УДК 004.41:004.056.5

Д.В. Ратов

РОЗРОБКА МЕТОДУ ТА ПРОГРАМНОГО ЗАСОБУ СТИСНЕННЯ ТА ШИФРУВАННЯ ІНФОРМАЦІЇ

Ключові слова: інформаційна технологія, стиснення даних, коефіцієнт компресії, циклічний надлишковий код, архіватор, обробка потокових даних.

Keywords: information technology, data compression, compression ratio, cyclic redundancy code, archiver, stream data processing.

Вступ

Всебічне впровадження і розвиток інформаційних технологій [1, 2] у сучасному інформаційно-технологічному світі призвело до необхідності архівування та шифрування даних. Архівація — це стиснення, упаковка інформації з метою її більш раціонального розміщення на зовнішніх носіях [3, 4]. Необхідність архівації пов'язана з резервним копіюванням інформації з метою збереження програмного забезпечення комп'ютера і захисту його від псування і знищення (умисного, випадкового або під дією комп'ютерного вірусу). Для запобігання втрати інформації необхідно мати резервні копії програм і файлів.

Архіватори — це програми, які реалізують процес архівації та дозволяють створювати архіви та розпаковувати їх. Архів, в основному, використовують для об'єднання великої кількості файлів, полегшення їх розміщення та передачі через Інтернет, але це не єдине, для чого потрібна архівація даних. За допомогою архівації можливе збереження будь-яких важливих файлів. Архів не може бути заражений вірусом завдяки своєму внутрішньому захисту від доступу, крім того, в результаті архівації зменшується розмір розміщених у ньому файлів, що робить його зручним для резервного копіювання на інформаційні носії. Тому це — хороший захист електронних даних.

В даний час саме розвитку програм-архіваторів приділено особливу увагу. Різного роду розробки стали з'являтися не лише в державних службах, але і у приватних осіб. Саме сьогодні активно ведуться дослідження різних способів архівації, модернізуються старі методи та з'являються нові. Інтерес до даної сфери триватиме до тих пір, поки йде розвиток інформаційної інфраструктури глобальних мереж Інтернет [5, 6]. Світ переходить на електронний документообіг [7], і саме архіви допоможуть у збереженні та захисті цих даних.

Сьогодні доступні як носії інформації великого обсягу, так і високошвидкісні канали передачі даних. Проте одночасно з цим ростуть і обсяги інформації, що передаються. Іноді стиснення не тільки корисне, але й необхідне. Наприклад, це може бути [8]

- пересилання документів електронною поштою (особливо великих обсягів документів з використанням мобільних пристроїв) [9, 10];
- потреба в економії трафіку при публікації документів на сайтах;
- економія дискового простору.

Методологія вирішення проблем стиснення даних

В основі всіх методів стиснення лежить принцип: якщо при збереженні блоку даних для елементів, що часто використовуються, задіяти короткі коди, а для елементів, що рідко використовуються, — довгі, то потрібен менший об'єм пам'яті, ніж коли всі елементи мають коди однакової довжини.

Точний зв'язок між ймовірностями і кодами встановлено в теоремі Шеннона про кодування джерела [11], яка свідчить, що елемент S_i, імовірність появи якого дорівнює $p(S_i)$, найвигідніше представляти $-\log_2 p(S_i)$ бітами. Якщо при кодуванні розмір кодів завжди дорівнює $-\log_2 p(S_i)$ бітів, то в цьому випадку довжина закодованої послідовності буде мінімальною для всіх можливих способів кодування. Коли розподіл ймовірностей $F = \{p(S_i)\}$ незмінний і вірогідність появи елементів умовна, то можна визначити середню довжину кодів як зважене середнє:

$$H = -\sum_i p(S_i) \cdot \log_2 p(S_i).$$

Це значення є ентропією розподілу ймовірностей F або ентропією джерела в заданий момент часу. Зазвичай ймовірність появи елемента є умовною, тобто залежить від якоїсь події. У цьому випадку при кодуванні чергового елемента S_i розподіл ймовірностей F приймає одне з можливих значень F_k, тобто $F = F_k$ і, відповідно, $H = H_k$. Можна сказати, що джерело знаходиться в стані k, якому відповідає набір ймовірностей $p_k(S_i)$ створення всіх можливих елементів S_i. Тому середню довжину кодів можна розрахувати за формулою [11]:

$$H = -\sum_k P_k \cdot H_p = -\sum_{k,\, i} P_k \cdot p_k(\mathrm{S}_i), \qquad (1)$$

де P_k — імовірність того, що F отримає k-е значення (ймовірність знаходження джерела в стані k).

Отже, якщо відомо розподіл ймовірностей елементів, що генеруються джерелом, то дані можна представити найбільш компактно, при цьому середню довжину кодів може бути обчислено за формулою (1).

Здебільшого справжня структура джерела невідома, тому необхідно будувати модель джерела, яка дозволила б у кожній позиції вхідної послідовності оцінити ймовірність $p(S_i)$ появи кожного елемента S_i алфавіту. У цьому випадку оперуємо оцінкою $q(S_i)$ ймовірності елемента S_i.

За допомогою методів стиснення можна будувати модель джерела адаптивно у міру обробки потоку даних або використовувати фіксовану модель, що створено на основі апріорних уявлень про природу типових даних, які вимагають стиснення.

Процес моделювання може бути або явним, або прихованим. Ймовірності елементів можуть використовуватися в методі як явно, так і неявно. Але стиснення завжди досягається шляхом усунення статистичної надлишковості при поданні інформації.

Основною характеристикою алгоритмів стиснення є коефіцієнт стиснення [11], що визначається як відношення різниці обсягу нестиснених даних і стиснених до обсягу вхідних даних, тобто

$$k = \left(1 - \frac{S_c}{S_0}\right) \cdot 100\,\%,$$

де k — коефіцієнт стиснення, S_0 — обсяг вхідних даних, а S_c — обсяг стиснених. Таким чином, чим вище коефіцієнт стиснення, тим алгоритм ефективніше.

Алгоритмічні підходи кодування з мінімальною надлишковістю та упаковки інформації із застосуванням словника

В основі будь-якого способу стиснення лежить модель джерела даних, або, точніше, модель надмірності [12]. Тобто для стиснення даних використовуються деякі апріорні відомості про те, якого типу дані стискаються. Не володіючи такими відомостями про джерело, неможливо зробити ніяких припущень щодо перетворення, яке дозволило б зменшити обсяг повідомлення. Модель надмірності може бути статичною, незмінною при стисненні повідомлення або будуватися чи параметризуватися на етапі стиснення (і відновлення). Адаптивні методи дозволяють на основі вхідних даних змінювати модель надмірності інформації. Неадаптивними є зазвичай вузькоспеціалізовані алгоритми, що використовуються для роботи з даними, які мають певні добре окреслені і незмінні характеристики. Здебільшого універсальні алгоритми тією чи іншою мірою є адаптивними.

Всі методи стиснення даних діляться на два основні класи [13]: без втрат (lossless) та з втратами (lossy) (рис. 1).

Метод стиснення без втрат — це коли при відновленні даних отримуємо точну копію вихідних даних. Даний метод зазвичай використовується для передачі і зберігання текстових даних, комп'ютерних програм, рідше — для скорочення обсягу аудіо та відео, цифрових фотографій і т.п., коли спотворення неприпустиме або небажане.

Метод стиснення з втратами (характерний приклад — файли jpeg), що має значно більшу ефективність, ніж стиснення без втрат, зазвичай застосовується для скорочення обсягу аудіо, відео та цифрових фотографій в тих випадках, коли таке скорочення є пріоритетним, а повна відповідність вихідних і відновлених даних не потрібна.

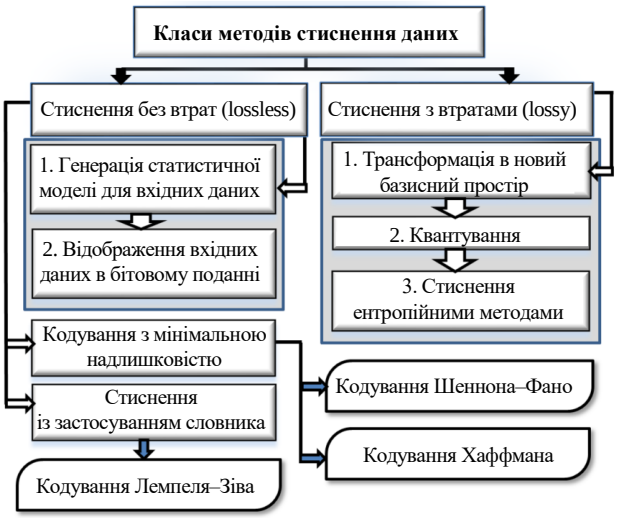


Рис. 1

До алгоритмів стиснення без втрат (рис. 1) відноситься:

1) кодування з мінімальною надлишковістю (minimum redundancy coding);

2) стиснення із застосуванням словника (dictionary compression).

Кодування з мінімальною надлишковістю — це метод кодування байтів, при якому байти, які частіше зустрічаються, кодуються меншою кількістю бітів, ніж ті, які зустрічаються рідше.

До класу кодування з мінімальною надлишковістю відноситься алгоритм стиснення за методом Шеннона–Фано, де аналізуються вхідні дані, на основі яких будується бінарне дерево мінімального кодування. При використанні цього дерева повторно виконується зчитування і кодування вхідних даних.

Алгоритм кодування Хаффмана [14] дуже схожий на алгоритм стиснення Шеннона–Фано, але виявився більш ефективним. Це обумовлено тим, що алгоритм Хаффмана математично гарантовано створює найменший за розміром код для кожного з символів вихідних даних. Аналогічно алгоритму Шеннона–Фано потрібно побудувати бінарне дерево, яке також буде префіксним, де всі дані зберігаються в листі. Але на відміну від алгоритму Шеннона–Фано, який є спадним, на цей раз побудова дерева буде здійснюватися знизу вгору. Спочатку переглядаються вхідні дані, підраховується кількість появ значень кожного байта, як і при використанні алгоритму Шеннона–Фано. Після того як створено таблицю частот появи символів, можлива побудова дерева.

При стисненні із застосуванням словника дані розбиваються на великі фрагменти символів (лексеми). Потім застосовується алгоритм кодування лексем з певною мінімальною кількістю бітів. До класу стиснення із застосуванням словника відноситься метод Лемпеля–Зіва (LZW) [15], при якому використовується підхід, заснований не на частоті появи байтів в тексті, а на повторенні слів або їх частин в запакованому тексті. Якщо слово або його фрагмент повторюється, то виконується їх заміна посиланням на попереднє слово. Архіватор, побудований за принципом такого методу, працює в один прохід і створює лише словник, заснований на повторюваних ділянках тексту. Ефективність стиснення тут залежить лише від розмірів текстового файлу і числа повторень.

LZW використовується в деяких форматах графічних файлів (наприклад, GIF). На основі метода Лемпеля–Зіва створювалося безліч архіваторів. В сучасних архіваторах здебільшого також застосовується цей алгоритм спільно з методом Хаффмана для стиснення текстових файлів.

Дослідження та програмна реалізація стиснення та шифрування інформації

Програмним засобом стиснення та шифрування інформації є розроблюваний програмний комплекс Archiver. Механізм роботи архіватора засновано на створенні і обробці потокових даних. Ядро архіватора — функції стиснення і розпаковування файлів методом Лемпеля–Зіва (LZW). Програмний алгоритм LZW полягає в тому, що в процесі обробки вихідного файлу програма формує словник, слова в якому є частинами коду вихідного файлу. При формуванні упакованого файлу в нього замість довгої послідовності байтів записується тільки ідентифікатор відповідного слова зі словника (його номер) або сам символ, якщо потрібного слова немає в словнику. При цьому отримуємо зменшення даних, оскільки дана послідовність (слово зі словника) може зустрічатися у файлі декілька разів, а довжина ідентифікатора слова значно коротша самого слова. При збереженні стисненого файлу не потрібно зберігати словник, тому що даний алгоритм дозволяє створити словник у процесі розпаковування стисненого файлу.

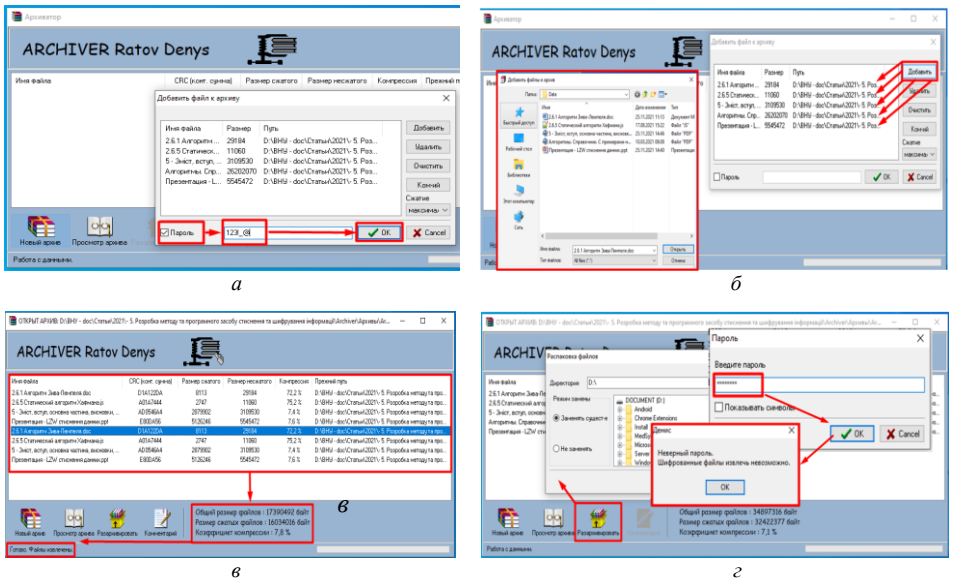
При роботі Archiver як метод і засіб захисту інформації було задіяно поліалфавітну підстановку (шифр Віжінера [16]). Для підвищення стійкості шифру ключ

(що використовується також як пароль до архіву) застосовується при заміні відповідних байтів стисненого файлу і не зберігається в тілі файлу архіву; зберігається лише ознака наявності такого ключа в заголовку архіву.

Як спосіб цифрової ідентифікації послідовності даних було використано алгоритм обчислення контрольної суми (CRC, Cyclic Redundancy Code, циклічний надлишковий код), який полягає в обчисленні контрольного значення циклічного надлишкового коду упакованих файлів.

Рис. 2

Розроблюваний в інтегрованому середовищі розробки додатків Embarcadero RAD Studio XE8 програмний комплекс Archiver складається з шести модулів (рис. 2). Модуль MainUnit формує головне вікно програми, її інтерфейс і зв'язок з іншими модулями. Основну роботу зі стиснення та розпакування файлів реалізовано в модулі BasicZip. Модулі PasswordUnit, AddingUnit, EditCommentUnit, ExtractingUnit відповідають за внесення пароля, читання файлів з диска і їх запис, редагування коментарів, вибір архіву для розпакування.



а *б* *в* *г*

Рис. 3

На рис. 3 представлено етапи роботи програми Archiver.

1. Створення архіву та додавання групи файлів до нього (рис. 3, *а*). Після процесу стиснення програма відображає перелік архівних файлів, розмір кожного файлу до стиснення та після, ступінь стиснення, циклічний надлишковий код і шлях до файлу.

2. Вибір ступеня стиснення файлів і введення паролю (рис. 3, *б*).

3. Введення паролю для розархівування і дешифрування (рис. 3, *в*). Якщо введено невірний пароль — з'являється відповідне повідомлення. Якщо кількість спроб введення невірного паролю перевищує задане число, то архів блокується без можливості дешифрування.

4. При введенні коректного паролю файли архіву успішно розпаковуються (рис. 3, *г*). Відображається відповідне повідомлення та виконується декодування і розархівування файлів архіву. Для перевірки та ідентифікації розпакованих файлів використовується алгоритм обчислення контрольної суми (CRC).

Для випробування програмного забезпечення розробленого архіватора Archiver проаналізовано ступінь стиснення для різних файлів.

	Розмір архіву
Початковий розмір каталогу з даними	20 278 Кбайт
WinRar	13 326 Кбайт
Zip	14 631 Кбайт
Archiver	14 622 Кбайт

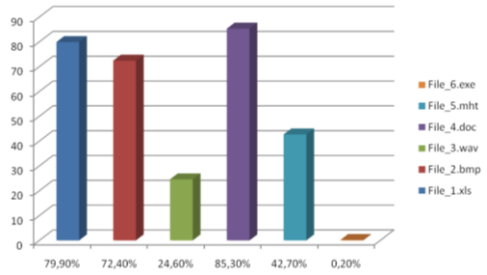


Рис. 4

Для процесу стиснення взято довільні файли розширень: *.doc, *.exe, *.wav, *.xls, *.bmp, *.mht, тобто графічного, текстового, мультимедійного та інших форматів.

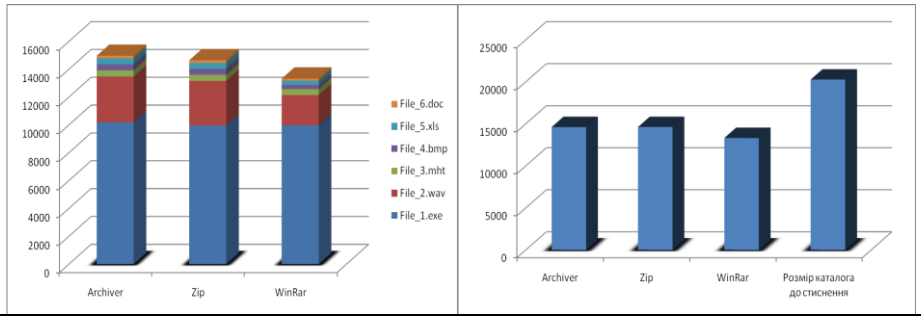


Рис. 5

На рис. 4, 5 представлено результати стиснення програмним комплексом Archiver та іншими архіваторами WinRar, ZIP. Для rar- і zip-формату виставлено параметр звичайного стиснення, який застосовується і в програмі Archiver.

Висновок

У рамках дослідження розроблено програмний спосіб шифрування даних, за допомогою якого можливе створення зашифрованого архіву з паролем, що ще в більшій мірі сприяє захищенню інформації від атак та несанкціонованого використання.

У процесі створення програмного засобу шифрування та архівування досліджено алгоритм стиснення з мінімальною надлишковістю та алгоритм стиснення із застосуванням словника. Як вхідні дані використано файли графічного, текстового, мультимедійного та інших форматів. Практичне застосування алгоритму стиснення інформації без втрат даних продемонстровано при розробці архіватора з шифруванням Archiver.

Отриманий програмний засіб шифрування та архівування використано в одному з модулів програмного комплексу «Дипломи СНУ v.2.6.1», розробленому в «Східноукраїнському Національному університеті імені Володимира Даля» [4, 17].

Цей комплекс призначено для створення в університеті єдиного реєстру дипломів, автоматизації створення файлів-дипломів про вищу освіту у багатофункціональному графічному редакторі Adobe Photoshop. Усі дані для аналізу й формування дипломів контролер експортує з параметрів відповідних xml-файлів, завантажених з єдиної державної бази освіти в стиснених файлах zip-архівів. Розроблений програмний засіб виконує процес розархівування і отримання xml-файлів з параметрами для подальшої роботи комплексу «Дипломи СНУ v.2.6.1».

Отримані в роботі результати демонструють перспективність вивчення проблематики даної теми, безсумнівну корисність використання методів стиснення й шифрування, які підвищують захист інформації. Розроблена програма архівування не лише досягла поставленої мети, а й показала результати, що конкурують з результатами провідних архіваторів, таких як WinRar і Zip. Коефіцієнт стиснення файлів для WinRar склав близько 34,28 %, для Zip — 27,84 %, а для Archiver — 27,89 %. Тобто розроблений архіватор Archiver в проведених дослідженнях спрацював ефективніше, ніж Zip, і трохи гірше, ніж Rar.

Д.В. Ратов

РОЗРОБКА МЕТОДУ ТА ПРОГРАМНОГО ЗАСОБУ СТИСНЕННЯ ТА ШИФРУВАННЯ ІНФОРМАЦІЇ

Проведено дослідження предметної галузі стиснення інформації без втрат та з втратами та розглянуто алгоритми стиснення даних з мінімальною надмірністю (кодування Шеннона–Фано, кодування Хаффмана) та стиснення із застосуванням словника (кодування Лемпеля–Зіва). У процесі роботи використано теоретичні основи стиснення даних, проведено дослідження різних способів стиснення інформації, виявлено найкращі способи архівації з шифруванням та зберігання різноманітних даних. Метод архівації даних у роботі використано з метою безпечного та раціонального розміщення на зовнішніх носіях інформації та її захисту від навмисного чи випадкового знищення чи втрати. В інтегрованому середовищі розробки Embarcadero RAD Studio XE8 виконано програмний комплекс архіватора з кодовим захистом інформації. Механізм роботи архіватора засновано на створенні та обробці потокових даних. Ядром архіватора є функції стиснення та розпакування файлів методом Лемпеля–Зіва. Як метод і засіб захисту інформації в архіві використано поліалфавітну підстановку (шифр Віжінера). Результати роботи, зокрема розроблене програмне забезпечення, можуть бути практично використані при архівному зберіганні захищеної інформації. Механізм архівування та шифрування даних може бути використано у системах передачі інформації з метою зменшення трафіку в мережі та забезпечення захисту даних. Отриманий програмний засіб шифрування та архівування використано у модулі програмного комплексу «Дипломи СНУ v.2.6.1», який розроблено у «Східноукраїнському Національному університеті імені Володимира Даля». Цей комплекс призначений для створення в університеті єдиного реєстру дипломів, автоматизації створення файлів-дипломів про вищу освіту у багатофункціональному графічному редакторі Adobe Photoshop. Усі дані для аналізу та формування дипломів контролер експортує з параметрів відповідних XML-файлів, завантажених з єдиної державної бази освіти у стиснених файлах zip-архівів. Розроблений модуль виконує процес розархівування та отримання XML-файлів із параметрами для подальшої роботи комплексу «Дипломи СНУ v.2.6.1».

D.V. Ratov

DEVELOPMENT OF METHOD AND SOFTWARE FOR COMPRESSION AND ENCRYPTION OF INFORMATION

Researches of the subject area of lossless information compression and with data loss are carried out and data compression algorithms with minimal redundancy are con-

