

Прискорення досягається за рахунок того, що в процедурі перевірки підпису при обчисленні контрольної складової цифрового підпису замість операції скалярного добутку великого цілого числа на базову точку $P \in E$ використовується лише сама базова точка P . При цьому показано, що рівень криптостійкості запропонованого методу не зменшився порівняно з відомим.

Здійснено програмну реалізацію запропонованого методу, а також проведено порівняльний аналіз часу формування/перевірки підпису за даним та відомим методом ЦП. Запропонований метод ЦП має приблизно в 1,5-2 рази більш швидко процедуру перевірки підпису порівняно з відомими методами ЦП на основі ЕК.

Таким чином, запропонований метод дозволив підвищити швидкість перевірки цифрового підпису порівняно з відомими методами.

Література: 1. ANSI X9.62-1999. *Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)*. – 1999. 2. Болотов А. А., Машиков С. Б., Фролов А. Б., Часовских А. А. *Элементарное введение в эллиптическую криптографию. Алгебраические и алгоритмические основы*. – М.: КомКнига, 2006. – 328 с. 3. Беспалов А. В., Телиженко А. Б. *Криптосистемы на эллиптических кривых: Учеб. Пособие*. – К.: ИОЦ «Видавництво «Політехніка», 2004. – 224 с.: іл. 4. Menezes A. J., van Oorschot P. C., Vanstone S. A. *Handbook of Applied Cryptography*. CRC Press, 1996. 5. Adrian Antipa, Daniel R. L. Brown, Robert P. Gallant, Robert J. Lambert, Rene Struik, Scott Vanstone. "Accelerated Verification of ECDSA Signatures". *Certicom Research, Canada, 2005*: p. 307-318. 6. Пат. EP1306750 JP, МКИ G09C1/00; G06F7/72; G09C1/00; G06F7/60; (IPC1-7): G06F7/72. "Multi-scalar multiplication computation in elliptic curve signature verification": Пат. EP1306750, МКИ G06F7/72F1. // Okeya Katsuyuki (JP) - № EP20020255073; Заявл. 19.07.2002; Опубл. 02. 05. 2003. 7. Пат. US2007064932 CA, МКИ H04L9/30; H04L9/28. "Accelerated verification of digital signatures and public keys" // Struik Marinus; Brown Daniel; Vanstone Scott; Gallant Robert; Antipa Adrian; Lambert Robert - №US20060333296. Заявл. 18. 01. 2006; Опубл. 22. 03. 2007. 8. Пат. UA24459, МПК(2006) H03M 13/00. «Спосіб цифрового підписування на основі математичного апарату еліптичних кривих» // Яремчик Ю. С.; Черняхівич К. В. - № U200705052. Заявл. 07.05.2007; Опубл. 25.06.2007, Бюл. №9, 2007. 9. Anthony Mulcahy. *BorZoi – An Elliptic Curve Cryptography Library* // Available at <http://dragongate-technologies.com/products.html>. 10. ДСТУ 4145-2002. *Криптографічний захист інформації. Цифровий підпис, що ґрунтується на еліптичних кривих. Формування та перевіряння*. – К.: Держстандарт України, 2003. – 94 с. 11. Standards for Efficient Cryptography Group. *SEC 2: Recommended Elliptic Curve Domain Parameters, September 2000. Version 1.0*. Available at www.secg.org. 12. Яремчик Ю. С., Черняхівич К. В. *Метод цифрового підписування на основі математичного апарату еліптичних кривих з прискореною процедурою перевірки підпису*. // *Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні*. – К., 2007. – №1 (14), – С. 119 – 127.

УДК 681.31

ДОСЛІДЖЕННЯ МЕТОДІВ ЗАВАДОСТІЙКОГО КОДУВАННЯ ДЛЯ АВТОМАТИЗАЦІЇ ВИЗНАЧЕННЯ КОГНІТИВНИХ ВЛАСТИВОСТЕЙ ІНФОРМАЦІЇ

Сергій Троценко

Кафедра СКС ФПМ НТУУ "КПІ"

Анотація: Розглядається можливість застосування принципів завадостійкого кодування для визначення такої когнітивної властивості інформації, як її оригінальність. Приводяться визначення основних понять та постановка проблеми. Розглядається можливість порівняння послідовностей на основі принципів завадостійкого кодування, та їх застосування для визначення кількісних характеристик оригінальності інформації та пошуку запозичень.

Summary: The possibility of application of principles of error control codes for definition such cognitive properties of the information, as its originality is considered in this article. Definitions of the basic concepts and statement of a problem are given. The possibility of comparison of sequences on the basis of the error control codes, and also their application for definition of quantitative characteristics of originality of the information and search of loans is considered.

Ключові слова: Захист інформації, оригінальність інформації, завадостійке кодування.

В межах такої предметної області як інформатика інформацією називають відповідним чином упорядковану в результаті аналітико-синтетичної обробки множину даних, відомостей, знань, у тому числі комп'ютерних програм, які використовуються в автоматизованих системах (АС) незалежно від способу їх

фізичного та логічного подання. У процесі інформатизації суспільства накопичується велика кількість інформації і дуже важливою стає така когнітивна властивість інформації, як її унікальність та оригінальність [1].

Поява електронних носіїв та комп'ютерних мереж досить гостро поставила перед людством питання захисту інформації з точки зору її унікальності та оригінальності. Легкість, з якою можна видати чужу програму або статтю за свою, зробивши певні зміни, викликає занепокоєння. Спроби якнайкраще захистити цінну інформацію спричинили появу складних шифрів. Їх поява була пов'язана з проблемою захисту інформації при передачі повідомлень лініями зв'язку від виникаючих при цьому помилок. Останнім часом спостерігається посилення впливу ідей і методів теорії кодування на інші області комп'ютерної науки. Відомі приклади, коли використання тих або інших кодових конструкцій приводило до суттєвих досягнень в питаннях, які на перший погляд досить далекі від задач теорії кодування. Відзначимо, наприклад, використання кодів Лемінга при отриманні асимптоти мінімальної кількості контактів, необхідної для реалізації будь-якої функції алгебри логіки від n змінних [2].

Принципи теорії завадостійкого кодування можуть бути корисними для визначення кількісних характеристик оригінальності інформації та пошуку запозичень. Оскільки проблему визначення оригінальності інформації до сих пір не можна вважати абсолютно вирішеною, то це дає поштовх для дослідження інших напрямків комп'ютерної науки для вирішення не типових для цих напрямків питань.

Звісно, питання визначення оригінальності інформації є досить широким, та не має загального оптимального рішення. Тому питання оптимальності має вирішуватись окремо для кожної предметної області, пристосовуючись під її потреби та особливості. Наприклад, для оптимального пошуку запозичень в програмному коді слід враховувати певну специфіку. Від простого тексту код відрізняється тим, що має певну структуру та складається із множини лексем, які можна розділити на множину логічних блоків. З іншого боку методи, які успішно працюють з порівнянням програмного коду або тексту, можуть бути не досить ефективними для визначення кількісних характеристик подібності між послідовностями двійкових символів. Але в будь-якому випадку, ключові ідеї, які стосуються вирішення задачі співставлення послідовностей певного алфавіту, будуть спільними.

Дійсно, пошук запозичень в програмному коді або в текстових файлах є більш ефективним, якщо представляти інформацію у вигляді скінчених послідовностей. У випадку пошуку запозичень у програмному коді, такий підхід називається токенізація [3], коли кожному оператору ставиться у відповідність певний код, з яких будеться відповідна послідовність. Під час аналізу програми у вигляді токенів ігноруються назви змінних та функцій, а також незначні зміни у коді програми.

Розглянемо основні підходи, які використовуються для співставлення послідовностей. Найпростішим методом є їх безпосереднє посимвольне порівняння. Фактично цей метод підраховує відстань Хемінга, тобто кількість символів, що відрізняє одну послідовність від іншої. Однак таке порівняння в межах послідовності стає зовсім неефективним, якщо окрім так званих заміщень символів (тобто зміни одного символу певного алфавіту на інший) має місце перестановка, видалення або вставка одного чи групи символів. Відстань між двома послідовностями, яка окрім заміщень підраховує також і кількість видалень та вставок, які необхідні для того, щоб перетворити одну послідовність в іншу, називається відстанню Левенштейна.

Для оцінки схожості послідовностей використовують також значення довжини найбільшої спільної послідовності. Ці методи мають високу оцінку часової складності, що неприйнятно для аналізу послідовностей великого розміру або співставлення баз даних (БД). Далі розглянемо можливість застосування принципів завадостійкого кодування як новий напрямок вирішення поставленої задачі.

Основна задача завадостійкого кодування полягає в необхідності безпомилково передати деяку інформаційну послідовність P через канал зв'язку з шумами. Для виявлення або навіть виправлення помилок, які можуть з'явитись при передачі, необхідно попередньо за допомогою процедури кодування сформувати кодове слово C :

$$F_{\text{код}}(P)=C. \quad (1)$$

Теорія кодування вивчає такі класи кодів, для яких кодування та декодування виконується не перебором, а в результаті деяких регулярних правил, визначених алгебраїчною структурою кодових комбінацій. У випадку блокових кодів надлишкова інформація, що містить кодове слово, дозволяє відновити вихідне інформаційне слово. Вона може бути змішаною з інформаційною частиною повідомлення у випадку нероздільних кодів або бути відокремленою послідовністю R у випадку роздільних кодів (рис. 1).

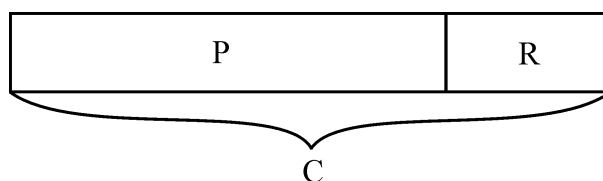


Рисунок 1 - Схема роздільного кодового слова

В результаті передачі кодового слова C каналом зв'язку з шумами отримаємо повідомлення C' , яке може відрізнитися або не відрізнитися від C . Необхідно виконати деякий алгоритм декодування $F_{\text{декод}}(C')$, результатом роботи якого буде повністю відновлена інформаційна послідовність P або деяка оціночна інформація про наявність помилок.

Такий алгоритм має працювати коректно при заданій максимальній кількості помилок як в інформаційній частині, так і в контрольній послідовності. Іншими словами декодер аналізує кількість та характер змін, які відбулися з P' відносно P , на базі інформації R . Цей підхід використовується в так званих систематичних кодах. Слід зазначити, що код гарантовано може виявити всі помилки кратності $q < d$, де d – відстань Хемінга. Якщо використовується кодування з виправленням q помилок, необхідною умовою є $q < \frac{d}{2}$.

Для виявлення факту помилки під час передачі інформації в несистематичних кодах використовується також дещо інший принцип, що оперує з контрольними сумами. В результаті роботи деякого алгоритму кодування $F_{\text{код}}(P)$ утворюється додаткова частина R , яку називають контрольною сумою. На іншому боці каналу приймач, в свою чергу, теж формує контрольну суму на основі прийнятого повідомлення:

$$F_{\text{код}}(P') = R'. \quad (2)$$

Приймач порівнює отримане R з обчисленим R' . Якщо ці значення відрізняються, то фіксується факт помилки в прийнятій послідовності. Іншими словами, на обох кінцях каналу обміну повідомленнями існує одна кодова послідовність $F_{\text{код}}$, яка формує контрольні суми. Висновок про наявність помилки в прийнятому повідомленні формується шляхом порівняння значень цих сум.

Проаналізуємо можливість використання нових методів порівняння послідовностей заснованих на базових принципах завадостійкого кодування, враховуючи основні вимоги, яким вони повинні відповідати:

- коректно аналізувати послідовності при видаленні, вставці та перестановці символів;
- здійснювати лінійну оцінку часової складності, що є важливим при проведенні аналізу по БД або послідовностям великого розміру;
- мати не дуже велику величину надлишкової частини, якщо є необхідність використання такої інформації.

Спираючись на описані вище ідеї з теорії кодування, можна запропонувати такий перший підхід для аналізу послідовностей. Нехай послідовність P' була утворена шляхом модифікації вихідної P , а саме перестановки, видалення або вставки одного чи групи символів. Тоді ступенем подібності двох послідовностей є число q , при $0 < q < d$, де d – задана максимальна кількість модифікацій послідовності. Зазначимо, що q і d фактично являють собою метрику Левенштейна, а при $q = 0$ порівнювані послідовності є точними копіями. Модифіковану послідовність P' можна вважати послідовністю P , передану через канал з помилками типу видалення, або вставки та або заміщення символів. На відміну від алгоритмів з теорії завадостійкого кодування, для оптимізації алгоритму співставлення послідовностей слід враховувати наступні умови:

- кодеру доступна результуюча послідовність P' ;
- перевірна надлишкова послідовність R не зазнає змін та модифікацій;
- декодеру доступна вихідна послідовність P ;

За допомогою функції кодування створюємо кодове слово:

$$F_{\text{код}}(P) = C = P \parallel R \quad (3)$$

При порівнянні еталонної послідовності з модифікованою сформуємо кодове слово:

$$C' = P' \parallel R \quad (4)$$

Застосуємо до C' функцію декодування $F_{\text{декод}}(C')$ і отримуємо кількість помилок в інформаційній частині P' відповідно до P . В цьому випадку кількість помилок відповідає кількості відмінностей між послідовностями. В найпростішому випадку відсутність помилок означає, що порівнюються точні копії. Вхідна послідовність аналізується разом з додатковими частинами послідовностей з БД. Чим менша кількість помилок виявляється, тим більше спільного між послідовностями, за умови, що алгоритм декодування виявляє помилки типу видалення та вставка групи символів.

Однак майже всі дослідження, які ведуться в області завадостійкого кодування, стосуються каналів, в яких відбуваються помилки типу заміщення символів. Підхід з використанням принципів завадостійкого кодування не може конкурувати по швидкодії з методом безпосереднього співставлення послідовностей в разі аналізу послідовностей лише з модифікацією типу заміщення символів. Тому інтерес викликають дослідження в теорії кодування, що стосуються кодів для каналів з видаленням, вставкою та заміщенням символів.

Теоретично існують коди, що здатні виявляти та навіть виправляти помилки типу видалення та вставки. Левенштейн розглянув модель каналу з помилками типу випадання та вставка та описав клас двійкових кодів з виправленням випадань, вставок та заміщень символів [4]. Також була представлена метрика Левенштейна, що показує можливість коду виправляти випадання/вставки, та дистанцію між двома послідовностями у вигляді кількості видалень вставок та заміщення символів.

На основі двійкового коду Левенштейна було розроблено недвійковий код, що виправляє заміщення або вставку одного символу [5]. Цей код використовує перетворення недвійкової послідовності в двійкову для визначення місця випадання або вставки. Скориставшись наведеним принципом побудови коду та наведеними специфічними умовами можна зробити перетворення недвійкової послідовності в двійковий код, що здатний виправляти задану кількість помилок видалення, вставки та заміщення символів. В даному випадку пошук відмінностей, а не виправлення помилок, є головною задачею.

Іншим підходом є використання принципів контрольної суми. Тоді потрібно не декодувати C' , а порівнювати додаткові частини R та R' послідовностей. Оскільки необхідно визначити кількісні характеристики подібності між послідовностями, а не сам факт їх відмінності, то необхідно висунути додаткові умови до контрольної суми. Вона повинна зберігати всю ключову інформацію про структуру послідовності для того, щоб можна було виявити модифікації типу видалення, вставка та переміщення символів.

Використання запропонованих методів є виправданим з точки зору швидкодії, оскільки завадостійке кодування використовується в телекомунікаційних системах, що вимагає від них роботи в режимі реального часу. Тому можна говорити про лінійний час виконання алгоритмів для визначення кількісних характеристик оригінальності інформації, розроблених з використанням принципів завадостійкого кодування.

Поняття оригінальності є відносним. Звісно, можна говорити про оригінальність однієї послідовності відносно іншої, але частіше постає питання оригінальності інформації відносно певної накопиченої до цього БД, часто досить великої. В цьому випадку окрім лінійного часу роботи алгоритму аналізу, слід звернути увагу також на прийнятні для вирішення поставленої задачі розміри надлишкових послідовностей.

Крім часу виконання аналізу БД, необхідно досягти оптимальної величини додаткової інформації. Вона повинна з одного боку не бути занадто великою, щоб не спричинити неприйнятно великого об'єму БД. З іншого боку додаткова інформація повинна зберігати ключову інформацію про структуру послідовності в разі використання методу контрольних сум.

Для ефективної роботи зазначених методів, дуже важливу роль відіграє метод представлення інформації, що аналізується, у вигляді послідовності певного алфавіту. Таке представлення має зберігати ключову, та ігнорувати поверхневу інформацію вихідного файлу. Це, окрім збереження важливої (з точки зору пошуку запозичень) інформації, повинно суттєво зменшувати об'єм інформації, що аналізується. Це позитивно відбивається на швидкості такого аналізу та його точності. Як вже зазначалося, в разі аналізу програмного коду дуже ефективним є представлення програми у вигляді послідовності токенів.

У випадку обробки текстових файлів скористатися таким підходом немає можливості, оскільки неможливо множини слів співставити множині певних числових кодів. В цьому випадку можна використовувати методи стиснення текстів, які не спотворюють даних. Збереження такої інформації можна досягти, якщо використати принцип алгоритму Soundex, описаний Дональдом Кнутом [6]. Цей алгоритм генерує однаковий ключ для слів, які мають схожу вимову. Відповідно, чим більше схожого між

згенерованими ключами, тим більш подібні відповідні послідовності.

