УДК 512.624.95

**Летенко Ю.О., Рябухо О.М., Турка Т.В.**

[1] *студентка 5 курсу фізико-математичного факультету, ДДПУ*

[2] *доцент кафедри алгебри, ДДПУ*

[3] *доцент кафедри алгебри, ДДПУ*

e-mail: letenko.yuliya@yandex.ru

# ПРОТОКОЛИ РОЗПОДІЛУ ТА УЗГОДЖЕННЯ КЛЮЧА

Дана робота присвячена вивченню проблеми створення, обміну та розподілу ключів. Вивчаються і описуються основні правила розподілу та узгодження ключів, досліджуються протоколи початкового розподілу ключів Diffie-Hellman та схема Bloma. На їх основі розроблена програмна реалізація криптографічних протоколів розподілу ключів.

**Ключові слова:** *шифр, ключ, криптографія, асиметричні криптографічні системи, криптографічні протоколи.*

## Вступ

В наш час велика увага приділяється проблемі використання криптографічних методів в інформаційних системах.

Відомо, що криптографічні системи з відкритим ключем (асиметричні) мають перевагу над системами з закритим ключем (симетричнними), оскільки не вимагають безпечного каналу для передачі таємного ключа. Та нажаль, більшість асиметричних систем (наприклад RSA) набагато повільніші за симетричні системи (наприклад AES). Так, на практиці, всі системи з закритим ключем використовуються щоб зашифрувати «довгі» повідомлення. Тому виникає проблема пересилки таємних ключів.

Розв'язання цієї проблеми базується на криптографічних протоколах. Це відносно молода галузь математичної криптографії (перші протоколи з'явилися близько 40 років тому), але вона бурхливо розвивається і на даний момент перетворилася в основний об'єкт дослідження в теоретичній криптографії.

Blom представив свою схему попереднього розполіду ключа в [2]. Узагальнення цієї схеми можна знайти в працях Blunda та інших [3], а також Beimela і Chora [1]. Diffie і Hellman опублікували свій алгоритм обміну ключа в [4]. Незалежно від них ідею обміну ключа сформулював Merkle [7]. Інформація про обмін ключа з підтвердженням преставлена в роботах Diffie, van Oorschota і Wienera [5].

# 1. Основні теоретичні відомості сучасної криптології

Розглянемо симетричні криптографічні системи. При використанні симетричної криптосистеми дві сторони, що вступають в інформаційний обмін повинні спочатку узгодити секретний сесійний ключ, тобто ключ для шифрування всіх повідомлень, переданих в процесі обміну. Цей ключ повинен бути вказаний всім іншим і повинен періодично оновлюватися одночасно у відправника і одержувача. Процес узгодження сесійного ключа називають також обміном або розподілом ключів.

Установлена послідовність дій, які виконуються для розв'язання певного криптографічного завдання, називається *криптографічним протоколом.*

Криптографічні протоколи є важливою складовою частиною криптографічної системи. Основна відмінність протоколу від алгоритму полягає в тому, що реалізація алгоритму припускає активні дії одного суб'єкта, в той час як протокол реалізується в ході взаємодії декількох суб'єктів (сторін протоколу).

Якщо сторони, що взаємодіють, довіряють один одному й готові спільно вирішувати криптографічне завдання, то в цьому випадку використовуються двосторонні протоколи (протоколи без посередника).

Якщо між сторонами можуть виникати розбіжності або їм потрібна підтримка третьої сторони, то використовуються протоколи з посередником (незацікавленою довіреною стороною — (ТА) технічним адміністратором), які називають тристоронніми протоколами. Завдання посередника — забезпечити виконання всіх етапів протоколу, аж до його завершення. ТА відповідальний за підтвердження ідентифікації користувачів, видання сертифікатів, вибір та передачу ключів користувачам.

Існують наступні види протоколів:

- *попередній розподіл ключа;*
- *розподіл ключа сеансу;*
- *узгодження ключів;*

Процес розподілу ключа, а також протокол його узгодження полягають у тому, що наприкінці реалізації протоколу обидві сторони процесу будуть володіти спільним ключем $K$, значення якого не відоме жодній іншій стороні (окрім ТА).

