УДК 378 + 681.324

І73

Друкується за рішенням Ученої ради
Вінницького державного технічного університету
Міністерства освіти та науки України

І73 **"ІНТЕРНЕТ, ОСВІТА, НАУКА - 2000", друга міжнародна конференція ІОН –2000,** 10 – 12 жовтня, 2000. Збірник матеріалів конференції. – Вінниця: УНІВЕРСУМ – Вінниця, 2000 – 343 с.

ISBN 966-641-000-1

Друга міжнародна конференція "ІНТЕРНЕТ – НАУКА – ОСВІТА - 2000" (ІОН –2000. Нові інформаційні і комп'ютерні технології в освіті та науці) присвячена обговоренню питань застосування в освіті та наукових дослідженнях нових інформаційних технологій, що спираються на можливості Інтернет.

Доповіді у збірнику згруповані по секціях, відповідно до основних напрямків конференції:

A. Освітянські та науково-інформаційні комп'ютерні мережі: проблеми, рішення, перспективи

B. Інтернет та інформаційні технології в освіті

C. Методологічні та практичні аспекти дистанційної освіти

D. Інтернет у бібліотечній справі, економічних та наукових дослідженнях

Всі матеріали доповідей представлені також на Web-сайті конференції (http:/www.vstu.vinnica.ua/ies2000), що містить електронну версію даного збірника (з можливістю пошуку по ключових словах у назвах доповідей, їхніх текстах, прізвищах авторів), доповнену наданими авторами перекладами деяких доповідей, і базу даних з відомостями про учасників конференції.

Тексти доповідей друкуються в авторській редакції.

Відповідальний за випуск В. В. Грабко

ISBN 966-641-000-1 УДК 378 + 681.324

Підготовлено до друку: *В.В.Грабко, В.І.Месюра, І.Р.Арсенюк, Г.О.Лосев, О.В.Лосева, О.М.Хошаба*

ШИФРУВАННЯ І СТИСНЕННЯ ІНФОРМАЦІЇ В КОМП'ЮТЕРНИХ МЕРЕЖАХ

В.П. Майданюк, А.В. Шевченко, С.А. Піддубчак

Вінницький державний технічний університет, Вінниця, Україна
Тел. 44-04-86, E-mail: maydan@vstu.vinnica.ua

Abstract

The article deals with the problems of the encryption and data compression. The encryption and data compression are tie together. In booth case frequency distribution of characters in output data is changed. Data compression before the encryption troubles cryptanalysis (decryption) of ciphertext. The analysis of algorithms data compression and encryption shows, that at the same time with data compression possible the data encryption. It is allows to decrease computational expenses and increase the security of the cryptosystem.

1. Вступ

Широке застосування компьютерної техніки в різних сферах діяльності, бурхливий розвиток комп'ютерних мереж робить все більш актуальними питання захисту інформації від несанкціонованого доступу, оскільки наслідки цього можуть бути непередбачувані.

Незважаючи на те, що захист інформації в комп'ютерних мережах пов'язаний з цілим рядом комплексних заходів, як чисто технічних, так і організаційних, без шифрування інформації неможливо побудувати надійну систему її захисту.

Наукова криптологія (cryptos – таємний, logos – слово) бере початок з роботи К.Шеннона "Теорія зв'язку в секретних системах" (1949р.), в якій було показано, що для деякого випадкового шифру кількість знаків шифротексту, отримавши який криптоаналітик при необхідних обчислювальних ресурсах зможе відновити ключ (тобто розкрити шифр), становить:

$$n = \frac{H(Z)}{r \log N}, \tag{1}$$

де H(Z) – ентропія ключа, r – надлишковість відкритого тексту, N-обсяг алфавіту.

З виразу (1) видно, що зниження надлишковості (стиснення даних) може значно збільшити криптостійкість навіть для коротких ключів [1].

Однак, існуючі алгоритми і стандарти шифрування не передбачають зниження надлишковості одночасно з виконанням шифрування. Тобто спочатку виконується стиснення даних, а потім їх шифрування.

Метою даної роботи є аналіз можливостей одночасного стиснення (compression) і шифрування інформації, що може значно знизити обчислювальні витрати.

2. Шифрування інформації

Розглянемо деякі традиційні методи криптографії.