Класичний алгоритм Soundex має коефіцієнт стиснення близько 2.1. При кількості 10000 байт вихідного тексту, маємо 4592 байт ключа, 3783 з яких не нульові. Підвищити коефіцієнт стиснення можна, якщо не дописувати нульові значення ключів при їх форматуванні. В цьому випадку розмір послідовності відносно вихідної збільшується до 2.6, причому результуюча послідовність не має незначущих символів, що підвищує ефективність пошуку. При попередній обробці тексту для оптимізації словосполучень (наприклад, замінити РН на F і т. д.), отримуємо стиснення більше, ніж в 2.7 рази. Наприклад, при обробці довільного тексту обсягом 1000 байт, результуюча послідовність складає 345 байт. При аналізі результуючих послідовностей, отриманих з такого тексту, за допомогою визначення дистанцій Левенштейна, порядок економії пам'яті складає 418,9 Кбайт (645*645 байт).

Застосування в методах пошуку запозичень підходів, які притаманні іншим областям комп'ютерної науки, є необхідним кроком, що дозволить винайти більш прості та ефективні алгоритми аналізу послідовностей. При аналізі методів завадостійкого кодування слід відштовхуватися від ідей, які закладені в конкретні методи, і абстрагуватися від конкретних реалізацій алгоритмів. Так, використавши ідею контрольних сум, і оптимізувавши алгоритм, описаний Дональдом Кнутом, ми отримали алгоритм, який спрощує аналіз вихідних послідовностей для пошуку запозичень. Використання методів завадостійкого кодування для автоматизації визначення когнітивних властивостей інформації стає можливим завдяки теорії кодів для каналів з помилками типу видалення, вставка та заміщення символів. Найбільш актуальною стає проблема пошуку недвійкових кодів для каналів з помилками типу видалення, вставка та заміщення символів.

Література: 1. Тарасенко В. П., Михайлюк А. Ю., Тесленко О. К., Осипов О. С. *Методологічні та термінологічні аспекти інформаційної стійкості освітніх комп'ютерних технологій та мереж.* – К.: *Радіоелектронні та комп'ютерні системи*, №7, 2006. 2. Лупанов О. Б. *О синтезе некоторых классов управляющих систем.* - Сб.: «Проблемы кибернетики», вып. 10, 1963. 3. Prechelt L., Malpohl G., and Philippsen M. *JPlag: Finding plagiarisms among a set of programs. Technical Report No. 1/00, University of Karlsruhe, Department of Informatics, March 2000.* 4. Левенштейн В. И., *Двоичные коды с исправлением выпадений, вставок и замещений символов*, Докл. АН СССР, 163, 4, 1965, 845 – 848. 5. Tenengolts G. M., *'Nonbinary Codes, Correcting Single Deletion or Insertion'*, *IEEE Transactions on Information Theory*, Vol. IT-30, No. 5, pp. 766 – 769, September, 1984. 6. D. Knuth, *The Art Of Computer Programming*, vol. 3: *Sorting And Searching*, Addison-Wesley, 1973.

УДК 378:147.157+004:37

АЛГОРИТМ МНОГОПРОГРАММНОЙ ГЕНЕРАЦИИ ПАРОЛЬНОГО ДОСТУПА К ДАННЫМ

Георгий Кожевников, Татьяна Бондаренко

Украинская инженерно-педагогическая академия, м. Харьков

Анотація: Наведені алгоритми парольного доступу до даних з використанням програмного методу генерації парольного доступу. Проаналізовані недоліки існуючих методів формування парольного доступу та запропоновані методи їх подолання. Зазначені алгоритми доведені до програмної реалізації.

Summary: In article was carried out the development of algorithm of password access to the lacks of existing methods of formation of password access. Programming language ASP.net was used for realization of algorithm of password access with using by maximum complex symbolical combination of the password.

Ключові слова: Пароль, програмна генерація, доступ, захист даних, алгоритм, випадкове число, макрос.

1 Исходные предпосылки

Одним из наиболее распространенных методов защиты информации является парольный доступ к данным. Однако, наряду с несомненными достоинствами, этот метод защиты данных обладает и определенными недостатками: пароль можно забыть, его могут «взломать», если он недостаточно сложный.

Таким образом, при использовании парольного доступа возникает дилемма: простой пароль никому не нужен, т. к. его легко взломать; сложный пароль никому не нужен, т.к. его легко забыть или потерять, или обнаружить, записанным где-либо.