Більшість криптографічних систем вимагають проведення попереднього розподілу секретних ключів. Для попереднього розподілу сторони можуть обмінятися ключами при особистій зустрічі, або доручити доставку ключів спеціально призначеному ТА, чи використовувати для передачі деякий виділений захищений канал.

Залежно від призначення криптографічної системи іноді зручним виявляється розподіляти не самі ключі, а деякі допоміжні ключові матеріали, на підставі яких кожен учасник або група користувачів можуть самостійно обчислити необхідний ключ, використовуючи для цього деяку встановлену заздалегідь процедуру.

Спочатку опишемо основний варіант протоколу. Для кожної пари користувачів ТА вибирає випадково ключ $K_{U,V} = K_{V,U}$ та передає його поза мережею до $U$ і $V$ безпечним каналом (передача ключів відбувається поза мережею, тому що мережа досить небезпечна). Цей підхід дає безумовну безпеку, проте вимагає безпечний канал для передачі інформації між ТА і кожним користувачем.

В основному варіанті ТА генерує $C_n^2$ ключі, передаючи кожен з них парі користувачів мережі. Для передачі ключів ми маємо потребу в безпечному каналі між ТА і кожним із користувачів. Це дає істотний прогрес відносно ситуації, в якій кожна пара користувачів незалежно обмінює між собою ключі за допомогою безпечного каналу, бо завдяки цьому можна обмежити число необхідних безпечних з'єднань з $C_n^2$ до $n$.

Проте, якщо число користувачів велике, то цей метод є недостатньо практичним, однаковою мірою як з точки зору кількості інформації, яку необхідно безпечно передати, так і з точки зору інформації, яку кожен споживач мусить безпечно зберегти (кожен користувач повинен зберегти $n-1$ ключ, натомість ТА необхідно безпечно передати в цілому $C_n^2$ ключі).

## Схема Bloma

Запропонована схема Bloma дозволяє скоротити кількість таємної інформації, яку мусять зберігати користувачі мережі.

Розглядається мережа, що складається з $n$ користувачів. Ключі вибрані із скінченої групи $Z_p$, де $p \geqslant n$ є простим числом.

Значення $k$ визначає максимальну кількість об'єднань атак, яку система зможе витримати, де $1 \leqslant k \leqslant n-2$. У схемі Bloma ТА передає безпечним каналом $k+1$ елемент $Z_p$ кожному користувачеві (на відміну від тривіальної схеми попереднього розподілу ключів, в якій передається $n-1$ елемент). Тут також кожна пара користувачів $U$ і $V$ може підрахувати ключ $K_{U,V} = K_{V,U}$.

Умова безпеки полягає в наступному: будь-яка множина, що найбільше $k$ користувачів, не включаючи $U$ і $V$, не в змозі встановити будь-яку інформацію про ключ $K_{U,V}$ (ми говоримо тут про безумовну безпеку).

## 2. Метод розподілу ключа Діффі-Хелмана

У 1976 році Diffie і Hellman винайшли метод відкритого розподілу ключів. Завдяки цьому методу користувачі можуть обмінюватися ключами по незахищених каналах зв'язку. Його безпека обумовлена трудомісткістю обчислення дискретних логарифмів в скінченному полі, на відміну від легкості розв'язання прямої задачі дискретного піднесення до степеня в тому ж скінченному полі. Суть методу Diffie-Hellman зображено на рисунку 1.

