УДК 519.87

Є. В. Івохін, *д.ф.-м.н, доцент*
Київський національний університет імені Тараса Шевченка,
просп. акад. Глушкова, 4 Д., Київ, 03127, Україна
ivohin@univ.kiev.ua

ПРО МОДИФІКАЦІЮ АЛГОРИТМУ ШИФРУВАННЯ ДАНИХ МЕРКЛА-ХЕЛМАНА

В роботі розглянуто алгоритм шифрування даних на основі задачі про рюкзак. Визначено завдання щодо підвищення криптостійкості алгоритму. Запропоновано схему шифрування, побудовану з використанням простих чисел та операцій над ними спеціального вигляду. Проілюстровано використання алгоритму та його модифікації на прикладі конкретної текстової послідовності.

Ключові слова: *задача про рюкзак, надзростаючий рюкзак, криптостійкість алгоритму, шифрування, розшифрування, послідовності простих чисел.*

Проблеми шифрування з використанням задачі про рюкзак. Одним із перших алгоритмів для узагальненого шифрування з відкритим ключем є алгоритм рюкзака, розроблений Р. Мерклом та М. Хелманом [1]. Цей алгоритм, що отримав назву алгоритму Меркла-Хелмана, спочатку використовувався лише для шифрування, але пізніше Аді Шамір [2] адаптував криптосистему для створення цифрового підпису. Безпека алгоритмів рюкзака спирається на проблему вирішення задачі про рюкзак, яка є NP-повною проблемою.

Задача про рюкзак добре відома. Припускається, що задано множину предметів різної ваги. Необхідно визначити, як можна покласти деякі з цих предметів у рюкзак таким чином, щоб вага рюкзака стала рівною наперед заданому значенню. Більш формально, для заданого набору значень M_1, M_2,..., M_n (n – потужність заданої множини) та величини S потрібно визначити значення b_i, $i=\overline{1,n}$, такі що

$$S = b_1 M_1 + b_2 M_2 + ... + b_n M_n, \qquad (1)$$

де b_i, $i=\overline{1,n}$, може бути або нулем, або одиницею. Одиниця показує, що предмет кладуть у рюкзак, а нуль – що не кладуть.

Наприклад, ваги предметів мають значення 1, 5, 6, 11, 14 та 20. Можна наповнити рюкзак таким чином, щоб його вага стала рівною 22, використавши величини 5, 6 і 11. З другого боку, неможливо упакувати рюкзак так, щоб його вага дорівнювала 24. У загальному випадку час, необхідний для вирішення цієї проблеми, зі зростанням кількості предметів у наборі зростає експоненційно.

В основі алгоритму Меркла-Хелмана лежить ідея шифрувати повідомлення на основі ключа – послідовності ваг задачі про рюкзак (відкритий ключ). Предмети з набору обираються за допомогою блоку відкритого тексту, рівного за довжиною кількості предметів у наборі (біти відкритого тексту відповідають значенням b_i, $i=\overline{1,n}$), а шифротекст є отриманою сумою.

Однак, виявилось, що ця схема є криптографічно нестабільною. Як суттєве покращення базового алгоритму Меркла-Хелмана було запропоновано створення закритого ключа, який є перетвореною послідовністю ваг задачі про рюкзак спеціального вигляду. Цей варіант використовувався у двох модифікаціях: одноетапній та багатоетапній. Але запропоновані вдосконалення не забезпечили криптостійкості алгоритму. Вперше про його небезпеку було повідомлено у роботі [2]. Більш того, схема алгоритму дозволяє визначити вхідну послідовність без використання будь-якого закритого ключа [3]. Тому зрозуміло, що практично одразу після створення розпочався пошук модифікацій запропонованого алгоритму, що забезпечують підвищений захист від зламу. Для подолання недоліків базової схеми Родні Гудман та Ентоні Маколі [4] розробили процедуру, що базується на модульних рюкзаках. Надалі з'ясувалося, що ця схема також небезпечна. Окрім використання модульних рюкзаків, були запропоновані схеми використання інших видів рюкзака. У 1986 р. Харальд Нідерайтер [5] опублікував рюкзачну криптосистему на основі алгебраїчної теорії кодування, яка також була зламана, а у 1988 р. Масакацу Морії та Масао Касахара [6] розробили криптосистему з використанням мультиплікатив-

ного рюкзака. Ця ідея виявилася вдалою і поки що система на мультиплікативних рюкзаках не зламана. Також вдалою виявилась ідея Хусейна Алі Хусейна, Джафара Ваді Абдули Сада та М. Каліфа [7], які у 1991 р. запропонували багатоетапну рюкзачну криптосистему. У ній фіксується рюкзачний вектор на кожному етапі, а вихід (зашифроване повідомлення) після кожної стадії алгоритму використовується як вхідні дані (текст) на наступному етапі. Про вдалу атаку на цю схему на поточний момент не відомо. Продовжено покращення і класичного алгоритму Меркла-Хелмана.

Надзростаючі рюкзаки. *Означення:* надзростаючою послідовністю називається числова послідовність, кожний член якої більший суми усіх попередніх членів.

Наприклад, числова послідовність {1,3,6,13,27,52} є надзростаючою, а послідовність {1,3,4,9,15,25} – ні. В цьому контексті варто згадати і добре відому послідовність чисел Фібоначчі, яка, незважаючи на швидке зростання елементів, не є надзростаючою.

Розглянемо просту проблему рюкзака. Якщо перелік ваг предметів у наборі є надзростаючою послідовністю, то отриману проблему рюкзака можна легко вирішити. Розв'язок надзростаючого рюкзака здійснюється наступним чином. Необхідно взяти повну вагу і порівняти її з найбільшим числом послідовності. Якщо повна вага менша цього числа, то його не кладуть у рюкзак. Якщо повна вага більша або дорівнює цьому числу, то воно кладеться у рюкзак. Зменшимо вагу рюкзака на це значення і перейдемо до наступного за величиною числа послідовності. Будемо повторювати ці дії, доки процес не завершиться. Якщо повна вага зменшується до нуля, то розв'язок знайдено. У протилежному випадку – ні.

Покладемо для прикладу [8], що повна вага рюкзака має бути 70, а надзростаюча послідовність ваг {2,3,6,13,27,52}. Найбільша вага – 52, яка менша 70, кладеться у рюкзак. Віднімаючи 52 від 70, отримуємо 18. Наступна вага – 27, більша 18, тому число 27 у рюкзак не кладуть. Наступну вагу 13, яка менше 18, кладуть у рюкзак. Віднімаємо 13 з 18 і отримуємо 5. Чергова вага – 6, більша за 5, не кладеться у рюкзак. Продовжуючи цей процес, отримаємо, що ваги 3 і 2 кладуть у рюкзак, і повна вага зменшується до 0, що свідчить про знайдений розв'язок. Якщо розглядати цю процедуру як блок шифрування методом рюкзака Меркла-Хелмана, відкритий текст, отриманий із значення шифротексту 70, був би рівний 110101.

Пошук заповнення нормальних рюкзаків, які не є надзростаючими, являє собою складну проблему. Швидкого алгоритму для вирішення цієї задачі в реальному часі поки не знайдено. Єдиним відомим способом визначити, які предмети упаковано в рюкзаку, є методична перевірка можливих розв'язків до знаходження правильного. Найшвидший алгоритм, беручи до уваги різні евристики, має експоненційну залежність від кількості можливих предметів. Тому, якщо додати до послідовності ваг лише один елемент, знаходження розв'язку задачі стає вдвічі складнішим. Це набагато важче надзростаючого рюкзака, в якому, якщо додати один предмет до послідовності, складність пошуку розв'язку зростає на одну операцію.

Алгоритм Меркла-Хелмана базується на цій властивості. Закритий ключ є послідовністю ваг задачі надзростаючого рюкзака. Відкритий ключ – це послідовність ваг проблеми нормального рюкзака з тим самим розв'язком. Р. Меркл та М. Хелман [1], застосовуючи цілочислову арифметику, розробили спосіб перетворення проблеми надзростаючого рюкзака в проблему нормального рюкзака.

Підсумовуючи, відзначимо, що існує дві різні задачі, одна з яких вирішується за лінійний час, а інша, як вважається, – ні. Просту задачу можна перетворити у складну. Відкритий ключ являє собою складну (важку) проблему, яку дуже просто можна використати для шифрування, але неможливо – для розшифрування повідомлень. Закритий ключ є простою (легкою) проблемою, що надає простий спосіб розшифрування повідомлення. Якщо невідомий закритий ключ, потрібно розв'язувати складну задачу про рюкзак.

Створення відкритого ключа. Розглянемо роботу алгоритму, не заглиблюючись у теорію чисел. Для отримання нормальної послідовності рюкзака візьмемо надзростаючу послідовність рюкзака, наприклад наведену раніше K ={2,3,6,13,27,52}, і домножимо всі значення на число T за модулем N. Значення модуля повинне бути більшим суми всіх чисел послідовності, тобто утворювати з заданою послідовністю надзростаючу послідовність. Оберемо, наприклад, N =105. Множник T повинен бути взаємно простим числом з модулем N, тобто НСД (N , T) = 1. Покладемо, наприклад, T = 31. Нормальною послідовністю рюкзака у цьому випадку буде {62,93,81,88,102,37}, де 62 = 2*31 mod 105;

93 = 3*31 mod 105; 81 = 6*31 mod 105; 88 = 13*31 mod 105; 102 = 27*31 mod 105; 37 = 52*31 mod 105.

Надзростаюча послідовність рюкзака разом з числами N та T {2,3,6,13,27,52; 105, 31} є закритим ключем, а нормальна послідовність рюкзака {62,93,81,88,102,37} – відкритим.

Шифрування. Для шифрування повідомлення розбивається на блоки, що за довжиною дорівнюють кількості елементів послідовності рюкзака. Далі, вважаючи, що одиниця відповідає наявності члена послідовності, а нуль – його відсутності, обчислюємо повні ваги рюкзаків – по одному для кожного блоку повідомлень.

Якщо припустити, що повідомлення у бінарному вигляді подається як 011000110101101110, шифрування, яке використовує попередню послідовність рюкзака, буде здійснюватися таким чином: вхідне повідомлення у блочному вигляді = 011000 110101 101110, звідки 011000 відповідає числу 93 + 81 = 174; 110101 відповідає числу 62 + 93 + 88 + 37= = 280; 101110 відповідає числу 62+81 + 88 + 102= = 333.

У результаті шифротекстом повідомлення 011000110101101110 є числова послідовність 174,280,333.

Розшифрування. Законний отримувач даного повідомлення знає закритий ключ: оригінальну надзростаючу послідовність, а також значення N та T, які використовувалися для перетворення її в нормальну послідовність рюкзака. Для розшифрування повідомлення отримувач визначає мультиплікативне обернене T^{-1}, таке що $T*(T^{-1})\ (mod\ N)$ = 1. Кожне значення щифротексту домножується на $T^{-1}\ mod\ N$, а потім розгортається в суму за допомогою закритого ключа, щоб отримати значення відкритого тексту.

Для наведеного вище прикладу надзростаючою послідовністю є {2,3,6,13,27,52}, $N = 105$, T = 31. Шифротекстом буде послідовність 174,280,333. У цьому випадку T^{-1} дорівнює 61 (31*61mod105=1891mod105=1), тому значення шифротексту домножуються на величину 61mod105. Таким чином, маємо 174*61 mod 105 = 9 = 3 + 6, що відповідає 011000; 280*61 mod 105 = 70 = 2 + 3 + 13 + 52, що відповідає 110101; 333*61 mod 105 = 48 = 2 + 6 + 13 + 27, що відповідає 101110.

Розшифрованим відкритим текстом є бінарні послідовності 011000 110101 101110, що повністю відповідає вхідному повідомленню.

Підвищення крипостійкості алгоритму шифрування. Сучасні дослідження по вдосконаленню алгоритму здійснюються за двома основними напрямками. Перший з них об'єднує роботи, що спрямовані на використання різних варіантів задачі про рюкзак. Другий напрямок досліджує застосування різних додаткових схем та процедур, що підвищують захищеність алгоритму.

Запропонуємо модифікацію базової схеми на основі використання послідовностей простих чисел спеціального вигляду.

Позначимо $P_k(a)$ – k-те просте число, що не менше цілого $a \ge 0$, $k = 0,1,2,...$.

Нескладно перевірити, що справедливі співвідношення:

1) $P_0(0) = 0$, $P_0(1) = 1$, $P_1(0) = 1$;

2) $P_0(a) = a$, якщо число $a \ge 0$ – просте, $P_0(a)$ – не існує, якщо $a \ge 0$ – непросте;

3) $P_j(a) \le P_k(a)$, якщо $j \le k$, $P_j(a) < P_k(a)$ при $j < k$, $j \in Z$, $k \in Z$;

4) $P_j(a) = P_j(a+1) = ... = P_j(a+l)$ для усіх $1 \le l < P_{j+1}(a) - P_j(a)$, $j = 0,1,2,...$, $a \ge 0$;

5) $P_k(a) = P_1(P_{k-1}(a)) = ... = P_{k-1}(P_1(a))$, $k = 1,2,3,...$, $a \ge 0$;

6) $P_j(a) \ge a$ для усіх $j = 0,1,2,...$, якщо число $a \ge 0$ – просте, $P_j(a) > a$ для усіх $j = 1,2,...$, якщо $a \ge 0$ – непросте;

7) $P_j(a) < a$ для $j = -1,-2,...,j_0$, де номер $j_0 < 0$ визначається як найменший індекс простого числа з послідовності, для якого $0 \le P_{j_0}(a) < a$.

Надалі будемо розглядати випадки, коли $a > 0$ – просте число або $a = 0$. У даному випадку існує число $P_0(a)$. Крім цього, виходячи з наведених вище властивостей, справедливі співвідношення:

$$P_j(a)/P_k(a) \le 1,\ P_j(a)/P_k(a) = P_j(a)/P_1(P_{k-1}(a)) = \ldots = P_j(a)/P_j(P_{k-j}(a)) = P_j(a)/P_{k-j}(P_j(a)),$$

для будь-яких $j,k \in Z$, $j \le k$.

Крім традиційних арифметичних операцій додавання, віднімання, множення та ділення, на послідовності чисел $P_k(a)$, $a \ge 0$, $k = 0,1,2,...$, введемо операцію зсуву на m, $m \in N \cup \{0\}$, простих чисел у вигляді

$$P_k(a) \oplus m = P_{k+m}(a) = P_m(P_k(a)), \quad (2)$$

яка не виводить за межі заданої послідовності, та операцію p- кратної композиції ($p \in N$) відношення двох чисел $P_k(a)$ та $P_l(a)$, $k = 0,1,2,...$, $l = 0,1,2,3,...$, у вигляді

$$P_k(a) \big/ P_l(a) \circ p = P_k(a) \oplus p \big/ P_l(a) \oplus p = P_{k+p}(a) \big/ P_{l+p}(a) = P_p(P_k(a)) \big/ P_p(P_l(a)). \quad (3)$$

Числа, що входять до відкритого ключа $K = \{r_1,...,r_n\}$, є довільними цілими числами. Для визначення величини рівня складності кодування кожного числа $r_j \in K, j = \overline{1,n}$ доповнимо відкритий ключ двома довільними значеннями k та l, $k \le l$, $k = 0,1,2,...$, $l = 1,2,3,...$, за якими з послідовності простих чисел визначаються величини $s = P_k(a)$, $q = P_l(a)$ для деякого числа $a \ge 0$. За таких умов отримуємо, що $0 \le s/q \le 1$. Ця величина може служити показником складності кодування, а за допомогою значень s та q можна змінити ваги відкритого ключа K і величини N та T, наприклад, за такою схемою:

- для чисел T та N, які є взаємно простими числами, обчислюються числа $P_s(T)$ та $P_q(N)$ відповідно, де $s = P_k(a)$, $q = P_l(a)$;

- для цілих чисел $r_j \in K, j = \overline{1,n}$, які входять до ключа K, обчислюються величини $u_j = P_s(r_j) - r_j$ та $v_j = P_q(r_j)$, $j = \overline{1,n}$, $s = P_k(a)$, $q = P_l(a)$;

- формується вектор пар елементів $\overline{K} = \{(u_1, v_1),...,(u_n, v_n)\}$, який буде новим значенням відкритого ключа *K*. Величини $P_s(T)$, $P_q(N)$ та вектор $\overline{K}$ передаються отримувачу разом з зашифрованим за допомогою елементів v_j, $j = \overline{1,n}$, повідомленням.

Зрозуміло, що ця схема допускає різні модифікації. Як основні з них можна розглянути:

- схему з кодуванням лише чисел N і T,

- паралельне кодування чисел N, T та чисел ключа K,

- послідовне кодування чисел N і T у числа $P_s(T)$ та $P_q(N)$, відповідно, з подальшим перетворенням елементів ключа $K = \{r_1,...,r_n\}$ за традиційною процедурою $r_j * P_s(T) \bmod P_q(N)$, $j = \overline{1,n}$, $s = P_k(a)$, $q = P_l(a)$ і кодуванням отриманих значень вектором пар елементів $\overline{K} = \{(u_1, v_1),...,(u_n, v_n)\}$.

Іншою важливою рисою запропонованої схеми є можливість динамічно змінювати значення $c = P_k(a)$ і $z = P_l(a)$. Використовуючи операцію зсуву для зміни чисел s та q, у вхідній ключовій послідовності задається величина зсуву m, $m \in N \cup \{0\}$, яка дозволяє обчислити нові прості числа $P_k(a) \oplus m$ та $P_l(a) \oplus m$ за формулою (2). Отримані значення можна розглядати як нові параметри процедур шифрування та розшифрування текстових повідомлень.

Для демонстрації дії розробленої схеми повернемося до попереднього прикладу. Покладемо a = 1. Припустимо, що k = 4, l = 7. Відповідні прості числа s = 7, q = 17. Для елементів відкритого ключа *K*={62,93,81,88, 102,37} та величин N = 105, T = 31 обчислимо відповідні значення u_1 = 35, v_1 = 149, u_2 = 34, v_2 = 179, u_3 = 28, v_3 = 167, u_4 = 25, v_4 = 173, u_5 = 35, v_5 = 191, u_6 = 30, v_6 = 109, $P_7(31)$ = 61, $P_{17}(105)$ = 193. Тоді новим відкритим ключем буде множина пар значень $\overline{K}$ = {(35,149), (34,179), (28,167), (25,173), (35,191), (30,109)}. Величини $P_s(T)$ = 61 та $P_q(N)$ =193 є зашифрованими значеннями T = 31, N = 105. При цьому, усі отримані значення нескладно розшифровуються у попередній стан за допомогою простих чисел s = 7, q = 17.

Зашифровані елементи вхідного повідомлення 011000 110101 101110 з новим відкритим ключем будуть мати вигляд: 011000 задається числом 179+167=346; 110101 задається числом 149+179+173+109 = 610; 101110 задається числом 149+167+173+191 = 680, а шифротекстом заданого повідомлення 011000110101101110 буде числова послідовність 346,610,680.

Висновки. На основі використання алгоритму про рюкзак та послідовностей простих чисел запропоновано підхід для шифрування повідомлень, який підвищує криптостійскість традиційного алгоритму Меркла-Хелмана за рахунок розширення послідовності шифротексту. Схеми шифрування та розшифрування проілюстровано рядом конкретних прикладів. Відзначено конструктивність та ефективність запропонованої методики на основі використання

послідовностей простих чисел та операцій над ними спеціального вигляду.

Список літератури

1. Merkle, R. and Hellman, M. (1978). Hiding information and signatures in trapdoor knapsacks. *IEEE Trans. Information Theory,* 24 (5), pp. 525-530.
2. Adi Shamir. (1982). A polynomial time algorithm for breaking the basic Merkle-Hellman cryptosystem. *Lecture Notes in Computer Science (CRYPTO-1982)*, Springer, Berlin, pp. 279-288.
3. Brickell, E. F. (1985). Breaking iterated knapsacks. *Lecture Notes in Computer Science (CRYPTO-1984),* Springer, Berlin, (196), pp. 342-358.
4. Goodman, R. M. F. and McAuley, A. J. (1985). New trapdoor-knapsack public-key cryptosystem. *IEE Proceedings*, 132 (E), 6, pp. 289-292.
5. Niederreiter, H. (1986) Knapsack-type cryptosystems and algebraic coding theory. *Problems of Control and Information Theory*, (15), pp. 159-166.
6. Masakatu Morii and Masao Kasahara (1988). New public key cryptosystem using discrete logarithm over GF(p). *IEICE Transactions*, J71-D, num. 2, pp. 448-453.
7. Hussain Ali Hussain, Jafar Wadi Abdul Sada and Saad M. Kalipha (1991). New multistage knapsack public-key cryptosystem. *International Journal of Systems Science*, (22), num.11, pp. 2313-2320.
8. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си / Б. Шнайер. – М. : Триумф, 2002. – 693 с.

References

1. Merkle, R. and Hellman, M. (1978). Hiding information and signatures in trapdoor knapsacks. *IEEE Trans. Information Theory,* 24 (5), pp. 525-530.
2. Adi Shamir. (1982). A polynomial time algorithm for breaking the basic Merkle-Hellman cryptosystem. *Lecture Notes in Computer Science (CRYPTO-1982)*, Springer, Berlin, pp. 279-288.
3. Brickell, E. F. (1985). Breaking iterated knapsacks. *Lecture Notes in Computer Science (CRYPTO-1984),* Springer, Berlin, (196), pp. 342-358.
4. Goodman, R. M. F. and McAuley, A. J. (1985). New trapdoor-knapsack public-key cryptosystem. *IEE Proceedings*, 132 (E), 6, pp. 289-292.
5. Niederreiter, H. (1986) Knapsack-type cryptosystems and algebraic coding theory. *Problems of Control and Information Theory*, (15), pp. 159-166.
6. Masakatu Morii and Masao Kasahara (1988). New public key cryptosystem using discrete logarithm over GF(p). *IEICE Transactions*, J71-D, num. 2, pp. 448-453.
7. Hussain Ali Hussain, Jafar Wadi Abdul Sada and Saad M. Kalipha (1991). New multistage knapsack public-key cryptosystem. *International Journal of Systems Science*, (22), num.11, pp. 2313-2320.
8. Schneier, B. (2002) Applied cryptography*:* protocols, algorithms, and source codes in C. Moscow: Triumph, 693 p. [in Russian].

Стаття надійшла до редакції 22.04.2014.

E. V. Ivohin, *D.Sc., associate professor*
Kyiv National Taras Shevchenko University
akad. Glushkov blvd, 4 D., Kyiv, 03127, Ukraine
ivohin@univ.kiev.ua

ABOUT MERKLE-HELLMAN ALGORITHM OF DATA ENCRYPTION MODIFICATION

In this paper a data encryption algorithm based on the knapsack problem is considered. The problem of algorithm's cryptographical security increasing is determined. An encryption scheme, based on the use of prime numbers and operations on them by special form, is offered. The use of the algorithm and its modification on the example of specific text sequence is illustrated.

Keywords: *knapsack problem, superincreasing knapsack, algorithm's cryptographical security, encryption, decryption, sequences of simple numbers.*