Найбільш простим в реалізації є методи прямої підстановки та перестановки. В прямих підстановках кожний знак початкового тексту замінюється одним або декількома іншими знаками. Розрізняють:

- моноалфавітну підстановку
- багатоалфавітну підстановку.

При моноалфавітній підстановці встановлюється відповідність між кожним знаком a_i алфавіта повідомлення А і відповідного знаку зашифрованого тексту. Всі методи моноалфавітної підстановки можливо представити як числові перетворення букв початкового тексту у відповідності з виразом:

$$C = (ap + S) \bmod k, \tag{2}$$

де a - десятковий коефіцієнт, S - коефіцієнт зсуву, k - розмір алфавіту, p - символ початкового тексту [2].

Недоліком моноалфавітної підстановки є низька криптостійкість, оскільки в шифрованому повідомленні зберігається частотний розподіл символів початкового тексту.

Ці недоліки можна зменшити за рахунок використання багатоалфавітної підстановки. При n-алфавітній підстановці знак m_1 з початкового повідомлення замінюється знаком з алфавіту B_1, m_2 – відповідно з алфавіту B_2, …, m_n – знаком з алфавіту B_n, m_{n+1} – знову з алфавіту B_1 і т.д. Ефект використання багатоалфавітної підстановки полягає в тому, що забезпечується маскування природної

частотної статистики початкової мови повідомлення, оскільки конкретний знак з алфавіту А може бути перетворений в декілька різних знаків шифрувального алфавіту В.

Знаки початкового тексту можливо також переставляти у відповідності з деяким правилом. Недолік перестановок той самий, що і моноалфавітної підстановки.

Ефективність шифрування можливо підвищити, використовуючи комбінацію перестановок і підстановок.

Значно більшу криптостійкість забезпечує шифрування з використанням генератора псевдовипадкових чисел (ПВЧ). Найбільш широке застосування знаходять лінійні конгруентні генератори ПВЧ. Цей генератор формує послідовність псевдовипадкових чисел T_1, T_2,…, T_m, використовуючи співвідношення:

$$T_{i+1} = (aT_i + c) \bmod m, \qquad (3)$$

де a і c - константи, T_0 – початкова величина, вибрана в якості породжуючого числа.

Період повторення псевдовипадкових чисел залежить від вибраних значень a і c. Лінійний конгруентний генератор має максимальну довжину m тільки тоді, поки c - непарне, а $a \bmod 4 = 1$.

При шифруванні псевдовипадкові числа, отримані з виразу (3), об'єднуються деяким чином з відповідним обсягом тексту, але так, щоб можна було відновити початковий текст. Наприклад, з використанням додавання по модулю 2 (накладання гами-ключа на початковий текст). Якщо періодичність генератора більша довжини всіх посланих повідомлень початкового тексту і якщо криптоаналітику невідомий ніякий початковий текст, то шифр теоретично неможливо відкрити [2].

Такі відомі схеми кодування як DES, RSA [3], алгоритм криптографічного перетворення згідно ГОСТ 28147-89 [7] повністю або частково являються комбінацією методів, розглянутих вище (не розглядаються в даній роботі).

3.Одночасне стиснення і шифрування даних

Основною метою кодування або стиснення даних є перетворення вхідного потоку символів в потік бітів мінімальної довжини. Це досягається за рахунок зменшення надлишковості вихідного потоку. При цьому символи, які найбільш часто зустрічаються, представляються короткими кодовими комбінаціями, за рахунок чого і досягається стиснення. Однак, при стисненні даних деякими методами існує можливість одночасного шифрування або без додаткових обчислювальних затрат, або з низькими затратами. Оскільки в більшості випадків, файли, які передаються по комп'ютерним мережах або зберігаються в пам'яті, представлені в стиснутому вигляді, то було б доцільно надати можливість користувачам виконувати при стисненні файлів і їх шифрування.

Розглянемо з цієї точки зору деякі методи стиснення інформації.

Достатньо простий і ефективний метод стиснення даних з невідомим розподілом частот, відомий як кодування за ступенем новизни, був відкритий у 1980 р. Рябко.