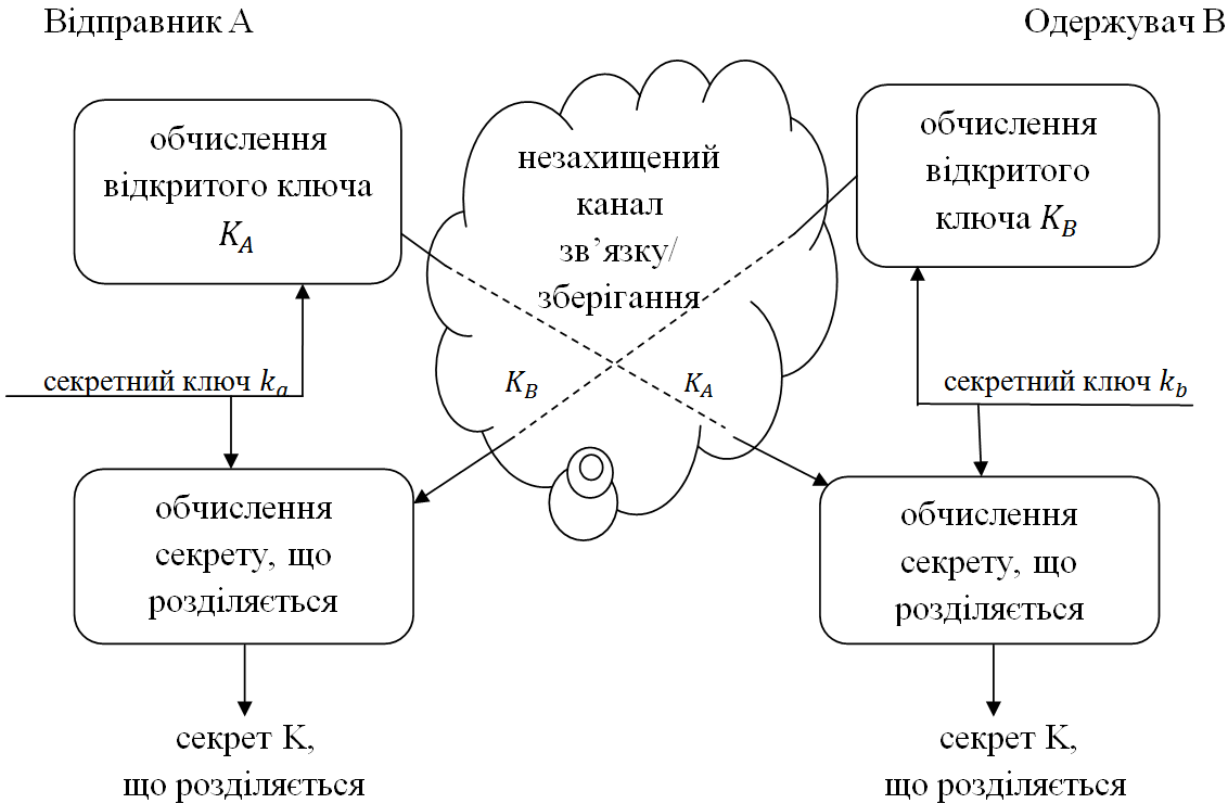


**Рис. 1:** Схема відкритого розподілу ключа Diffie-Hellman

Зловмисник, який перехопив значення відкритих ключів $K_A$ і $K_B$, не може обчислити сесійний ключ $K$, тому що він не має секретних ключів $k_A$ і $k_B$.

Унікальність методу Diffie-Hellman полягає в тому, що пара абонентів має можливість отримувати відоме тільки їм секретне число, передаючи по відкритій мережі відкриті ключі. Завдяки використанню односпрямованої функції операція обчислення відкритого ключа незворотна, тобто неможливо за значенням відкритого ключа абонента обчислити його секретний ключ. Схема Diffie-Hellman дає можливість шифрувати дані при кожному сеансі зв'язку на нових ключах. Не слід забувати, що будь-яке зберігання секретів підвищує ймовірність потрапляння їх в руки конкурентів або супротивника.

### Протокол обміну ключів Diffie-Hellman

Якщо з якихось причин співпраця on-line з ТА непрактична або небажана, можна звернутися до часто застосовуваного методу: протоколу узгодження ключа. При такому підході користувачі $U$ і $V$ разом вибирають ключ, підтримуючи зв'язок за допомогою відкритого каналу.

