УДК 004.056.53

С.О. ГУБКА

Національний аерокосмічний університет ім. М.Є. Жуковського «ХАІ», Україна

ПІДСИСТЕМА ГЕНЕРАЦІЇ ТА РОЗПОДІЛУ КЛЮЧІВ ДЛЯ ШИФРУВАННЯ ДАНИХ У ВІДКРИТИХ КАНАЛАХ ЗВ'ЯЗКУ

Розроблено підсистему, яка об'єднує у собі функції розподілу ключів та шифрування даних. Також доопрацьовано алгоритм Діффі-Хелмана з метою підтримки ним шифрування даних. Використана комбінація алгоритмів Леманна та генератора випадкових послідовностей BBS для більш ефективної роботи алгоритму Діффі-Хелмана.

алгоритм Діффі-Хелмана, криптографія, захист інформації, клієнт-сервер

Вступ

В сучасних бізнес-процесах велику роль відіграють комп'ютерні мережі, за допомогою яких працівники компаній можуть отримати інформацію з віддалених баз даних, організовувати розподілені обчислення, обмінюватися різними документами і багато іншого. Інформація, яка передається, часто є конфіденційною, тому постає проблема її захисту, адже передача відбувається каналами, незахищеними від прослуховування. Захист інформації, що передається – цілком необхідна риса програмного забезпечення. Не важливо, скільки часу та зусиль було витрачено на контроль доступу і захист об'єктів системи, все це втрачає значення, якщо немає впевненості в наступних двох речах:

– сервер завжди повинен знати, хто його клієнт. Інакше кажучи, сервер повинен мати змогу встановити аутентичність суб'єкта, з яким він взаємодіє;

– сервер повинен гарантувати, що інформація, яка передається між ним і клієнтом, не змінюється і не переглядається сторонніми особами.

Зазвичай, серверна програма володіє великими правами в тій системі, в якій вона працює, і якщо сервер гарантує ці два моменти, то можна бути впевненим, що можливостями сервера не зловживають і буде забезпечена безпека системи.

Постановка задачі. Таким чином, треба вирішувати такі задачі як конфіденційність, цілісність та неможливість відстеження інформації.

Для вирішення поставлених задач створена криптографія – прикладна наука, яка займається розробкою, аналізом, обґрунтуванням стійкості криптографічних засобів захисту інформації від загроз з боку сторонніх осіб.

На відміну від фізичних та організаційних методів захисту інформації, під криптографічними розуміють такі технічні засоби, які використовують математичні засоби перетворення інформації, що захищається [1].

Для захисту інформації використовується безліч алгоритмів шифрування. Серед них є декілька найбільш надійних та ефективних [2], однак жоден з них не підходить повністю для вирішення поставленої задачі, тому їх треба доопрацювати. Для доопрацювання вибраний алгоритм Діффі-Хелмана, оскільки він навіть в базовому вигляді найбільш перспективний, а також має добрий потенціал для модернізації [3].

Таким чином, задачею є розробка підсистеми генерації та розподілу ключів для шифрування інформації, що пересилається по відкритих каналах зв'язку (зокрема у мережі Internet) шляхом удосконалення існуючих алгоритмів шифрування та розпо-

ділу ключів та більш ефективного поєднання допоміжних алгоритмів.

1. Вирішення задачі

1.1. Модернізація алгоритму Діффі-Хелмана. Алгоритм використовується для генерації секретного ключа, але його не можна використовувати для шифрування повідомлень.

Математика, що використовується в алгоритмі, нескладна. Спочатку Учасник 1 та Учасник 2 разом обирають велике просте число n і випадкове ціле число g. Ці два числа необов'язково зберігати в таємниці, тому Учасник 1 та Учасник 2 можуть домовитися про їх використання по несекретному каналу. Ці числа можуть навіть використовуватися групою користувачів.

Потім виконується наступний протокол:

1) Учасник 1 обирає випадкове велике ціле число x і надсилає його Учаснику 2:

$$X = g^x \bmod n;$$

2) Учасник 2 обирає випадкове велике ціле число y і надсилає його Учаснику1:

$$Y = g^y \bmod n;$$

3) Учасник 1 обчислює значення:

$$k = Y^x \bmod n;$$

4) Учасник 2 обчислює значення:

$$k` = X^y \bmod n.$$

Значення k і $k`$ обидва дорівнюють $g^{xy} \bmod n$. Стороння особа, що прослуховує канал, не зможе обчислити це значення, їй відомі тільки n, g, X, Y. Доки вона не зможе обчислити дискретний логарифм і розкрити значення x і y, то не зможе розв'язати вказану проблему. Тому k – це секретний ключ, що обчислюється Учасником 1 та Учасником 2 незалежно один від одного.

Після встановлення сеансового ключа робота алгоритму закінчена. Згідно з доопрацюванням далі починається наступна стадія аутентифікації і процес проходить наступні кроки (рис. 1):

1) Учасник 1, який є клієнтом, надсилає Учаснику 2, який є сервером, в зашифрованому вигляді своє ім'я (Повідомлення Переставлене Зашифроване (ППЗ));

2) сервер розшифровує повідомлення, бере хеш-значення імені і перевіряє базу даних на вміст такого значення. Якщо знайдено відповідний запис, то значення поля, в якому міститься результат хеш-функції над паролем, в якому виконана специфічна перестановка символів, надсилається клієнту у зашифрованому вигляді;

3) клієнт розшифровує повідомлення, бере хеш-значення власного пароля з переставленими символами, і порівнює його з отриманим повідомленням. Якщо вони ідентичні, то клієнт шифрує хеш-значення пароля без перестановки символів і надсилає його серверу;

4) сервер розшифровує повідомлення і перевіряє, чи є таке значення в базі даних. Якщо все в порядку, то встановлюється з'єднання між клієнтом і сервером.

Таке доопрацювання виключає можливість того, що стороння особа зможе мати доступ до даних на сервері. Якщо їй вдасться встановити сеансовий ключ, то не знаючи пароля, вона не зможе пройти аутентифікацію. Якщо вона перехопить повідомлення з паролем, це теж нічого не дасть, адже воно зашифроване. Перший і другий етап необхідні для того, щоб клієнт міг аутентифікувати сервер, адже клієнт повинен знати, кому він надсилає свій пароль. На третьому і четвертому етапі уже клієнт підтверджує свою достовірність перед сервером, після чого встановлюється з'єднання і може відбуватися обмін даними. Специфічна перестановка символів означає, що параметром для хеш-функції є пароль з символами, переставленими в зворотному порядку. В базі даних клієнтів поруч з іменем містяться і два варіанти хеш-значення пароля.

В доопрацьованому вигляді алгоритм Діффі-Хелмана поєднує простоту і ефективність з можливістю взаємної аутентифікації учасників, що робить його ідеальним для вирішення поставленої задачі.

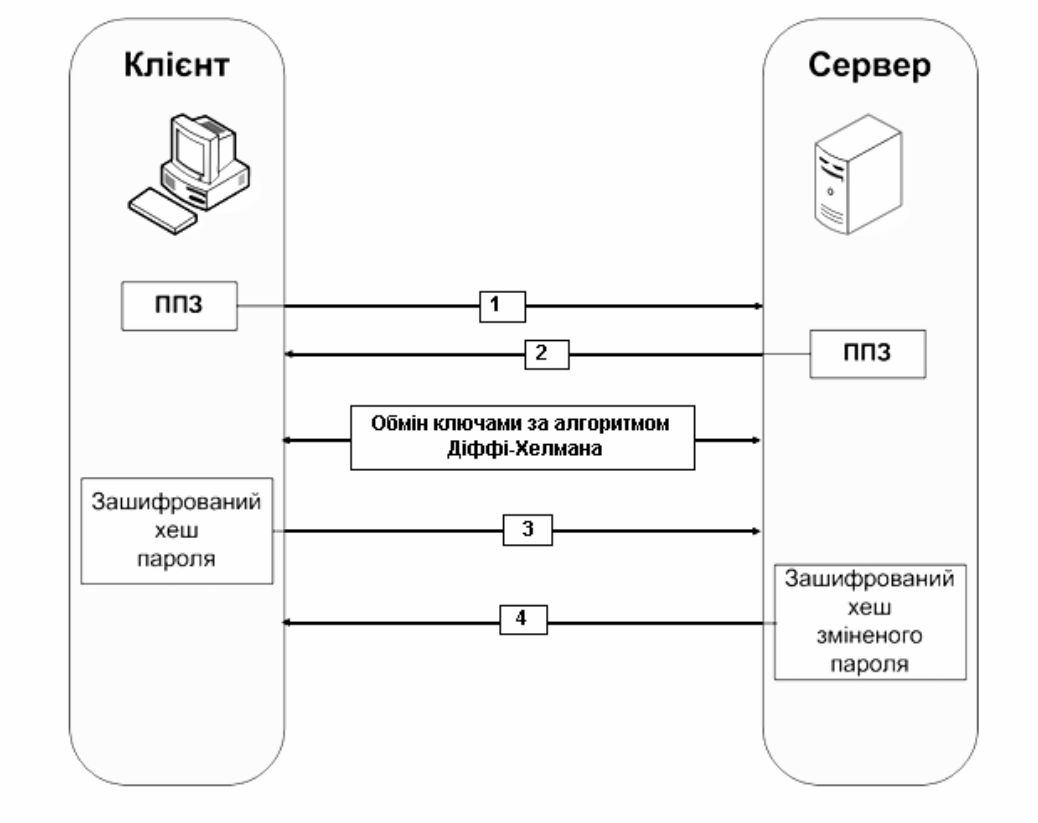

Рис. 1. Робота підсистеми з доопрацьованим алгоритмом Діффі-Хелмана

1.2. Генератор псевдовипадкових послідовностей BBS (Blum-Blum-Shub). Для генерації відкритих і закритих ключів, які використовуються в алгоритмі Діффі-Хелмана, потрібна генерація випадкових чисел [4]. Простий і найбільш ефективний генератор – це генератор BBS, який іноді називають генератором квадратичних лишків. Теорія генератора BBS використовує квадратичні лишки по модулю *n*.

Спочатку знайдемо два простих числа, *p* і *q*. Добутком цих чисел *n* буде число Блюма. Виберемо інше випадкове ціле число *x*, взаємно просте з *n*. Обчислимо:

$$x_0 = x^2 \bmod n.$$

Це початкове значення генератора. Тепер можна почати обчислювати біти. Псевдовипадковим бітом з номером і буде молодший значущий біт x_i, де

$$x_i = x_{i-1}^2 \bmod n.$$

Характерною властивістю цього генератора є те, що для отримання і-го біту не потрібно обчислювати (*i*–1) попередні біти. Якщо відомі значення *p* і *q*, можна обчислити *i*-й біт напряму.

Далі, x_0 – це молодший значущий біт x_i, де

$$x_i = x_0^{2^i \bmod ((p-1)(q-1))}.$$

Дана властивість означає, що цей криптографічно стійкий генератор псевдовипадкових чисел можна використовувати в якості потокової криптосистеми для файлу з вільним доступом. Стійкість цієї схеми ґрунтується на складності розкладення *n* на множники. Можна навіть опублікувати *n* так, щоб хто завгодно міг генерувати біти за допомогою генератора. Але доки криптоаналітик не зуміє розкласти *n* на множники, він ніколи не зможе ні передбачити вихід генератора, ні навіть стверджувати щось на зразок: «Наступний біт з ймовірністю 51% буде одиницею».

Більше того, генератор BBS непередбачуваний вправо. Це означає, що, отримавши послідовність від генератора, криптоаналітик не зможе передбачи-

ти ні наступний, ні попередній біт послідовності. Це зумовлено не надійністю складного генератора біт, а математичною складністю розкладання n на множники.

Цей алгоритм повільний, але є способи його прискорення. Виявляється, що в якості псевдовипадкових біт, окрім молодших біт, можна використовувати ще кілька біт кожного x_i. Якщо n – довжина x_i, можна використовувати $\log_2 n$ молодших значущих біт x_i. Генератор BBS порівняно повільний і не використовується для потокових шифрів. Але для високонадійних додатків, наприклад, для генерації ключів, цей генератор кращий від багатьох інших.

1.3. Перевірка чисел на простоту за тестом Леманна. Окрім генерації псевдовипадкових чисел потрібно упевнитися, що вони дійсно є простими. Для цього потрібно провести це один тест. Леманн розробив алгоритм імовірнісного тестування чисел на простоту. Ось послідовність дій при перевірці простоти числа p:

1) вибирається випадкове число a, менше p;

2) обчислюється $a^{(p-1)/2} \bmod p$;

3) якщо $a^{(p-1)/2} \neq 1$ або $a^{(p-1)/2} \neq p-1$, то число p напевно не просте;

4) якщо $a^{(p-1)/2} \equiv 1$ або $a^{(p-1)/2} \equiv p-1$, то ймовірність того, що число p не просте, не перевищує 50 %.

Ймовірність того, що випадкове число a стане свідком складної природи числа p (число p буде складатися із простих множників) складатиме не менше як 50 %. Якщо повторити тест t разів, і результат завжди буде дорівнювати 1 або $p-1$, то можна стверджувати, що p – просте число з ймовірністю $1/2^t$.

Висновки

Розроблена підсистема дозволяє суттєво покращити захист даних, що передаються по відкритим каналам зв'язку, за рахунок введення подвійного захисту. Також гарантована якість вхідних даних для шифрування за рахунок використання алгоритму BBS та тесту Леманна.

Література

1. Шнайер Б. Прикладная криптография. – М.: ТРИУМФ, 2003. – 820 с.

2. Домашев А.В. Программирование алгоритмов защиты информации. – М.: Нолидж, 2000. – 288 с.

3. Барсуков В.С., Водолазкий В.В. Современные технологии безопасности. – М.: Нолидж, 2001. – 453 с.

4. Программно-аппаратные средства обеспечения информационной безопасности. Защита программ и данных: Учебное пособие для вузов / П.Ю. Белкин и др. – М.: Радио и связь, 1999. – 168 с.

Надійшла до редакції 26.11.2007

Рецензент: д-р техн. наук, проф. А.Ю. Соколов, Національний аерокосмічний університет ім. М.Є. Жуковського «ХАІ», Харків.