При стисненні інформації цим методом можливе одночасне виконання шифрування методом багатоалфавітної підстановки без додаткових затрат, або з застосуванням інших методів і їх комбінацій з незначними додатковими затратами. Суть цього методу полягає в наступному [4]. Нехай передається повідомлення:

ω=ABCDDAAACAB

в алфавіті AL={A,B,C,D}. При появі в слові ω чергової букви "a" по лінії зв'язку передається монотонний код номера позиції, яку займає в даний момент символ "a" в списку, а символ "a" переміщується на початок списку. Таким чином символи, які часто зустрічаються, завжди будуть знаходитись на початку списку і відповідно представлятись короткими кодовими комбінаціями. Порядок кодування за ступенем новизни демонструє табл. 1.

Таблиця 1. Кодування за ступенем новизни

Вхід	Вихід	Алфавіт
A	0_2	{ABCD}
B	10_2	{ABCD}
C	110_2	{BACD}
D	111_2	{CBAD}
D	0_2	{DCBA}
A	111_2	{DCBA}
A	0_2	{ADCB}

Якщо список алфавіту "AL" згенерувати, наприклад, використовуючи генератор ПВЧ, то вихідна послідовність стане іншою. Не знаючи початковий стан алфавіту, неможливо дешифрувати повідомлення. При цьому реалізується шифрування методом багатоалфавітної підстановки і стиснення даних одночасно. Для одночасного стиснення і шифрування даних методом багатоалфавітної підстановки може використовуватись така схема:

- згідно з формулою (3) генерується алфавіт повідомлення. Оскільки символи в комп'ютерних системах представлені 8-и бітовими комбінаціями, то m=256, а константи a, c і породжуюче число T_0 можна вводити в якості ключа шифра.
- виконується стиснення згідно приведеного методу

Додаткові затрати відсутні, оскільки генерація символів алфавіту виконується в будь-якому випадку.

При незначних додаткових затратах можливо застосувати комбінацію методів, яка включає два основні етапи:

Етап1. Кодування за ступенем новизни для стиснення і шифрування методом багатоалфавітної підстановки.

Етап2. Шифрування за допомогою додаткового генератора ПВЧ з m рівним довжині початкового файлу результатів етапу 1. Цей або інший генератор ПВЧ може бути використаний також для виконання перестановок, оскільки він генерує всі числа в межах довжини початкового файлу. Якщо в стисненні даних немає потреби, то ця перестановка дозволить замаскувати стиснені дані в межах файлу такого ж розміру як і початковий з попередньо записаною випадковою інформацією.

Аналіз таких відомих алгоритмів стиснення інформації як кодування Хаффмана та LZW [5] показує також, що їх легко адаптувати до одночасного виконання стиснення і шифрування інформації. А при деяких додаткових обчисленнях згідно формули (3), ще і шифрування методом ПВЧ. Для цього достатньо лише сформувати таблицю перекодування кожного зчитаного з пам'яті символу. Ця таблиця може бути сформована, наприклад, за допомогою генератора ПВЧ (3) з m=256. При цьому зчитаний з початкового файлу символ даних задає позицію в таблиці перекодування, з якої вибирається код підстановки.

4. Висновки

1. Аналіз відомих алгоритмів стиснення даних показує, що вони дозволяють виконувати одночасно з стисненням і шифрування інформації без або при незначних додаткових затратах.
2. Для підвищення криптостійкості найбільш доцільно додатково до стиснення-шифрування використовувати генератори ПВЧ для накладання гами ключа на стиснуті дані і виконання перестановок у вихідних даних, оскільки додаткові обчилювальні затрати при цьому незначні.

Література:

1. Новосельский А. Алгоритмы шифрования // Компьютеры + программы 1996-№5.- С.70-77
2. Хоффман Л. Современные методы защиты информации: Пер. с англ. -М.: Сов. радио, 1980, 264с.
3. Защита информации в персональных ЭВМ/ Спесивцев А.В., Вегнер В.А., Крутяков А.Ю. и др.-М.: Радио и связь, НП «Весна», 1992.
4. Кричевский Р.Е, Сжатие и поиск информации. –М.: Радио и связь, 1989.-168с.
5. Кохманюк Д. Сжатие данных: как это делается // Компьютеры+программы,1995.-№5. -С.18-19
6. Гончаров С. Криптография: ломайте головы… // Компьютер-пресс, 1998.-№6.-С.59-61
7. ГОСТ 28147-89. Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования.