P., Villebrun E., Hoeher P. A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain // in Proc. Int. Conf. on Commun., ICC-95. – 1995. – June. – P. 1009-1013. ***7.*** *William E. Ryan Concatenated Convolutional Codes and Iterative Decoding // Department of Electrical and Computer Engineering - The University of Arizona, 2001. – July 10. – P. 8-23.* ***8.*** *Qi J. Turbo code in IS-2000 code division multiple access communications under fading // Wichita State University. – 1999. P. 38-59.* ***9.*** *Jeong Woo Lee The Study of Turbo Codes And Iterative Decoding // Dissertation for the degree of Doctor of Philosophy in Electrical Engineering in Graduate College of the University of Illinois at Urbana-Champaign, 2003. P. 86-96.* ***10.*** *Malardel F. Simulation and Optimisations of the Turbo Decoding Algorithm // Signal Processing Research Institute – University of South Australia, 1996. – July-November. – P. 23-26.* ***11.*** *Золотарев В. В., Овечкин Г. В. Помехоустойчивое кодирование. Методы и алгоритмы: Справочник // Москва: Горячая линия – Телеком. – 2001. С. 104.* ***12.****Ливенцев С. П., Алексеев Д. А., Зайцев С. В. Анализ характеристик перемежителей, используемых в турбокодах // Зв'язок. – 2005. – № 3. – С. 57-61.*

УДК 621.391.7

РОЗПОДІЛ СЕКРЕТНИХ КЛЮЧІВ НА ОСНОВІ РЕКУРЕНТНИХ ПОСЛІДОВНОСТЕЙ

Юрій Яремчук
Вінницький національний технічний університет

Анотація: **Пропонується метод розподілу секретних ключів, в основі якого лежить використання властивостей класу рекурентних послідовностей, для обчислення елементів яких вірні рекурентні співвідношення з коефіцієнтами, що пов'язані з початковими елементами послідовностей.**
Summary: **In the given work the method of secret keys distribution is offered, in which base use of properties of the class recurrent of sequences lays, at calculation of which elements the recurrences with coefficients coupled to initial elements of sequences are used.**
Ключові слова: **Захист інформації, криптографія, розподіл ключів, рекурентні послідовності.**

І Вступ

Шифрування інформації може забезпечувати ефективний захист лише за умови вирішення проблеми керування ключами. Розподіл ключів є однією з фундаментальних задач криптографії. Найбільш гостро проблема розподілу ключів стоїть в симетричних криптосистемах, де перед початком роботи необхідно передати секретний ключ обом сторонам.

Існує декілька шляхів вирішення проблеми розподілу ключів [1].

Фізичний розподіл. Розподіл ключів традиційним фізичним шляхом за допомогою довірених кур'єрів або озброєної охорони ключів. До 70-х років це був чи не єдиний безпечний шлях передавання ключів. Однак, такий спосіб передавання залежав від кур'єра: якщо його буде вбито, або підкуплено, то система шифрування буде скомпрометована.

Для двох сторін A та B фізичний розподіл ключів може організовуватись такими способами [2]:

1) сторона A вибирає ключ і фізично доставляє стороні B;

2) ключ вибирає третя сторона C і фізично доставляє його учасникам A і B;

3) якщо учасники обміну ключів A і B використовують деякий загальний ключ, то одна з сторін може передати новий ключ іншій стороні в шифрованому вигляді, використовуючи старий ключ;

4) якщо учасники обміну ключів A і B мають криптографічно захищені канали зв'язку з третьою стороною C, то остання може доставити ключ учасникам A і B цими захищеними каналами.

Розподіл за допомогою методів з секретним ключем. Генерування ключів і розподіл між будь-якими двома користувачами здійснюється на основі довгострокових секретних ключів, що розподілені між цими користувачами та третьою стороною – певним центром довіри. Такий шлях розподілу є достатньо ефективним, однак має недоліки. Зокрема, обидва користувача та центр довіри повинні працювати у режимі онлайн. Крім того, необхідно забезпечувати розподіл довгострокових ключів. Найбільш відомими протоколами розподілу ключів, що базуються на симетричному шифруванні, є протоколи Барроуза [3], Нідхейма-Шредера [4], Отвей-Ріса [5] та Kerberos [6].

Розподіл ключів за допомогою методів з відкритим ключем. Розподіл ключів між користувачами здійснюється в режимі онлайн використовуючи криптосистеми з відкритим ключем. Такий шлях розподілу ключів є найбільш поширеним, оскільки виключає необхідність третьої сторони – посередника та фізичного розподілу ключів секретним каналом зв'язку. Перша практична схема такого обміну була запропонована Діффі та Хеллманом [7].

Метод розподілу ключів Діффі-Хеллмана в класичному вигляді є недостатньо криптостійким, оскільки він не передбачає перевірки автентичності джерела повідомлень. Для того, щоб ключі були узгоджені лише між двома користувачами, необхідно, щоб обидві сторони були впевнені один в одному. Цей недолік може бути усунутий завдяки модифікацій базового методу, наприклад, так як це було показано в роботі [8].

Криптостійкість методу Діффі-Хеллмана базується на складності обчислення дискретних логарифмів, що на сьогодні відноситься до задач, що важко вирішуються. Слід зазначити, що проблема дискретного логарифмування досліджується на сьогодні доволі активно, про що свідчать результати робіт [9 – 12]. Тому пошук ефективних методів розподілу ключів, що базуються на асиметричному шифруванні, залишається актуальним.

Так Сміт і Скіннер [13] запропонували аналог методу ключового обміну Діффі-Хеллмана, який назвали LUCDIF. В основу запропонованого методу покладено рекурентні послідовності Люка за модулем простого числа p замість модулярного піднесення до степеня. Блейхенбахер, Босма і Ленстра [14] показали, що функції Люка, які є аналогом проблеми дискретного логарифмування, в багато разів послаблюють проблему дискретного логарифмування, а сам метод не володіє якимось суттєвими перевагами в порівнянні з оригінальним методом.

Рекурентні послідовності Люка є окремим випадком узагальненої рекурентної послідовності, яка породжується таким співвідношенням [15]

$$u_n = a_1 \cdot u_{n-1} + a_2 \cdot u_{n-2} + \mathrm{K} + a_k \cdot u_{n-k},$$

де a_1, a_2, . . , a_k – коефіцієнти, k – порядок послідовності, виходячи з початкових елементів u_0, u_1, . . . , u_k.

В даній роботі для процедури розподілу ключів пропонуються до використання більш загальні рекурентні послідовності та зовсім інший підхід до їх застосування, ніж був запропонований Смітом і Скіннером.

II Рекурентні V_k та U_k -послідовності

В роботі [16] запропоновані такі рекурентні послідовності

– V_k^+ –послідовність – послідовність чисел, що обчислюються за формулою

$$v_{n,k} = g_k v_{n-1,k} + g_1 v_{n-k,k} \tag{1}$$

для початкових значень $v_{0,k} = 1$, $v_{1,k} = g_2$ для $k = 2$; $v_{0,k} = v_{1,k} = \ldots = v_{k-3,k} = 0$, $v_{k-2,k} = 1$, $v_{k-1,k} = g_k$ для $k > 2$; де g_1, g_k – цілі числа; n і k – цілі додатні;

– V_k^- –послідовність – послідовність чисел, що обчислюються за формулою

$$v_{n,k} = \frac{v_{n+k,k} - g_k \cdot v_{n+k-1,k}}{g_1}, \tag{2}$$

для n – від'ємних з початковими значеннями $v_{-1,k} = 0$, $v_{-2,k} = g_1^{-1}$ для $k = 2$; $v_{-1,k} = 0$, $v_{-2,k} = g_1^{-1}$, $v_{-3,k} = v_{-4,k} = \ldots = v_{-k,k} = 0$ для $k > 2$;

– V_k – послідовність – послідовність чисел, яка складається з V_k^+ –послідовності та V_k^- –послідовності.

– U_k –послідовність – послідовність чисел, що обчислюються за формулою

$$u_{n,k} = g_k u_{n-1,k} + g_1 u_{n-k,k} \tag{3}$$

для початкових значень $u_{0,k} = g_1$, $u_{1,k} = g_2$, $u_{2,k} = g_3$, ... $u_{k-1,k} = g_k$; де g_1, g_2, g_3, ... , g_k – цілі числа; n і k – цілі додатні числа .

Для будь-яких цілих додатних n, m та k отримана така властивість

$$u_{n+m,k} = v_{m+(k-2),k} \cdot u_{n,k} + g_1 \cdot \sum_{i=1}^{k-1} v_{m+(k-2)-i,k} \cdot u_{n-k+i,k} \,. \tag{4}$$

Для будь-яких цілих додатних n та k, таких що $n \geq k$, отримана властивість, яка дозволяє обчислювати елементи U_k – послідовності тільки на основі елементів V_k^+ – послідовності

$$u_{n,k} = g_k \cdot v_{n-1,k} + g_1 \cdot \sum_{i=1}^{k-1} g_i \cdot v_{n-i-1,k} \,. \tag{5}$$

III Метод розподілу ключів

Розробку методу розподілу ключів пропонується здійснювати за допомогою асиметричного шифрування, запропонованого в роботі [16]. Ідея методу наведена в роботах [17, 18].

Суть методу розподілу ключів, що буде розглянуто, базується на властивості (4), яка дозволяє обчислити елемент $u_{n+m,k}$ використовуючи елементи V_k^+ та U_k – послідовностей, причому зробити це двома шляхами: або використовуючи елементи $v_{m+i,k}$, $i = \overline{-1, k-2}$, та $u_{n-i,k}$, $i = \overline{0, k-1}$, або використовуючи елементи $v_{n+i,k}$, $i = \overline{-1, k-2}$, та $u_{m-i,k}$, $i = \overline{0, k-1}$.

Тоді, якщо один користувач для будь-якого вибраного ним випадкового числа a обчислить $u_{a-i,k}$, $i = \overline{0, k-1}$, а другий користувач аналогічним чином обчислить $u_{b-i,k}$, $i = \overline{0, k-1}$, то, обмінявшись обчисленими значеннями, кожен з них зможе отримати $u_{a+b,k}$, продовжуючи обчислення на своєму боці за формулою (4), використовуючи відповідно свої числа a або b. В цьому випадку $u_{a+b,k}$ буде ключем розподілу, а числа a і b секретним ключем кожного користувача. Причому, a і b – це частини секретного ключа кожного користувача, оскільки попереднє отримання ключа розподілу будь-яким користувачем не можливе без отримання відповідної інформації від іншого користувача.

Виходячи з цього схема розподілу ключів має вигляд, представлений на рис. 1.

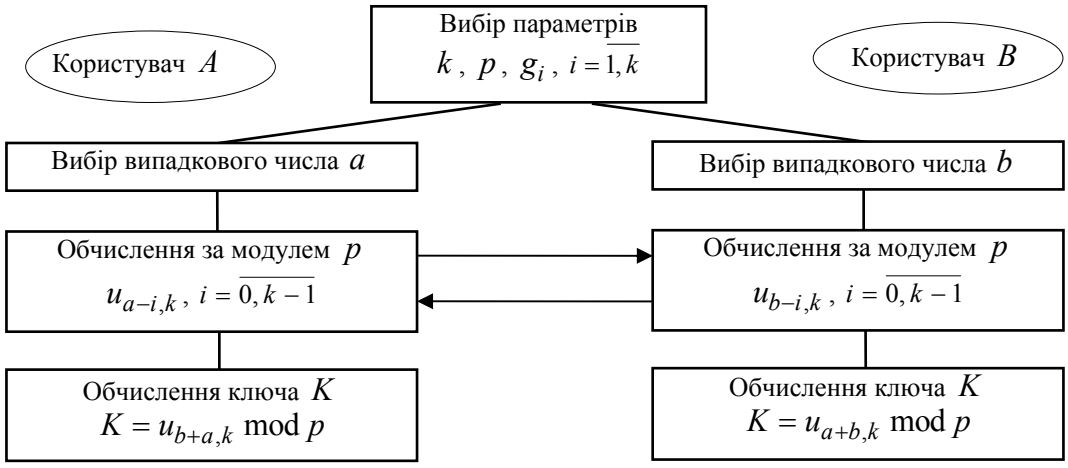

Рисунок 1. Схема розподілу ключів на основі елементів U_k – послідовностей.

Операція за модулем в схемі розподілу ключів використовується для обмеження розрядності чисел

під час виконання арифметичних операцій.

Відповідно до запропонованого методу розподілу ключів основні обчислення виконуються за формулою (4). Для обчислення елементу $u_{n+m,k}$ за цією формулою потрібні елементи $v_{m+i,k}$, $i=\overline{-1,k-2}$, та елементи $u_{n-i,k}$, $i=\overline{0,k-1}$. Обчислення останнього набору елементів здійснюється за формулою (5), для чого необхідно мати елементи $v_{n+i,k}$, $i=\overline{-2k+1,-1}$. Звідси виходить, що всього для обчислення елементу $u_{n+m,k}$ за формулою (4) потрібно мати елементи $v_{n+i,k}$, $i=\overline{-2k+1,k-2}$ V_k–послідовності. Для отримання останнього набору елементів достатньо обчислити будь-які k послідовних елементів з цього набору, оскільки інші можуть бути обчислені на їх основі за формулами (1) або (2).

Алгоритм розподілу ключів буде мати такий вигляд.

Алгоритм 1.

П.1. Задати параметр k.

П.2. Вибрати p.

П.3. Вибрати $g_1, g_2, \ldots, g_k$.

П.4. Опублікувати параметри.

П.5. Користувачу A вибрати випадкове число a, а користувачу B вибрати випадкове число b.

П.6. Користувачу A обчислити за модулем p $v_{a+i,k}$, а користувачу B обчислити за модулем p $v_{b+i,k}$ для $i=\overline{-(k-1),k-2}$.

П.7. Користувачу A обчислити за модулем p $v_{a+i,k}$, а користувачу B обчислити за модулем p $v_{b+i,k}$ для $i=\overline{-2k+1,-k}$ за формулою (2).

П.8. Користувачу A обчислити за модулем p $u_{a-i,k}$, а користувачу B обчислити за модулем p $u_{b-i,k}$ для $i=\overline{0,k-1}$ за формулою (5).

П.9. Користувачу A передати $u_{a-i,k}$ користувачу B, а користувачу B передати $u_{b-i,k}$ користувачу A, де $i=\overline{0,k-1}$.

П.10. Користувачу A обчислити ключ K за формулою

$$K = u_{b+a,k} \bmod p,$$

а користувачу B обчислити ключ K за формулою

$$K = u_{a+b,k} \bmod p,$$

де $u_{b+a,k}$ та $u_{a+b,k}$ обчислюються за формулою (4).

В п.2 проводиться вибір параметру p, який є модулем при обчисленнях в представленому алгоритмі та визначає верхню межу діапазону чисел, що отримуються під час цих обчислень.

В п.3 відбувається вибір параметрів g_i, $i=\overline{1,k}$. Оскільки значення будь-якого числа в розробленому алгоритмі обмежується параметром p, вказані параметри слід вибирати в діапазоні $[1, p-1]$. При цьому вибір можна здійснювати за допомогою будь-якого генератора випадкових чисел у вказаному діапазоні.

Обчислення елементів $v_{a+i,k}$ та $v_{b+i,k}$ для $i=\overline{-(k-1),k-2}$, яке виконується у п.6 алгоритму 1 може здійснюватись за одним з алгоритмів прискореного обчислення елементів V_k–послідовності. В [16] запропоновано два варіанти таких алгоритмів: або на основі відомого бінарного методу, або методу з розкладанням індексу. В [16] показано, що алгоритм на основі методу з розкладанням індексу є менш складним, ніж алгоритм на основі бінарного методу, але останній не потребує попередніх обчислень елементів V_k–послідовності та додаткової пам'яті для зберігання цих елементів в процесі роботи алгоритму.

IV Оцінювання складності виконання алгоритму розподілу ключів

Не важко помітити, що Користувач A і Користувач B виконують за алгоритмом 1 однакову кількість арифметичних операцій над великими числами. Тому для визначення складності обчислень за цим алгоритмом достатньо визначити складність обчислення з боку одного з них, а потім подвоїти отримане значення.

Складність обчислень за алгоритмом 1 з боку Користувача A визначається складністю обчислень за модулем p елементів $v_{a+i,k}$, $i = \overline{-(k-1), k-2}$, елементів $v_{a+i,k}$, $i = \overline{-2k+1, -k}$, за формулою (2), елементів $u_{a-i,k}$, $i = \overline{0, k-1}$ за формулою (5), та елементу $u_{b+a,k}$ за формулою (4). Обчислення першого набору елементів може бути здійснено за методом обчислення елементів V_k – послідовності з розкладанням індексу, який представлений в [16]. В тій же роботі визначено, що складність обчислень даного набору елементів складає приблизно $H^2 q \cdot [6H(k^2 + k) + 3(3k^2 + k)]$ операцій над машинними одиницями інформації, де H – кількість машинних одиниць інформації для зберігання великого числа, q – кількість розрядів машинної одиниці інформації.

Обчислення інших елементів V_k та U_k – послідовностей за модулем p за формулами (2), (4) та (5) потребує виконання приблизно $k^2 + 4k$ множень, k^2 додавань та k віднімань над машинними одиницями інформації. Враховуючи оцінки складності виконання арифметичних операцій за модулем над числами великої розрядності, що представлені в [16], складність обчислень за формулами (2), (4) та (5) буде складати приблизно $6H(H+1)(k^2 + 4k) + 2Hk^2(H+1) + 3Hk(H+1)$ операцій над машинними одиницями інформації. Виходячи з того, що під час реалізації криптографічних методів в сучасних комп'ютерних системах оперують ключами, що мають розмір 1024 і більше розрядів ($Hq \geq 1024$), отримана оцінка буде значно меншою за оцінку складності обчислення набору елементів $v_{a+i,k}$, $i = \overline{-(k-1), k-2}$, а тому може не враховуватись в загальній оцінці складності всього алгоритму розподілу ключів.

Таким чином, складність виконання запропонованого алгоритму розподілу секретних ключів з боку одного користувача складає $H^2 q \cdot [6H(k^2 + k) + 3(3k^2 + k)]$ операцій над машинними одиницями інформації.

Порівнюючи запропонований метод розподілу ключів з відомим методом Діффі-Хеллмана відносно складності виконання розподілу ключів слід відзначити таке. В запропонованому методі обом користувачам необхідно виконувати обчислення певного елементу U_k – послідовності по одному разу, в той час як за методом Діффі-Хеллмана їм необхідно виконувати піднесення до степеня по два рази. В [16] показано, що складність обчислення певного елементу U_k – послідовності має той же порядок, що і складність піднесення до заданого степеня. Тому можна стверджувати, що представлений метод має приблизно вдвічі меншу складність обчислень, ніж метод Діффі-Хеллмана. Крім того, запропонований метод має значно простішу процедуру завдання параметрів, оскільки їх вибір не потребує проведення складних обчислень над великими числами.

V Криптостійкість методу розподілу ключів

Визначимо теоретичну криптостійкість запропонованого методу розподілу ключів за допомогою теоретико-складносного підходу. Для цього визначимо, по-перше, що запропонований метод використовує фіксовані значення параметрів k, q, H, наприклад $k = 3$, $q = 16$, $H = 32$, а параметр безпеки – будь-яке натуральне число. По-друге, будемо вважати, що зловмиснику відома така інформація: алгоритм розподілу ключів; параметри алгоритму k, p, g_i, $i = \overline{1, k}$; елементи $u_{a-i,k} \bmod p$, $i = \overline{0, k-1}$, або $u_{b-i,k} \bmod p$, $i = \overline{0, k-1}$, що передаються від одного користувача до іншого в процесі розподілу ключів. По-третє, об'єм обчислень будемо вважати «практично нездійсненним», якщо найкращий алгоритм, який буде використовувати зловмисник для зламу, буде виконуватись не за поліноміальний час. Доведемо, що не існує поліноміальних алгоритмів для зламу запропонованого методу.

Виходячи з тієї інформації, яка відома зловмиснику, основні його спроби можуть бути спрямовані на

отримання секретного значення a або b відповідно з відомих елементів $u_{a-i,k} \bmod p$, $i = \overline{0,k-1}$, або $u_{b-i,k} \bmod p$, $i = \overline{0,k-1}$. Розглянемо можливі спроби зловмисника на основі інформації, яка буде надходити від користувача A, оскільки, очевидно, що спроби з використанням інформації від користувача B будуть аналогічними.

Перше, що може спробувати зловмисник, – отримати секретне значення a шляхом послідовних обчислень за модулем p за формулою (3), доки не буде отримано значення $u_{a,k} \bmod p$. Аналіз показує, що такі обчислення потребують виконання $3aH(4H+5)$ операцій над машинними одиницями інформації. Тобто, якщо продуктивність комп'ютера дорівнює 2^{34} операцій за секунду, для представлення a використовують 1024 розряди, $H = 32$, то для виконання цих операцій потрібно приблизно 2^{979} років, що є практично нездійсненним.

Для отримання значення $u_{a,k} \bmod p$ або набору елементів $v_{a+i,k}$, $i = \overline{-(k-1),k-2}$, зловмисник може застосовувати формули безпосереднього обчислення елементів відповідно U_k або V_k –послідовностей через початкові елементи. Аналіз цих формул, які наводяться в роботі [16], показує, що вони є більш складними за кількістю виконуваних операцій, ніж формула (3), тобто дана спроба теж є практично нездійсненною.

Оскільки елементи $u_{a-i,k} \bmod p$, $i = \overline{0,k-1}$, відомі, то можлива спроба знаходження елементів $v_{a+i,k}$, $i = \overline{-(k-1),k-2}$, використовуючи формулу (5). Реалізація цієї спроби зводиться до розв'язання системи з k рівнянь з $k+1$ невідомими. Математична задача розв'язання такої системи рівнянь, враховуючи велику розрядність коефіцієнтів та невідомих, наразі не має ефективного поліноміального алгоритму, а отже є практично нездійсненною.

Таким чином, злам запропонованого методу розподілу ключів не може бути виконаний за поліноміальний час, а, отже, запропонований метод є теоретично стійким.

Слід також зазначити, що запропонований алгоритм розподілу ключів дає можливість змінювати параметр k, що дасть змогу підвищувати криптостійкість за рахунок збільшення складності виконання алгоритму.

VI Висновки

Запропоновано метод розподілу секретних ключів, який базується на властивостях рекурентних V_k та U_k – послідовностей. Дослідження представленого методу розподілу ключів показало, що з точки зору теоретичної криптостійкості метод є стійким. В порівнянні з відомим методом розподілу ключів Діффі-Хеллмана запропонований метод має три важливі переваги. По-перше, з точки зору складності обчислень, порівняно з методом Діффі-Хеллмана, запропонований метод забезпечує для кожного користувача майже вдвічі меншу складність обчислень. По-друге, метод має простішу процедуру завдання параметрів. По-третє, метод дозволяє встановлювати необхідну криптостійкість залежно від параметру *k*.

*Література: **1.** В. Столлингс. Криптография и защита сетей: принципы и практика. – М.: ИД «Вильямс», 2001. – 672 с. **2.** Н. Смарт. Криптография. – М.: Техносфера, 2005. – 528 с. **3.** M. BURROWS, M. ABADI, AND R. NEEDHAM, "A logic of authentication", Proceedings of the Royal Society of London Series A: Mathematical and Physical Sciences, 246 (1989), 233–271. **4.** R. M. NEEDHAM AND M.D. SCHROEDER, "Using encryption for authentication in large networks of computers", Communications of the ACM, 21 (1978), 993–999. **5.** D. OTWAY AND O. REES, "Efficient and timelymutual authentication", Operating Systems Review, 21 (1987), 8–10. **6.** S. P. MILLER, B. C. NEUMAN, J. I. SCHILLER, AND J. H. SALTZER, "Kerberos authentication and authorization system", Section E. 2. 1 of Project Athena Technical Plan, MIT, Cambridge, Massachusetts, 1987. **7.** W. DIFFIE AND M.E. HELLMAN "New directions in cryptography", IEEE Transactions on Information Theory, 22 (1976), 644–654. **8.** W. DIFFIE, P. C. VAN OORSCHOT, AND M. J. WIENER, "Authentication and authenticated key exchanges", Designs, Codes and cryptography, 2 (1992), 107–125. **9.** A. M.ODLYZKO. Discrete logarithms: the past and the future. Designs, Codes and Cryptography, 19:129-154, 2000. **10.** E. F. BRICKELL AND A. M. ODLYZKO, "Cryptanalysis: A survey of recent results", Proceedings of the IEEE, 76 (1988), 578–593. **11.** Y.*