sidered: Shannon-Fano coding, Huffman coding and compression using a dictionary: Lempel-Ziv coding. In the course of the work, the theoretical foundations of data compression were used, studies of various methods of data compression were carried out, the best methods of archiving with encryption and storage of various kinds of data were identified. The method of archiving data in the work is used for the purpose of safe and rational placement of information on external media and its protection from deliberate or accidental destruction or loss. In the Embarcadero RAD Studio XE8 integrated development environment, a software package for an archiver with code protection of information has been developed. The archiver's mechanism of operation is based on the creation and processing of streaming data. The core of the archiver is the function of compressing and decompressing files using the Lempel-Ziv method. As a method and means of protecting information in the archive, polyalphabetic substitution (Viziner cipher) was used. The results of the work, in particular, the developed software can be practically used for archival storage of protected information; the mechanism of data archiving and encryption can be used in information transmission systems in order to reduce network traffic and ensure data security. The resulting encryption and archiving software was used in the module of the software package «Diplomas SNU v.2.6.1», which was developed at the Volodymyr Dal East Ukrainian National University. This complex is designed to create a unified register of diplomas at the university, automate the creation of files-diplomas of higher education in the multifunctional graphics editor Adobe Photoshop. The controller exports all data for analysis and formation of diplomas from the parameters of the corresponding XML files downloaded from the unified state education database in compressed zip archives. The developed module performs the process of unzipping and receiving XML-files with parameters for the further work of the complex «Diplomas SNU v.2.6.1».

1. Ratov D. Architectural paradigm of the interactive interface module in the cloud technology model. *Applied Computer Science*. 2020. **16**, N 4. P. 48–55.
2. Ратов Д.В., Иванов В.Г., Лыгина Л.А. Создание сетевой системы авторизации для ПО. *Математические машины и системы*. 2021. № 2. С. 35–44.
3. Архивация. https://ru.wikipedia.org/wiki/%D0%90%D1%80%D1%85%D0%B8%D0%B2%D0%B0%D1%86%D0%B8%D1%8F
4. Ратов Д.В. Программный контроллер автоматизации формирования документов с ограничением несанкционированного доступа. *Научные труды ДонНТУ. Серия «Информатика, кибернетика, вычислительная техника»*. Покровск, 2021. № 1(32). С. 49–56.
5. Глушаков С.В., Ломотько Д.В. Работа в сети Internet. [2-е изд., доп. и перераб]. Харьков : Фолио, 2003. 399 с.
6. Мельников В.П., Клейменов С.А., Петраков А.М. Информационная безопасность и защита информации. Учебное пособие для студ. высш. учеб. заведений. Под. ред. С.А. Клейменова. 3-е изд., стер. М. : Издательский центр «Академия», 2008. 336 с.
7. Ратов Д.В. Модель модуля пользовательского интерфейса информационной web-системы. *Математические машины и системы*. 2020. № 4. С. 74–81.
8. Гайдышев И. Анализ и обработка данных. СПб. : Питер, 2001. 750 с.
9. Романец Ю.В., Тимофеев П.А., Шаньгина В.Ф. Защита информации в компьютерных системах и сетях. М. : Радио и связь, 2001. 376 с.
10. Соколов А.В., Шаньгина В.Ф. Защита информации в распределенных корпоративных сетях и системах. М. : ГТК Пресс, 2002. 656 с.
11. Ватолин Д., Ратушняк А., Смирнов М., Юкин В. Способы сжатия данных. Устройство архиваторов, сжатие изображений и видео. М. : Диалог-МИФИ, 2003. 384 с.
12. Касилов О.В., Кравец В.А. Некоторые вопросы сжатия данных. https://ru.wikipedia.org/wiki/%D0%A1%D0%B6%D0%B0%D1%82%D0%B8%D0%B5_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85
13. Бердышев В.И., Петрак Л.В. Аппроксимация функций, сжатие численной информации, приложения. Екатеринбург, 1999. 296 с.
14. Алгоритм Хаффмана. https://neerc.ifmo.ru/wiki/index.php?title=%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%A5%D0%B0%D1%84%D1%84%D0%BC%D0%B0%D0%BD%D0%B0
15. Алгоритм Лемпеля–Зива. https://neerc.ifmo.ru/wiki/index.php?title=%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_LZW
16. Панасенко С.П. Алгоритмы шифрования. Специальный справочник. СПб. : БХВ-Петербург, 2009. 576 с.
17. Ratov D. Integration with the software interface of the com server for authorized user. *Applied Computer Science*. 2021. **17**, N 2. P. 5–13.

Отримано 30.11.2021