УДК 004.056.55

ХРАНИТЕЛЬ ПАРОЛЕЙ – ПРОГРАММА ЗАЩИЩЕННОГО ХРАНЕНИЯ ПАРОЛЕЙ ПОЛЬЗОВАТЕЛЕЙ

Пехотин Е.В.,
Донецкий национальный технический университет

Задача защиты информации довольно стара и на сегодняшний день остро стоит в информатике [1, с. 5-12]. Как известно, сегодня практически для любой программы легко достать так называемый кряк, т.е. в общем случае некоторый набор данных, предоставляющих к программе нелегальный, но полноценный доступ.

Сегодня не существует универсальных алгоритмов решения данной проблемы, и даже ранее с успехом работавший метод грубой силы (чем больше, тем лучше) дает осечку по причине существования инструментов, автоматизирующих процесс анализа и взлома. Побеждает не тот, кто сильнее, а тот, кто хитрее, кто может сделать неожиданный ход, ввести противника в заблуждение или выбить у него опору из-под ног. Лучшим решением будет комплексное применение защитных приемов, однако с нестандартным подходом к ним.

Данная работа направлена на разработку и реализацию одного из аспектов барьера на пути нелегального доступа к данным защищенной автоматизированной системы – хранение паролей авторизованных пользователей. Известно, что пользовательские пароли являют собой одновременно первый и один из уязвимых (а часто и наиболее слабых) мест в любой системе защиты, т.к. для любого злоумышленника обычно не составляет труда добраться до интерфейса ввода логина и пароля пользователей. А пользователи, как известно, не любят стойких, однако трудно запоминаемых паролей, предпочитая им более простые, но в то же время и менее криптостойкие.

В данной работе предпринята попытка в некоторой мере раз-

решить данную проблему. Создаваемая система для хранения паролей позволит пользователям хранить все их пароли в единой БД, не утруждая себя запоминанием, что приведет к тому, что они станут выбирать более сложные, однако и более криптостойкие пароли для использования в АС. Для защиты же самой БД, содержащей данные пароли (далее листа паролей) применяется мастер–пароль, с помощью которого шифруется эта БД. В качестве платформы для разрабатываемой системы по ряду причин было решено использовать Win32. Итак, можно привести требования, предъявляемые к этой системе:

- система должна хранить наборы троек: описание – логин – пароль (запись) в листе паролей;
- система должна шифровать участок файла БД, содержащий набор записей и критическую информацию о БД (контрольную сумму мастер-пароля); в алгоритмах шифрования/расшифровки должен использоваться мастер-пароль;
- при шифровании использовать мастер-пароль не как ключ шифрования, а как значение функции хеширования, а ее результат (хеш) использовать уже, как ключ шифрования;
- гарантировать отсутствие какой-либо информации, позволяющий восстановить мастер-пароль любым методом, кроме brute-force;
- использовать улучшенный генератор случайных чисел в соответствующих операциях;
- для создания элементов управления Windows использовать функции и объекты WinAPI.

Разработанная криптоархитектура системы показана на рис. 1.

Из рисунка видно, что в качестве функции хеширования мастер-пароля выбрана MD5 [2] – с одной стороны она безопаснее MD4 [3], с другой – проще и быстрее SHA-1 [4], и при этом она выдает 128-битное значение, которое можно разделить на 4 части по

32 бита (а при использовании SHA-1 пришлось бы добавлять еще один раунд в сеть Фейстеля, чтобы обеспечить разбиение, кратное 32 битам; также, при реализации алгоритма на x86 можно будет без проблем применить SSE–оптимизацию).

Также из рисунка видно, что в качестве режима блочного шифрования используется сцепление блоков шифротекста (CBC – Cipher Block Chaining) [5, с. 34-35], при котором каждый последующий блок сначала складывается по модулю 2 с предыдущим зашифрованным блоком, что дает зависимость текущего шифруемого блока от всех предыдущих. В данном случае размер блок CBC равен размеру получаемого хеша MD5 – 128 бит. В качестве IV (Init Vector, вектор инициализации) выбрана суперпозиция MD5(mpF(K, MP)), причем mpF(K,MP) :- strcat(strrev(MP), K), т.к. в этом случае фиксированная длина IV (128 бит) получается из нефиксированной конкатенации строк обращенного мастер-пароля и ключа – его MD5, что усложнит подбор исходных значений K и (главное) MP по известному IV.

В качестве базового алгоритма шифрования CBC используется 4-раундная сеть Фейстеля [5, с. 24-27]. В каждом раунде сети в качестве ключа используется соответствующая 32-битная часть 128-битного ключа. Знак ⊠ обозначает побитовое сложение по модулю 2 (операция XOR). В качестве основы для функций преобра-

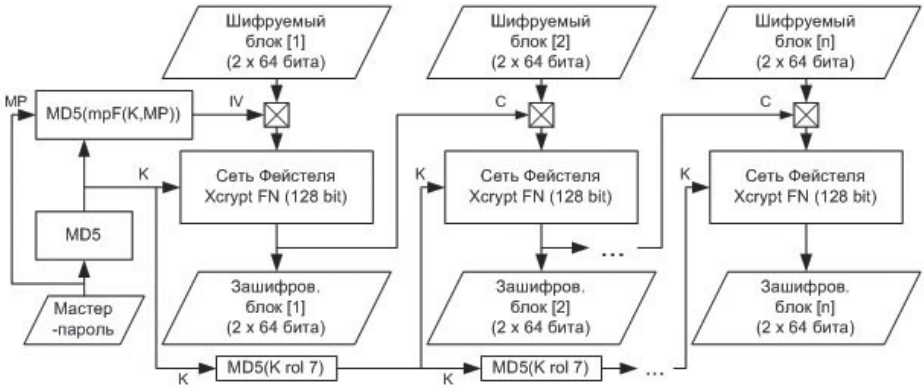

Рисунок 1 – Архитектура криптосистемы

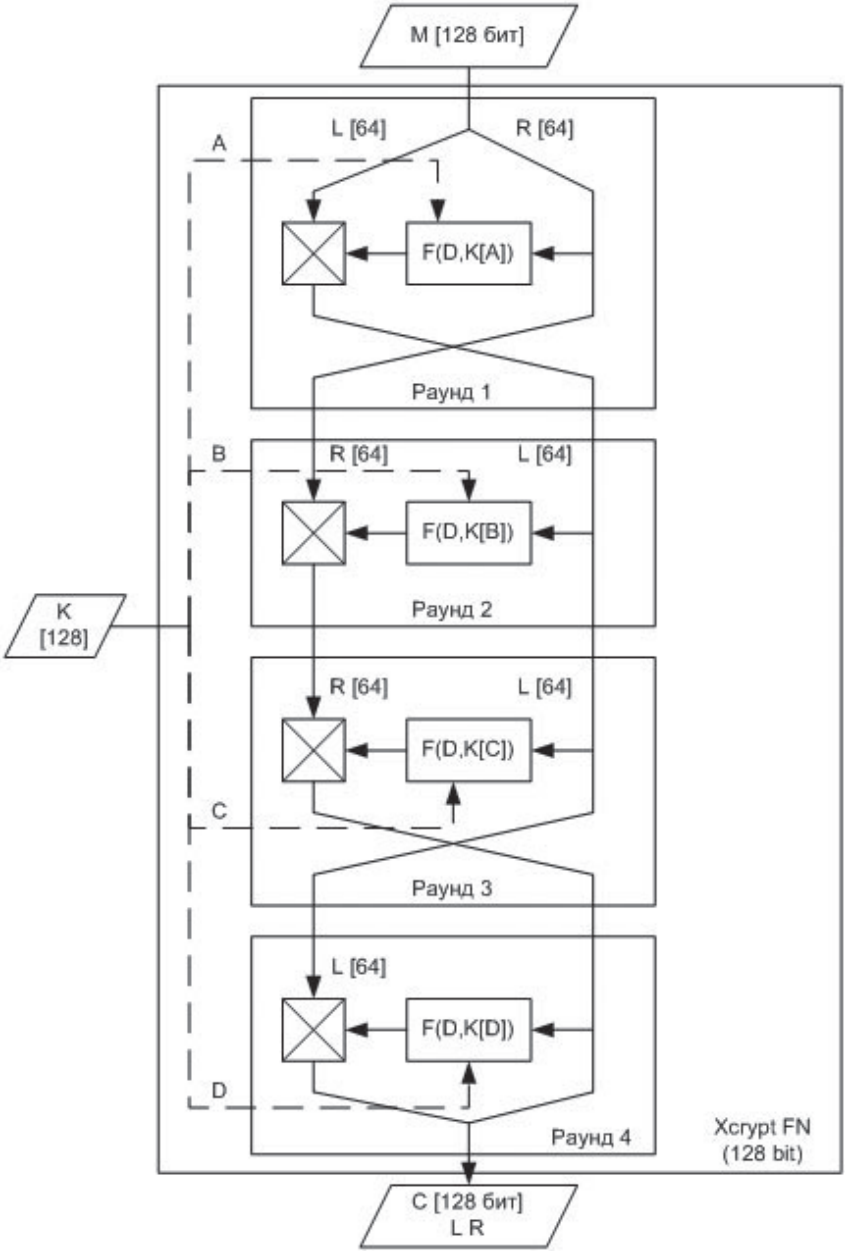

Рисунок 1 – Архитектура криптосистемы (продолжение)

зования правой части и ключа предложена такая идея: каждый бит 32-битового ключа обозначают маску преобразования для каждых 2 битов преобразуемой правой части. Предлагается такая четверка функций:

- X = R^2 (получим число размером 128 бит) и каждый бит ключа определяет, четный (0) или нечетный (1) бит берется из каждой 2-битовой пары X при формирования результата;
- X = R и каждый бит K означает, остается ли соответствующая 2-битовая пара X без изменений (0) или инвертируется (1; сложение по модулю 2 – XOR 11b);
- X = R и каждый бит K означает, складывается ли соответствующая 2-битовая пара X по модулю 2 с 01b (0; XOR 01b) либо 10b (1; XOR 10b);
- X = R и каждый бит K означает, остаются ли биты соответствующей 2-битовой пары X на своем месте (0) или меняются местами (1).

Преимущество преобразований такого типа состоит в том, что при превращении C=FN(M, K) могут получаться одинаковые C при одинаковых M и различных K или, другими словами, существуют несколько ключей K, преобразующий один и тот же M в один и тот же C. Это усложняет подбор правильного K.

Также, чтобы исключить возможность подбора ключа K путем поиска общей зависимости между блоками зашифрованного текста, для каждого следующего блока он модифицируется по алгоритму MD5(K rol 7).

Расшифровку информации, зашифрованной данной архитектурой, взломщику предстоит делать с конца цепочки CBC, при условии, что сигнификатор, указывающий на успех расшифровки, стоит в самом первом шифроблоке (контрольная сумма, см. ниже). Еще одно преимущество данной архитектуры – ни мастер-пароль, ни его хеш никак не хранятся в зашифрованном тексте, хотя математически и являются преобразованием в него исходного.

Единственный недостаток данной архитектуры – при

Рисунок 2 – Формат файла листа паролей

одинаковых значениях исходного текста и ключа получается один и тот же шифротекст. Данная проблема решается в формате файла листа паролей, который приведен на рис. 2.

Из рисунка видно, что в первом блоке CBC в файле идет контрольная сумма мастер-пароля (позволяющая путем расшифровки всего лишь одного блока CBC определить, правильно ли введен пароль, или нет).

Следом идет 32-битное случайное значение, генерируемое улучшенным генератором случайных чисел (BRG, Better Random Generator). Это дает возможность каждый раз менять первый блок CBC, что приводит к тому, что для одной и той же исходной информации (с т.з. пользователя) каждый раз получать различный шифротекст. Также, архитектура позволяет избежать хранения этого случайного значения в явном виде, что еще больше усиливает криптостойкость системы.

В качестве функции, вычисляющей контрольную сумму мастер-пароля, предлагается такая хеш-функция: [h = 0; for (i=0; (c = MP[i]) != 0; i++) { h = ((h rol 5) | (h ror 27)) + c; }; return h;]. Показано [6, с. 9], что данная хеш-функция при том, что она очень быстрая, имеет большую стойкость к появлению коллизий.

Далее идут записи БД – тройки ASCIIZ-строк описания-логин-пароль, после последней записи стоит 0 и далее идет выравнивание на 16. Для усиления последнего блока CBC предлагается заполнять это выравнивание случайными значениями, генерируемыми BRG.

Итак, результатом данной работы стал проект криптосистемы, предназначенной для безопасного защищенного хранения паролей

пользователей в базе данных паролей с использование мастер-пароля для доступа к ней.

Литература

[1] Соколов А.В., Степанюк О.М. Защита от компьютерного терроризма. Справочное пособие. – БХВ-Петербург, Арлит, 2002.
[2] MD5 [Электронный ресурс]: ru.wikipedia.org/wiki/MD5
[3] MD4 [Электронный ресурс]: ru.wikipedia.org/wiki/MD4
[4] SHA-1 [Электронный ресурс]: ru.wikipedia.org/wiki/SHA-1
[5] Методические указания и задания к лабораторным работам по курсу «Информационная безопасность» для студентов специальности «Программное обеспечение АС», часть 1 - «Криптографические и стеганографические методы защиты информации» / Сост.: Губенко Н.Е., Чернышова А.В. - Донецк, ДонНТУ, 2007.
[6] Win32 Assembly Components by The Last Stage of Delirium Research Group [Электронный ресурс]: pentest.cryptocity.net/files/exploitation/winasm-1.0.1.pdf