А. А. Шагун, А. В. Баранов, А. М. Кадан

ВОССТАНОВЛЕНИЕ ИНФОРМАЦИИ ПОВРЕЖДЕННЫХ АРХИВОВ

Задача восстановления поврежденных архивных файлов весьма актуальна в условиях проведения специальной компьютерной экспертизы, в случае умышленного или неумышленного повреждения архивов. В статье представлена учебная утилита восстановления поврежденных архивных файлов. Предварительно рассмотрен формат RAR-архива, используемые методы сжатия, подходы, предлагаемые для снижения риска повреждения архива.

Введение

В настоящее время RAR – один из самых популярных форматов сжатия данных и не менее популярная программа-архиватор. Формат был разработан российским программистом Евгением Рошалом [1]. Он также является автором программы-архиватора для упаковки/распаковки RAR-архивов. Основными преимуществами архиватора RAR перед другими (например, ZIP) является обеспечение хорошей степени сжатия, поддержка многотомных архивов, добавление информации для восстановления, которая позволяет восстановить физически поврежденный файл, и блокировка архивов для предотвращения случайной модификации особенно ценных данных. Дополнительные возможности защиты информации обеспечиваются за счет шифрования данных и имен файлов в архиве с использованием 128-битного алгоритма AES и сохранения данных о правах доступа, что делает использование RAR достаточно криптостойким. Применение RAR позволяет обрабатывать файлы практически неограниченного размера (до 8 экзабайт). Для сжатия программа использует алгоритм PPMd, о котором будет рассказано немного позже.

Архивирование обычно применяется для хранения пока ненужной информации и для ее переноса или передачи по каналам связи, с другой же стороны наиболее дешевым и простым вариантом защиты данных является их сжатие. В этом случае можно создать архив со сложным паролем, который непросто взломать. Архиватор также позволяет нам разбить на несколько томов сжимаемый файл, это бывает необходимо для записи архива на носители ограниченной емкости.

Особенности формата RAR-архива

Файл RAR-архива имеет сложную структуру [1] и состоит из блоков разной длины, порядок следования которых может меняться. Первым блоком всегда должен быть блок-маркер, за которым следует блок заголовка архива. Каждый блок начинается со следующих полей: HEAD_CRC (2 байта, CRC всего блока или его части), HEAD_TYPE (1 байт, тип блока), HEAD_FLAGS (2 байта, флаги блока), HEAD_SIZE (2 байта, размер блока), ADD_SIZE (4 байта, необязательное поле – добавление к размеру блока).

Обработка архивного файла заключается в сканировании структуры и содержимого архива, поиске блоков нужного типа, определении их флагов и атрибутов, выборе информации из блоков и восстановлении сохраненных в архиве объектов с учетом использованного алгоритма сжатия:

1) прочитать и проверить блок-маркер;
2) прочитать заголовок архива;
3) прочитать или пропустить HEAD_SIZE;
4) если обнаружен конец архива, то обработка архива прекращается, иначе прочитать 7 байтов в полях HEAD_CRC, HEAD_TYPE, HEAD_FLAGS, HEAD_SIZE;
5) проверить HEAD_TYPE.

Методы сжатия, используемые в RAR

Существует 2 класса методов сжатия информации: сжатие без потерь, сжатие с потерями. При использовании сжатия без потерь возможно полное восстановление исходных данных, сжатие с потерями позволяет восстановить данные с искажениями, обычно несущественными с точки зрения дальнейшего использования восстановленных данных.

Основная характеристика алгоритма сжатия – коэффициент сжатия:

$$k = \text{объем исходных данных/объем сжатых}$$

Если $k = 1$, то алгоритм не производит сжатия. Если $k < 1$, то алгоритм порождает данные большего размера, нежели исходные, т. е. производит «вредную работу».

Основным алгоритмом сжатия, используемым в RAR, является PPM (Predication by Partial Matching) – адаптивный статический алгоритм сжатия данных без потерь, основанный на контекстном моделировании и предсказании [2], и его вариант для сжатия текстовых данных PPMd, используемый также в 7-Zip и WinZip. Модель PPM использует контекст – множество символов в несжатом потоке, предшествующих данному, чтобы предсказывать вероятность символа на основе статистических данных. Сама модель PPM лишь предсказывает значение символа, непосредственно сжатие осуществляется алгоритмами энтропийного кодирования, как, например, алгоритм Хаффмана, арифметическое кодирование [3].