DESMEDT AND A. M. ODLYZKO, "A chosen text attack on the RSA cryptosystem and some discrete logarithm schemes", Advances in Cryptology–CRYPTO '85 (LNCS 218), 516–522, 1986. ***12.*** *B. A. LAMACCHIA AND A. M. ODLYZKO, "Computation of discrete logarithms in prime fields", Designs, Codes and Cryptography, 1 (1991), 47–62.* ***13.*** *Smith P. and Skinner C. A public-key cryptosystem and a digital signature system based on the Lucas function analogue to discrete logarithms // In Advances in Cryptology, Asiacrypt '94, Springer-Verlag. - 1995. - Pp. 357-364.* ***14.*** *Bleichenbacher D., Bosma W., and Lenstra A. Some remarks on Lucas-based cryptosystems // In Advances in Cryptology, Crypto '95, Springer-Verlag. - 1995. - Pp.386-396.* ***15.*** *Маркушевич А. И. Возвратные последовательности. - М.: Наука, 1975. - 48 с.* ***16.*** *Яремчук Ю. Є. Методи та засоби шифрування інформації на основі рекурентних послідовностей. – Дис. ... канд. техн. наук: 05.13.21 – К., 2000. – 179 с.* ***17.*** *Яремчук Ю. Є. Криптографічні методи та засоби шифрування інформації на основі рекурентних послідовностей : Монографія. – Вінниця : Книга-Вега, 2002. – 136 с.* ***18.*** *Яремчук Ю. Є. Використання рекурентних послідовностей в задачах криптографічного захисту інформації // Захист інформації – 2002 – №6. – С. 37–45.*

УДК 681.3.06

МОДЕЛІ БЕЗПЕКИ ДЛЯ ПРОТОКОЛІВ УЗГОДЖЕННЯ КЛЮЧІВ, ЩО ЗАСНОВАНІ НА ВЛАСТИВОСТІ НЕРОЗРІЗНЕНОСТІ

Олексій Мелецький
Харківський національний університет радіоелектроніки

Анотація: **Розглядаються моделі безпеки для протоколів узгодження ключів на базі нерозрізненості. Проводяться порівняння існуючих моделей, визначаються недоліки моделей та даються рекомендації щодо використання.**
Summary: **The indistinguishability-based security models of key agreement protocols reviewed. In this report we review and make a comparison of existing security models, checking the flaws and a recommendation offers.**
Ключові слова: **Протокол узгодження ключів, модель безпеки.**

I Вступ

Загальні визначення протоколів узгодження ключів

Головним предметом розгляду даної роботи є моделі безпеки для протоколів узгодження ключів, які задовольняють певним критеріям безпеки. Щоб протокол міг мати практичне застосування, він повинен бути безпечним. Тому для початку дамо визначення терміну «безпечний протокол» [1].

Визначення 1. *Безпечний протокол – це розподілений алгоритм взаємодії, який визначає послідовність кроків точно визначених дій, в яких беруть участь дві або більше сторін з метою досягнення певних цілей безпеки.*

За визначені дії протоколу зазвичай приймаються відправлення та отримання повідомлень, які реалізуються на базових криптографічних примітивах. До базових криптографічних примітивів відносять такі: шифрування, цифровий підпис та гешування [2, 3]. У кожного протоколу є послідовність взаємодії між користувачами протоколу, тобто поведінка кожної сторони протоколу узгодження ключів може бути описана алгоритмом, який виробляє вихідні данні у відповідь на вхідні дані [4]. Для вхідних даних найбільш характерними є наступні дані – короткотермінові секрети (наприклад, сеансові ключі), довгострокові секрети (довгострокові ключі користувачів, майстер ключі уповноваженого), загальносистемні параметри, вхідні повідомлення та інші дані. Вихідні дані – це вихідні повідомлення, результати виконання протоколу, встановлені секрети.

В сучасній криптографії протоколи узгодження ключів відіграють одну з головних ролей, це базовий елемент при побудові безпечних, складних, багаторівневих протоколів та криптографічних систем. Жодна криптографічна система, яка претендує називатися захищеною, не може обійтися без розподілення та узгодження ключів.

Визначення 2. *Протокол узгодження ключів [5] – це механізм, який дає змогу двом або більше сторонам взаємодії узгодити спільну таємницю, діючи при цьому в мережах зв'язку, які повністю контролюються порушником.*