Першим, і водночас найбільш відомим таким протоколом є *протокол обміну ключа Diffie-Hellman* (винайдений в 1976 році при співпраці W. Diffie і M. Hellman, під впливом роботи R. Merkle). Для його виконання сторони повинні домовитися про значення великого простого числа $p$ і твірного елементу $\alpha$ мультиплікативної групи $Z_p^*$, причому значення $p$ і $\alpha$ відкриті кожному користувачеві мережі (як альтернатива, ці значення могли б бути вибрані користувачем $U$ і передані користувачеві $V$ на першому кроці реалізації протоколу).

Після закінчення протоколу обидва користувачі, $U$ і $V$, отримають в результаті один і той самий спільний ключ:

$$K = \alpha^{a_U a_V} \mod p.$$

Цей протокол, дуже схожий на раніше описану схему попереднього розподілу ключа Diffie і Hellman. Різниця полягає в тому, що показники $a_U$ і $a_V$ користувачів $U$ і $V$, відповідно, не є сталими, а вибираються кожного разу при введенні протоколу в дію. Крім того, в цьому протоколі $U$ і $V$ мають впевненість у актуальності ключа, тому що ключ сеансу залежить від обох випадкових показників $a_U$ і $a_V$.

## 3. Характеристики програм реалізованих протоколів

Розглянемо основні характеристики розроблених в межах даної роботи програм. Програмна реалізація виконана в середовищі програмування Delphi 7.

### Протокол розподілу ключа Blom

Ємнісна складність даного протоколу дорівнює $O(n)$, де $n$ — кількість користувачів. Всю інформацію необхідно зберігати ТА, тому він мусить мати відповідний об'єм пам'яті.

Часова ж складність протоколу розподілу ключа Blom для пар абонентів не залежить від загальної кількості користувачів та обраних значень змінних $p$ і $r_{1,\ldots,n}$. Вона є сталою і дорівнює $O(k)$, де $k$ — кількість атак, що може витримати система.

