}).$$
   - 2.2. Обчислити $x = \mathbf{A}\left(A', b', \varepsilon^{l_*+1}, \delta/2n\right)$.
   - 2.3. Покласти $m_2 = m - m_1$, $\Delta = m_2\left(\dfrac{q^{-1}(q-1) + p_{l_*}}{2}\right)$, $A'' = Y''_I$, $b'' = Y''_j$.
   - 2.4. Якщо $wt(b'' - A''x) \le \Delta$, покласти $G_j = (x, \underbrace{0, \ldots, 0}_{k-l})^T$, $j = j+1$;

     інакше покласти $G_j = (0, \ldots, \underset{l+1}{1}, \ldots, 0)^T$, $I = I \cup \{j\}$, $j = j+1$.

Результат: матриця $(G_1, \ldots, G_n)$.

Рис. 1. Алгоритм відновлення канонічної твірної матриці лінійного блокового коду за набором спотворених кодових слів

Нехай на $(j-1)$-му кроці алгоритму, $j \in \overline{2,n}$, вже побудовані стовпці $G_1, \ldots, G_{j-1}$ матриці G та множина $I = \{i_1, \ldots, i_l\}$, яка складається з номерів, у яких починаються східці матриці $(G_1, \ldots, G_{j-1})$, $1 \le l \le k$, $i_l < j \le i_{l+1}$. Тоді на j-му кроці алгоритму здійснюється перевірка, чи є стовпець G_j лінійною комбінацією стовпців $G_{i_1}, \ldots, G_{i_l}$, та знаходження такої лінійної комбінації у випадку позитивного результату перевірки.

Перевірка базується на лемі 3 та полягає в застосуванні алгоритму $\mathbf{A}$ до матриці A' і вектора b', які визначаються на кроці 2.1 (рис. 1). Алгоритм $\mathbf{A}$ повертає певний вектор $x \in F^l$, який розглядається як кандидат на розв'язок системи лінійних рівнянь зі спотвореними правими частинами $A'z = b'$. Далі підраховується вага вектора $b'' - A''x$, яка порівнюється з порогом Δ. Якщо $wt(b'' - A''x) \le \Delta$, то вектор G_j визначається за формулою: $G_j = (x, \underbrace{0, \ldots, 0}_{k-l})^T$ (при цьому множина I не змінюється). Якщо ж $wt(b'' - A''x) > \Delta$, то G_j визначається як двійковий стовпець довжини k, усі координати якого, за виключенням $(l+1)$-ї, дорівнюють нулю; при цьому множина I приймає нове значення $I = I \cup \{j\}$.

Алгоритм завершує роботу на кроці $j = n$.

Наступне твердження встановлює умови, за яких алгоритм на рис. 1 знаходить шукану матрицю G із заданою ймовірністю помилки, а також надає оцінку трудомісткості цього алгоритму.

Твердження 1. Припустимо, що $T_{\mathbf{A}}(l, \theta, \delta)$ та $N_{\mathbf{A}}(l, \theta, \delta)$ є неспадними функціями кожного з параметрів l, θ при фіксованих значеннях решти. Тоді за умови

$$m_1 \ge N_{\mathbf{A}}(k, \varepsilon^{w+1}, \delta/2n), \; m_2 \ge 2\left(\frac{q^{-1}(q-1)\varepsilon^{w+1}}{2}\right)^{-2} \ln(2n\delta^{-1}) \tag{6}$$

ймовірність помилки алгоритму на рис. 1 не перевищує δ. При цьому його трудомісткість становить $O(nT_{\mathbf{A}}(k, \varepsilon^{w+1}, \delta/2n) + m_2 n)$ операцій.

Доведення. Позначимо π_j ймовірність події, яка полягає в тому, що стовпці $G_1, \ldots, G_{j-1}$ матриці G відновлено правильно, а стовпець G_j – ні, $j \in \overline{2,n}$. Тоді ймовірність помилки алгоритму дорівнює $\pi_2 + \cdots + \pi_n$.

Для оцінювання ймовірності π_j позначимо $I = \{i_1, \ldots, i_l\}$ множину номерів стовпців, у яких починаються східці матриці $(G_1, \ldots, G_{j-1})$, та розглянемо два можливих випадки: $i_l < j < i_{l+1}$ та $j = i_{l+1}$.

