

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).

Застосовування 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. Харків

Анотація: Наведені алгоритми паролного доступу до даних з використанням програмного методу генерації паролного доступу. Проаналізовані недоліки існуючих методів формування паролного 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.

І 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 паролного 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.

Специалисты рекомендуют в качестве пароля использовать фразы. Но и эти пароли не лишены недостатков. Известно, что словари всех языков давно составлены вместе со всеми возможными формами слов и распространённые системы подбора паролей начинают перебор, используя словарь, в котором слова расставлены по частоте их употребления.

II Постановка задачи

В связи с выше изложенным возникает задача разработать алгоритм парольного доступа, который с одной стороны обладает достаточной степенью стойкости к попыткам «взлома» пароля, а с другой стороны не требует больших усилий для запоминания пароля и его хранения.

III Методы исследований

Указанные выше недостатки могут быть устранены, если вместо обычной схемы парольного доступа использовать алгоритм, использующий метод программной генерации парольного доступа к данным, приведенный на рис. 1.

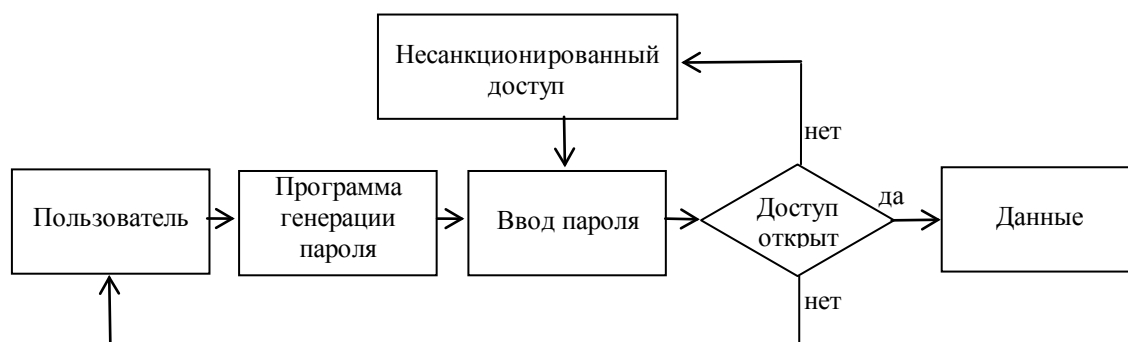


Рисунок 1 – Алгоритм программной генерации парольного доступа к данным

Как и в обычном алгоритме, доступ к данным будет открыт при условии правильного ввода пароля. Но, если несанкционированный доступ осуществляется с обычной точки ввода пароля, то пользователь для формирования пароля использует специальную программу генерации.

В качестве программы генерации пароля можно использовать известные программы, например, Advanced Password Generator. Однако делать это не рекомендуется, поскольку для опытного пользователя не составит труда включить данную программу в приведенный выше алгоритм программной генерации паролей. Разработка собственной программы не займет много времени, а секретность алгоритма генерации пароля будет служить дополнительной степенью защиты для системы хранения данных.

В качестве параметров генерации пароля можно использовать набор символов, которые могут быть использованы для записи пароля, длину пароля и ключ генерации пароля (любое число или сочетание символов). Последний параметр используется для инициализации генератора случайных чисел. Таким образом, длина пароля не зависит от длины ключа, то есть, вводя всего один символ, вы можете получить максимально возможную последовательность символов пароля. Чтобы исключить возможность несанкционированной генерации пароля целесообразно в программе генерации пароля кроме генератора случайных чисел использовать дополнительные функции (тригонометрические, экспоненциальные и т. п.), обеспечивающие искажение результатов работы датчика случайных чисел.

Если в качестве пароля выбрано случайное сочетание символов клавиатуры разных регистров, то при длине в 12 символов число возможных вариантов пароля будет примерно $142^{12} \approx 7 \cdot 10^{25}$. Тогда, если кто-либо попытается взломать такую защиту, перебирая пароль со скоростью 10 000 вариантов в секунду (средний компьютер), то перебор вариантов займёт [2]

$$\frac{142^{12} \text{ вариантов}}{10000 \text{ вар} / \text{сек}} \approx 7 \cdot 10^{21} \text{ сек} \approx 2 \cdot 10^{14} \text{ лет.}$$

Однако, маловероятно, чтобы кто-либо использовал для формирования пароля все возможные символы клавиатуры. Вместе с тем, при использовании алгоритма парольной генерации, если ввод сгенерированного пароля осуществляется не через клавиатуру, а через буфер обмена, число возможных

символов пароля равно не 142, как на клавиатуре, а 256 (полный набор символов ASCII). При этом число возможных вариантов пароля при его длине 12 символов будет равно $256^{12} \approx 8 \cdot 10^{28}$. Таким образом, использование алгоритма парольной генерации позволяет повысить стойкость парольной защиты данных.

Еще одна проблема парольного доступа к данным – необходимость использования нескольких паролей для различных целей. Специалисты настоятельно рекомендуют не использовать один пароль для нескольких целей. Предположим, что вы придумали и ухитрились запомнить один хороший пароль. Но что делать, если вам необходимо запомнить 3 – 4, а может и больше, паролей. В данном случае использование алгоритма программной генерации парольного доступа существенно облегчает решение данной проблемы. Используя алгоритм программной генерации, пользователь может использовать в качестве пароля номер телефона, имя любимой собаки, дату рождения и все возможные варианты, которые существенно облегчают запоминание, но не рекомендуют использовать специалисты при обычном парольном доступе.

При использовании описанного выше алгоритма для генерации пароля используется одна программа. Модификацией рассмотренного алгоритма является алгоритм многопрограммной генерации парольного доступа при котором для генерации пароля по заданному коду используется набор программ, реализующих различные алгоритмы генерации пароля. Алгоритм многопрограммной генерации парольного доступа с параллельным алгоритмом запуска программ приведен на рис. 2.

При использовании данного алгоритма пользователь вводит набор символов, обеспечивающий выбор одной из множества программ генерации паролей (ППП). В данном случае в набор программ генерации пароля могут быть включены различные типы программ, написанные с использованием различных технологий и приложений. Таким образом, увеличивается количество точек ввода в систему с парольным доступом, что является дополнительной степенью защиты от программ класса spyware. При этом задача формирования пароля сводится к задаче прохождения узлов дерева, которыми являются отдельные программы формирования пароля (ППП_i). Если узлы дерева формирования пароля содержатся в $m \geq 0$ попарно не пересекающихся множествах T_1, \dots, T_n , то мы имеем упорядоченное дерево, алгоритмы прохождения которого подробно описаны в [3].

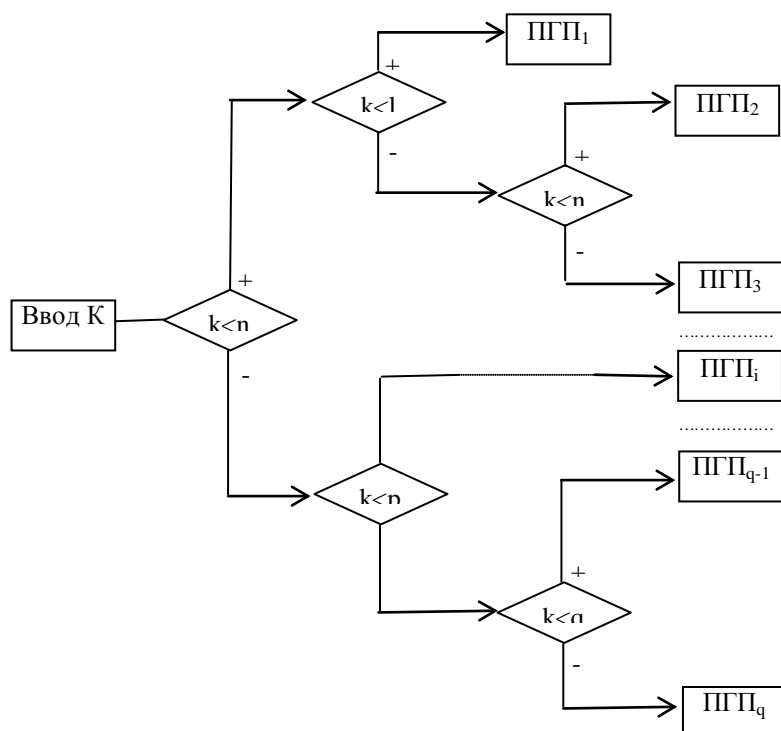


Рисунок 2 – Алгоритм многопрограммной генерации парольного доступа

При реализации многопрограммного алгоритма возможны два варианта формирования пароля.

- Пароль формируется только в конечных узлах дерева, а промежуточные узлы используются для реализации алгоритма ветвления. Количество вариантов формирования пароля в этом случае определяется количеством конечных узлов дерева;

- Пароль формируется за счет добавления и/или преобразования отдельных элементов в формируемом поле в каждом узле дерева. Количество вариантов формирования пароля в этом случае увеличивается в $\prod_i a_i$ раз, где a_i – количество узлов на i -м уровне дерева.

Для повышения стойкости системы парольной защиты данных можно также использовать алгоритм многопрограммной генерации парольного доступа с последовательным алгоритмом запуска программ. При этом в зависимости от ввода комбинации символов в данном случае на выполнение будут запускаться последовательно ряд программ из набора ПГП. В этом случае может быть реализован некий сценарий доступа к системе, при реализации которого от пользователя потребуется ввод ряда ответов на поставленные вопросы, что обеспечит последовательное выполнение ПГП в соответствии с заданным алгоритмом.

Известно, что при наборе пароля, хотя он, как правило, на экране не отображается, следует всё же избегать чужих глаз. Предлагаемый алгоритм программной генерации пароля предоставляет возможность легко справиться с данной задачей. Для этого необходимо программу генерации пароля оформить в виде макроса MS Word или MS Excel. Тогда порядок ввода пароля может быть следующим:

- пользователь вызывает, например, текстовый документ, содержащий встроенный макрос генерации паролей;
- при загрузке документа либо при выполнении определенных действий с текстом макрос запускается на выполнение и генерирует пароль, который записывает в буфер обмена Office;
- сгенерированный макросом пароль из буфера обмена вводится в поле ввода пароля и очищается буфер обмена.

Еще одно достоинство предлагаемого алгоритма заключается в том, что сгенерированный пароль можно передавать в поле ввода через буфер обмена. При этом не используется клавиатура, что является дополнительной степенью защиты пароля пользователя от так называемых программ «клавиатурных шпионов».

IV Результаты

Авторами разработан на языке ASP.net ряд простых, но эффективных программ генерации паролей. Данные программы используются для реализации как однопрограммного, так и многопрограммного алгоритма парольного доступа к данным. Опыт использования предложенных алгоритмов показал, что их применение не только не усложняет задачу парольного доступа к данным, но в отдельных случаях облегчает процедуру ввода пароля.

V Выводы

Описанный алгоритм, с одной стороны, облегчает пользователю решение проблемы запоминания и хранения стойких паролей, а с другой стороны – повышает стойкость системы хранения данных к несанкционированному доступу за счет использования максимально сложной символьной комбинации пароля.

Литература: 1. Бэнкс М. А. Информационная защита ПК: Перевод с английского – К.: БЕК+, 2001 – 272 с. 2. Скляров Д. В. Искусство защиты и взлома информации. – СПб.: БХВ – Петербург, 2004 – 288 с. 3. Кнут Д. Искусство программирования для ЭВМ. Т. 1. Основные алгоритмы. – М.: Мир, 1976, 734 с.

УДК 681.3.06

ЛИНЕЙНЫЙ СТОХАСТИЧЕСКИЙ АЛГОРИТМ ШИФРОВАНИЯ, БАЗИРУЮЩИЙСЯ НА КАНОНИЧЕСКОМ РАЗЛОЖЕНИИ СЛУЧАЙНОЙ ПОСЛЕДОВАТЕЛЬНОСТИ

Игорь Атаманюк

Николаевский государственный аграрный университет

Анотація: Розглянуто метод шифрування, який дозволяє перетворити повідомлення, що передається, в послідовність некорельованих значень, що суттєво ускладнює задачу відтворення початкових даних. Алгоритм базується на канонічному розкладі випадкової послідовності, що досліджується.