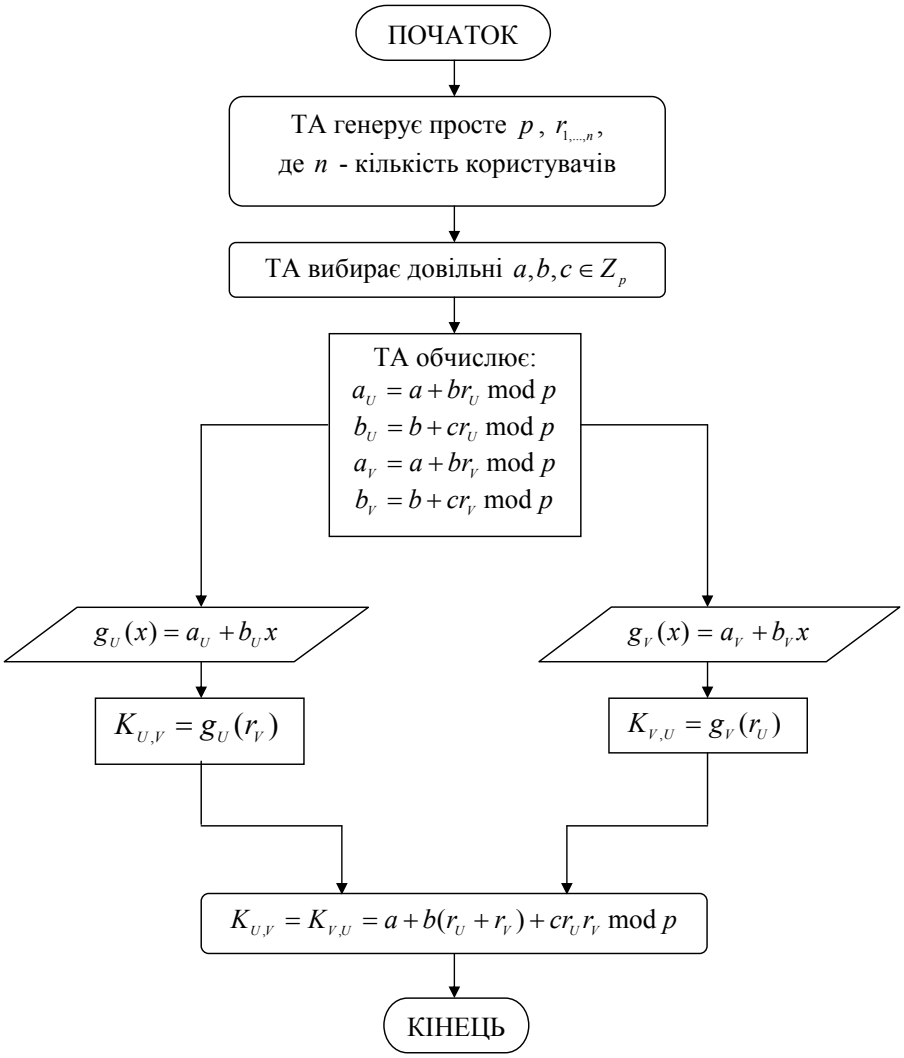


**Рис. 2:** Блок-схема протоколу розподілу ключа Blom (для двох користувачів)

## Протокол розподілу ключів Diffie-Hellman

Процедура підрахунку $b_U$ $(b_V)$ має часову складність $O(p)$. Процедура підрахунку $K_{V,U}$ також має часову складність $O(p)$, а отже весь алгоритм розподілу ключа для двох користувачів має асимптотичну складність $O(p^2)$ проте на практиці це значення в декілька разів менше.

Алгоритм має ємнісну складність $O(1)$, тобто незалежно від кількості користувачів, потребує від ТА одного й того ж об'єму пам'яті. Для кожного користувача при цьому ємнісна складність однакова та також дорівнює $O(1)$.

Програма, написана для реалізації протоколу, не потребує великих об'ємів пам'яті чи процесорного часу, тому можна стверджувати, що протокол Diffie-Hellman може ефективно використовуватися в пристроях, що не мають значних обчислювальних потужностей, проте потребують захищених каналів зв'язку.

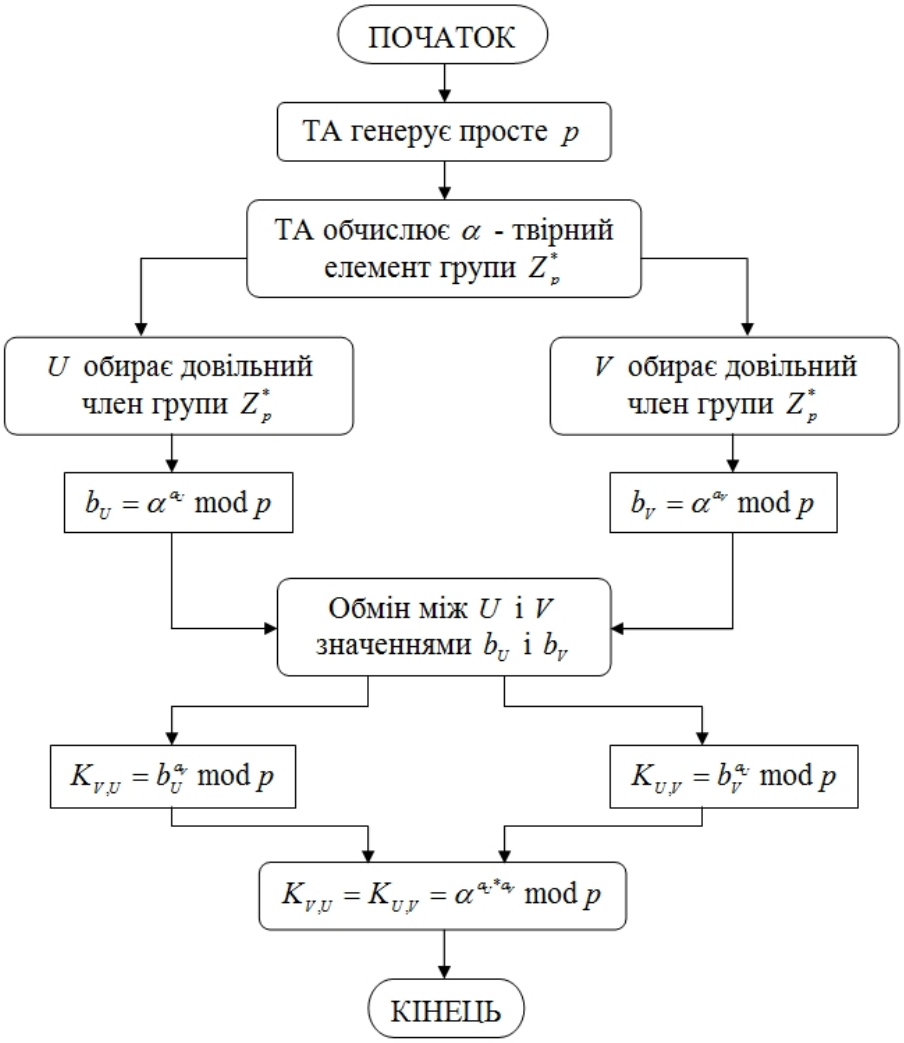


**Рис. 3:** Блок-схема протоколу розподілу ключа Diffie-Hellman

## Висновки

Легко бачити, що кожен з протоколів має свої недоліки та переваги. Протокол Blom не потребує (на відміну від протоколу Diffie-Hellman) значного процесорного часу, проте вимагає виділення порівняно значної кількості пам'яті для зберігання ТА згенерованих значень змінних. Майже всі обчислення в протоколі Blom виконуються ТА, і лише на останньому етапі потребують підрахунків від абонента, через що потребує безпечного каналу зв'язку для передачі отриманих даних кінцевим користувачам. Протокол Diffie-Hellman навпаки мінімізує вплив та діяльність ТА, перекладаючи всі обчислення на абонентів, тобто більш вимогливий до ресурсів користувача, але завдяки цьому не вимагає існування безпечного каналу.

Отже, кожен з протоколів має свої слабкі та сильні сторони, тому вибір оптимального алгоритму залежить від обставин, при яких він буде використовуватися: обчислювальні можливості, доступні об'єми пам'яті, існування чи відсутність безпечних каналів зв'язку та кількість користувачів. Лише оцінивши всі ці параметри можна обрати зручний та ефективний протокол розподілу ключа.

## Література


1. *Beimel A. and Chor B.* Interection in key distribution schemes // Lecture Notes in Computer Science. — 1994. — 773. — P. 444–455.
2. *Blom R.* An optimal class of semmetric key generation schemes // Lecture Notes in Computer Science. — 1985. — 209. — P. 335–338.
3. *Blundo C., De Santis A., Herzberg A., Kutten S., Vaccaro U. and Yung M.* Perfectly-secure key distribution for dynamic conferens // Lecture Notes in Computer Science. — 1993. — 740. — P. 471–486.
4. *Diffie W. and Hellman M.E.* Multiuser cryptographic techniques // AFIPS Conference Proceedings. — 1976. — 45. — P 109–112.
5. *Diffie W., Van Oorschot P.C. and Wiener M.J.* Authentication and authenticated key exchanges // Desings, Codes and Cryptography. — 1992. — 2. — P. 107–125.
6. *Koblitz N.* A Course in Number Theory and Cryptography. — Springer–Verlag, New York, Inc., 1994. — P. 235.
7. *Merkle R.C.* Secure communications over insecure channels // Communications of the ACM. — 1978. — 21. — P. 294–299.
8. *Salomaa A.* Public–Key Cryptography. — Springer–Verlag, 2 ed., 1996, P. 271.
9. *Stinson D.R.* Kryptografia. Theory and Practice. — Copyright CRC Press LLC, 1995. — P. 437.