У першому випадку ймовірність помилкового відновлення стовпця G_j не перевищує суми ймовірностей двох подій:

$$\Omega_1 = \{x \ne \mathbf{A}\left(A', b', \varepsilon^{l_*+1}, \delta/2n\right)\},$$

$$\Omega_2 = \{x = \mathbf{A}\left(A', b', \varepsilon^{l_*+1}, \delta/2n\right)\} \cap \{wt(b'' - A''x) > \Delta\}.$$

На підставі монотонності функції $N_{\mathbf{A}}$ та першої нерівності (6) маємо $m_1 \ge N_{\mathbf{A}}(k, \varepsilon^{w+1}, \delta/2n) \ge N_{\mathbf{A}}(l, \varepsilon^{l_*+1}, \delta/2n)$. Отже, за означенням функції $N_{\mathbf{A}}$ ймовірність події Ω_1 не перевищує $\delta/2n$. Далі, якщо відбулася подія Ω_2, то на підставі леми 3 $wt(b'' - A''x) = \sum_{i=1}^{m_2} \xi_i$, де $\xi_1, \ldots, \xi_{m_2}$ є незалежними випадковими величинами, розподіленими за законом $\mathbf{P}(\xi_i = 1) = 1 - \mathbf{P}(\xi_i = 0) = q^{-1}(q-1)(1-\varepsilon^{l_*+1})$, $i \in \overline{1,m}$. Звідси, використовуючи умову $l_* \le w$, нерівність Гефдінга [25] та другу нерівність (6), отримаємо, що ймовірність події Ω_2 не перевищує

$$\mathbf{P}(wt(b''-A''x)>\Delta)=\mathbf{P}\left(\sum_{i=1}^{m_2}\xi_i>m_2\left(\frac{q^{-1}(q-1)+p_{l_*}}{2}\right)\right)\le\exp\left\{-2m_2\left(\frac{q^{-1}(q-1)\varepsilon^{l_*+1}}{2}\right)^2\right\}\le$$

$$\le\exp\left\{-2m_2\left(\frac{q^{-1}(q-1)\varepsilon^{w+1}}{2}\right)^2\right\}\le\delta/2n.$$

Таким чином, у першому випадку ($i_l<j<i_{l+1}$) ймовірність π_j не перевищує $\mathbf{P}(\Omega_1)+\mathbf{P}(\Omega_2)\le\delta/n$. Аналогічно доводиться, що у другому випадку ($j=i_{l+1}$) ймовірність π_j не перевищує $\delta/2n$. Отже, $\pi_j\le\delta/n$ для кожного $j\in\overline{2,n}$ і ймовірність помилки алгоритму є $\pi_2+\cdots+\pi_n\le\delta$.

Нарешті, трудомісткість j-го кроку дорівнює

$$O(T_{\mathbf{A}}(l,\varepsilon^{l_*},\delta/2n)+m_2)=O(T_{\mathbf{A}}(k,\ \varepsilon^{w+1},\delta/2n)+m_2)$$

операцій, $j\in\overline{2,n}$, а, отже, трудомісткість алгоритму в цілому становить $O(nT_{\mathbf{A}}(k,\varepsilon^{w+1},\delta/2n)+nm_2)$. Твердження доведено.

Зауважимо, що при практичному застосуванні алгоритму на рис. 1 слушно використовувати різні алгоритми розв'язання систем лінійних рівнянь зі спотвореними правими частинами на кроці 2.2 в залежності від кількості невідомих у системах. Так, для помірних значень l (наприклад, $l\le20$) можна використовувати метод максимуму правдоподібності із застосуванням швидкого перетворення Фур'є [17]. Для більших значень l краще застосовувати алгоритми типу BKW [18, 19] або (при $q=2$, якщо кількість спотворених кодових слів є обмеженою) – алгоритми, викладені в [20, 21]. На останніх кроках (починаючи з $j=i_k+1$) можна прискорити роботу алгоритму на рис. 1, розв'язуючи одночасно декілька систем лінійних рівнянь зі спотвореними правими частинами та однаковою матрицею коефіцієнтів, як це робиться в [26] при відновленні систематичних лінійних блокових кодів. При цьому стовпці $G_{i_k+1},\dots,G_n$ шуканої матриці G співпадають з розв'язками відповідних систем рівнянь, а кроки 2.3, 2.4 (рис. 1) виконувати не треба.

Зауважимо також, що у випадку, коли відома нижня межа d' дуальної відстані коду C, виконання алгоритму на рис. 1 можна почати безпосередньо з кроку $j=d'$, визначаючи G_j як двійковий стовпець довжини k, усі координати якого, за виключенням j-ї, дорівнюють нулю, $j\in\overline{1,d'-1}$.

На завершення розглянемо модифікацію наведеного алгоритму (рис. 2), яка використовує в ролі $\mathbf{A}$ довільний алгоритм розв'язання розрізнювальної версії задачі LPN.

Позначимо $N_{\mathbf{A}}=N_{\mathbf{A}}(n,\varepsilon,\delta)$ нижню межу числа m векторів у послідовності (2), для якого алгоритм $\mathbf{A}$ розв'язує розрізнювальну версію задачі LPN з параметрами (m,n,q,ε) з максимальною ймовірністю помилки не вище ніж δ. Символом $T_{\mathbf{A}}=T_{\mathbf{A}}(n,\varepsilon,\delta)$ позначимо трудомісткість алгоритму $\mathbf{A}$.

Твердження 2. Припустимо, що $N_{\mathbf{A}}$ і $T_{\mathbf{A}}$ є неспадними функціями кожного з трьох аргументів при фіксованих значеннях решти. Тоді за умови $m\ge N_{\mathbf{A}}(k,\varepsilon^{w+1},\delta/qn^2)$ ймовірність помилки алгоритму на рис. 2 не перевищує δ. При цьому його трудомісткість становить $O(qn^2(T_{\mathbf{A}}(k,\varepsilon^{w+1},\delta/2n)+m))$ операцій.

Доведення. Позначимо π_j ймовірність того, що стовпці $G_1,\dots,G_{j-1}$ матриці G відновлено правильно, а стовпець G_j – ні, $j\in\overline{2,n}$. Для оцінювання ймовірності π_j позначимо $I=\{i_1,\dots,i_l\}$ множину номерів стовпців, у яких починаються східці матриці $(G_1,\dots,G_{j-1})$, та розглянемо два можливих випадки: $j=i_{l+1}$ та $i_l<j<i_{l+1}$.

Вхідні дані:

– $m \times n$-матриця Y, що складається з рядків (4).

– алгоритм $\mathbf{A}$ розв'язання розрізнювальної версії задачі LPN;

– числа $w \in \overline{1,k}$, $\varepsilon, \delta \in (0,1)$;

1. Покласти $G_1 = (1, 0, \ldots, 0)^T$, $I = \{1\}$, $j = 2$.

2. Поки $j \le n$:

Покласти

$$A = Y_I,\ b = Y_j,\ l = |I|,\ l_* = \min\{l, w\},\ p_{l_*} = q^{-1}(q-1)(1-\varepsilon^{l_*+1})$$

та застосувати алгоритм $\mathbf{A}$ до вхідних даних $A, b, \varepsilon^{l_*+1}, \delta/2n$.

Якщо $\mathbf{A}$ повертає висновок про справжність гіпотези H_0, покласти

$$G_j = (0, \ldots, \underset{l+1}{1}, \ldots, 0)^T,\ I = I \cup \{j\},\ j = j+1.$$

Інакше відновити розв'язок χ системи лінійних рівнянь зі спотвореними правими частинами $Az = b$ шляхом застосування алгоритму $\mathbf{A}$ до вхідних даних $A, b, \varepsilon^{l_*+1}, \delta/2qn^2$ (див. лему 2) та покласти $G_j = (x, \underbrace{0, \ldots, 0}_{k-l})^T$, $j = j+1$.

Результат: матриця $(G_1, \ldots, G_n)$.

Рис. 2. Модифікований алгоритм відновлення канонічної твірної матриці лінійного блокового коду за набором спотворених кодових слів

В першому випадку помилкове завершення алгоритму на рис. 2 є виключно наслідком помилки алгоритму $\mathbf{A}$ при його застосуванні до вхідних даних $A, b, \varepsilon^{l_*+1}, \delta/2n$ на кроці 2 і, оскільки $m \ge N_{\mathbf{A}}(k, \varepsilon^{w+1}, \delta/2qn^2) \ge N_{\mathbf{A}}(l, \varepsilon^{l_*+1}, \delta/2n)$, то на підставі означення функції $N_{\mathbf{A}}$ маємо $\pi_j \le \delta/2n$. В другому випадку π_j не перевищує суму ймовірностей двох подій, перша з яких полягає у помилковому завершенні алгоритму $\mathbf{A}$ при його застосуванні до вхідних даних $A, b, \varepsilon^{l_*+1}, \delta/2n$, а друга – у помилковому завершенні цього алгоритму при його застосовуванні до вхідних даних $A, b, \varepsilon^{l_*+1}, \delta/2qn^2$ (як зазначено у формулюванні леми 2) за умови заперечення першої події. Отже, згідно з лемою 2, $\pi_j \le \delta/2n + (qn)(\delta/2qn^2) = \delta/n$.

Таким чином, $\pi_j \le \delta/n$ для кожного $j \in \overline{2,n}$ і ймовірність помилки алгоритму є $\pi_2 + \cdots + \pi_n \le \delta$.

Нарешті, оцінка трудомісткості алгоритму на рис. 2 випливає безпосередньо з його опису та леми 2. Твердження доведено.

Висновки

Отримані результати показують, що відновлення $(n,k)_q$-коду C за набором його слів, спотворених у q-му симетричному каналі з ймовірністю помилки $q^{-1}(1-\varepsilon)$, є не складніше ніж розв'язання n систем лінійних рівнянь від k невідомих з правими частинами, спотвореними з імовірністю $q^{-1}(1-\varepsilon^{w+1})$, де w – максимальна вага стовпців канонічної твірної матриці коду C. Це надає можливість застосовувати для відновлення лінійних блокових кодів більш широкий клас алгоритмів в порівнянні з раніше відомими. При цьому, на відміну від раніше відомих методів [1 – 8], складність запропонованого методу залежить лінійно від довжини шуканого коду (проте зростає з ростом його вимірності відповідно до того, який алгоритм розв'язання задачі LPN застосовується).

Запропонований метод показує, що основним параметром, від якого залежить складність відновлення лінійного блокового коду, є його вимірність (а не довжина), що, в принципі, надає можливість помітно пришвидшити відомі алгоритми відновлення двійкових лінійних блокових кодів за наборами спотворених кодових слів.

Список літератури:

1. Valembois A. Detection and recognition of a binary linear code // Discrete applied mathematics. 2001. Vol. 111, No. 1–2. P. 199–218. DOI: https://doi.org/10.1016/s0166-218x(00)00353-x.

2. Cluzeau M. Block code reconstruction using iterative decoding techniques // IEEE Conference ISIT'06. 2006. P. 2269–2273. DOI: https://doi.org/10.1109/isit.2006.261971.

3. Barbier J., Sicot G., Houcke S. Algebraic approach of the reconstruction of linear and convolutional error correcting codes // World Academy of Science, Engineering and Technology. 2006. Vol. 16. P. 66–71.

4. Sicot G., Houcke S. Blind detection of interleaver parameters // *Signal Process.* 2009. Vol. 89 (4). P. 450–462.

5. Cluzeau M., Finiasz M. Recovering a code's length and synchronization from a noisy intercepted bitstream // IEEE Conference ISIT'09. 2009. P. 2737–2731. DOI: https://doi.org/10.1109/isit.2009.5205843.

6. Karimian Y., Ziapuor S., Attari M.A. Parity-check matrix recognition from noisy codewords // ArXiv: 1205.4641v1 [cs.IT]. 2012. 22 p.

7. Carrier K., Tillich J.-P. Identifying an unknown code by partial Gaussian elimination // Designs, Codes and Cryptography. 2018. Vol. 87, No. 2–3. P. 685–713. DOI: https://doi.org/10.1007/s10623-018-00593-7.

8. Fast Blind Recovery of Linear Block Codes over Noisy Channels / P. Wang et al. // 2023 IEEE International Symposium on Information Theory (ISIT), Taipei, Taiwan, 25–30 June 2023. 2023. DOI: https://doi.org/10.1109/isit54713.2023.10206775.

9. Олексійчук А.М, Шевчук О. С. Оцінка ефективності атак на основі підібраних відкритих текстів на криптосистему Рао-Нама над скінченною абелевою групою // Радіотехніка. 2021. Т. 1, № 205. С. 22–31. DOI: https://doi.org/10.30837/rt.2021.2.205.02.

10. Шевчук О. С. Рандомізована симетрична криптосистема Мак-Еліса на основі узагальнених кодів Ріда-Соломона // Радіотехніка. 2020. Т. 1, № 200. С. 25–36. DOI: https://doi.org/10.30837/rt.2020.1.200.03.

11. Canteaut A., Chabaud F. A new algorithm for finding minimum-weight words in a linear code: application to McEliece's cryptosystem and to narrow-sense BCH codes of length 511 // IEEE Transactions on Information Theory. 1998. Vol. 44, No.1. P. 367–378. DOI: https://doi.org/10.1109/18.651067.

12. Chose P., Joux A., Mitton M. Fast Correlation Attacks: An Algorithmic Point of View // Advances in Cryptology – EUROCRYPT 2002. Berlin, Heidelberg, 2002. P. 209–221. DOI: https://doi.org/10.1007/3-540-46035-7_14.

13. Dubiner M. Bucketing Coding and Information Theory for the Statistical High-Dimensional Nearest-Neighbor Problem // IEEE Transactions on Information Theory. 2010. Vol. 56, No. 8. P. 4166–4179. DOI: https://doi.org/10.1109/tit.2010.2050814.

14. Cryptographic Primitives Based on Hard Learning Problems / A. Blum et al. // Advances in Cryptology – CRYPTO' 93. Berlin, Heidelberg, 1994. P. 278–291. DOI: https://doi.org/10.1007/3-540-48329-2_24.

15. Pietrzak K. Cryptography from Learning Parity with Noise // SOFSEM 2012: Theory and Practice of Computer Science. Berlin, Heidelberg, 2012. P. 99–114. DOI: https://doi.org/10.1007/978-3-642-27660-6_9.

16. Bogos S. LPN in cryptography: an algorithmic study. PhD thesis. Ecole Polytechnique Federale de Lausanne. 2017. p.177. URL: https://infoscience.epfl.ch/record/228977/files/EPFL_TH7800.pdf.

17. Ігнатенко С. М. Методи розв'язання задачі LPN над скінченними кільцями для оцінювання стійкості симетричних постквантових шифросистем : дис. … канд. техн. наук : 05.13.21. Харків, 2021. 179 с. DOI: http://dspace.univer.kharkov.ua/handle/123456789/16047.

18. Blum A., Kalai A., Wasserman H. Noise-tolerant learning, the parity problem, and the statistical query model // Journal of the ACM. 2003. Vol. 50, No. 4. P. 506–519. DOI: https://doi.org/10.1145/792538.792543.

19. Bogos S., Tramer S., Vaudenay S. On solving LPN using BKW and variants. Implementation and analysis // Cryptology ePrint Archive, Report 2015/049. URL: http://eprint.iacr.org/2015/049.

20. Алексейчук А.Н., Грязнухин А.Ю. Быстрый алгоритм восстановления истинного решения фиксированного веса системы линейных булевых уравнений с искаженной правой частью // Прикладная дискретная математика. 2013. Т. 20. С. 59–70.

21. Grigorescu E., Reyzin L., Vempala S. On Noise-Tolerant Learning of Sparse Parities and Related Problems // Lecture Notes in Computer Science. Berlin, Heidelberg, 2011. P. 413–424. DOI: https://doi.org/10.1007/978-3-642-24412-4_32.

22.Zhang B., Xu C., Meier W. Fast Correlation Attacks over Extension Fields, Large-Unit Linear Approximation and Cryptanalysis of SNOW 2.0 // Lecture Notes in Computer Science. Berlin, Heidelberg, 2015. P. 643–662. DOI: https://doi.org/10.1007/978-3-662-47989-6_31.

23.Regev O. On lattices, learning with errors, random linear codes, and cryptography // Journal of the ACM. 2009. Vol. 56, No. 6. P. 1–40. DOI: https://doi.org/10.1145/1568318.1568324.

24.Глухов М.М., Елизаров В.П, Нечаев А.А. Алгебра : учебник в 2-х т., Т. 1. Москва : Гелиос АРВ, 2003. 336 с.

25.Hoeffding W. Probability Inequalities for Sums of Bounded Random Variables // Journal of the American Statistical Association. 1963. Vol. 58, No. 301. P. 13–30. DOI: https://doi.org/10.1080/01621459.1963.10500830.

26.Алексейчук А.Н., Грязнухин А.Ю. Метод восстановления систематических линейных кодов по наборам искаженных кодовых слов // Прикладная дискретная математика. 2013. Т. 12, № 2. С. 313–318.

Надійшла до редколегії 11.10.2023

Відомості про авторів:

Олексійчук Антон Миколайович – д-р техн. наук, доц., Інститут спеціального зв'язку та захисту інформації Національного технічного університету України “Київський політехнічний інститут імені Ігоря Сікорського”, професор спеціальної кафедри № 1; Україна; e-mail: alex-dtn@ukr.net

Шевчук Ольга Сергіївна – викладач спеціальної кафедри № 5; Інститут спеціального зв'язку та захисту інформації Національного технічного університету України “Київський політехнічний інститут імені Ігоря Сікорського”, Україна; e-mail: olia13511@gmail.com