RAR использует также алгоритм сжатия, основанный на LZSS [4]. В LZSS базовый алгоритм (LZ, алгоритм Лемпеля-Зива) был улучшен по 3 направлениям: упреждающий буфер сохранялся в циклической очереди; буфер поиска хранился в виде дерева двоичного поиска; метки имели 2 поля, а не 3.

Идея алгоритма заключается в добавление к каждому указателю и символу однобитового префикса, позволяющего различать эти объекты. Иначе говоря, однобитовый флаг указывает тип и, соответственно, длину непосредственно следующих за ним данных. Такая техника позволяет:

– записывать символы в явном виде, когда соответствующий им код имеет большую длину, и, следовательно, словарное кодирование только вредит;
– обрабатывать ни разу не встреченные до текущего момента символы.

В RAR размер окна для поиска данных (также известно под названием «дифференциальный словарь») может изменяться от 64 Кб до 4 Мб, минимальная длина совпадения составляет 2, используются специальные коды для лучшего сжатия повторяющихся офсетов. Символы, офсеты, и длины совпадения сжимаются методом Хаффмана.

Подходы к восстановлению поврежденной информации

Никогда нельзя исключить риск повреждения сжатого файла – из-за сбоев операционной системы или программного обеспечения, выхода из строя оборудования, неосторожности самих пользователей, а также при его передаче на мобильных носителях или по каналам Интернета.

Частично проблему снижения риска невозможности восстановления данных поврежденного архива решает добавление информации «на восстановление». Наличие такой информации позволяет восстановить физически поврежденный файл, однако риск получить некорректный сжатый файл все равно велик.

Также задача восстановления поврежденных архивных файлов становится весьма актуальной в случае их умышленного повреждения и в условиях проведения специальной компьютерной экспертизы.

Информация для восстановления может содержать до 524 288 секторов для восстановления. Если поврежденные данные составляют непрерывный участок, то с помощью каждого сектора для восстановления можно восстановить 512 байт поврежденной информации. Это значение может снизиться в случае многократного повреждения. Если поврежденный архив не защищен информацией для восстановления или если его невозможно полностью восстановить из-за крупного повреждения, то происходит вторая стадия процесса восстановления, в ходе которой реконструируется только структура архива. Файлы с неверной контрольной суммой (CRC) после этой операции восстановить не удастся, однако становится возможным восстановить неповрежденные файлы, которые ранее были недоступны из-за нарушения структуры архива. Этот метод работает только с обычными, но не с непрерывными архивами.

Можно самостоятельно вычислить приблизительный дополнительный объем (объем информации для восстановления), пользуясь следующей формулой:

[размер архива] / 256 +
[количество секторов для восстановления] × 512 байт.

Как говорилось ранее, RAR использует криптографический алгоритм AES (Advanced Encryption Standard) с длиной ключа 128 бит. Криптографическая система WinRAR содержит фрагменты кода AES-реализации Шимона Стефанека и Брайана Гладмана, а также исходный код SHA-1 Стива Рейда [1].

Не так давно в RAR использовался стандарт DES (Data Encryption Standard). Главной причиной перехода от DES к AES стало то, что AES отличается гораздо более высокой криптостойкостью. Например, принимая за аксиому, что для взлома DES-пароля потребуется 1 секунда (перебор 2^{55} паролей в секунду), то компьютеру с тем же быстродействием потребуется приблизительно 149 триллионов лет для взлома 128-битного AES-пароля.

Учебная утилита для восстановления поврежденных архивов

Нами разрабатывается учебная утилита, которая позволяет решать задачу восстановления поврежденных архивных файлов, сжатых архиватором RAR. Используя ее возможности, можно частично избежать или минимизировать потери сжатой информации.

Несмотря на то, что существует целый ряд программ для восстановления поврежденных архивов [5], данная работа актуальна и представляет как теоретический интерес, так и должна способствовать развитию глубоких практических навыков работы со сложными структурами данных.

Пользовательский интерфейс учебной утилиты выполнен в виде пошагового мастера. Интуитивно понятный и удобный интерфейс дает возможность легкой работы с программой даже для неподготовленного пользователя. Пример окна интерфейса мастера разработанной учебной утилиты представлен на рис. 1:

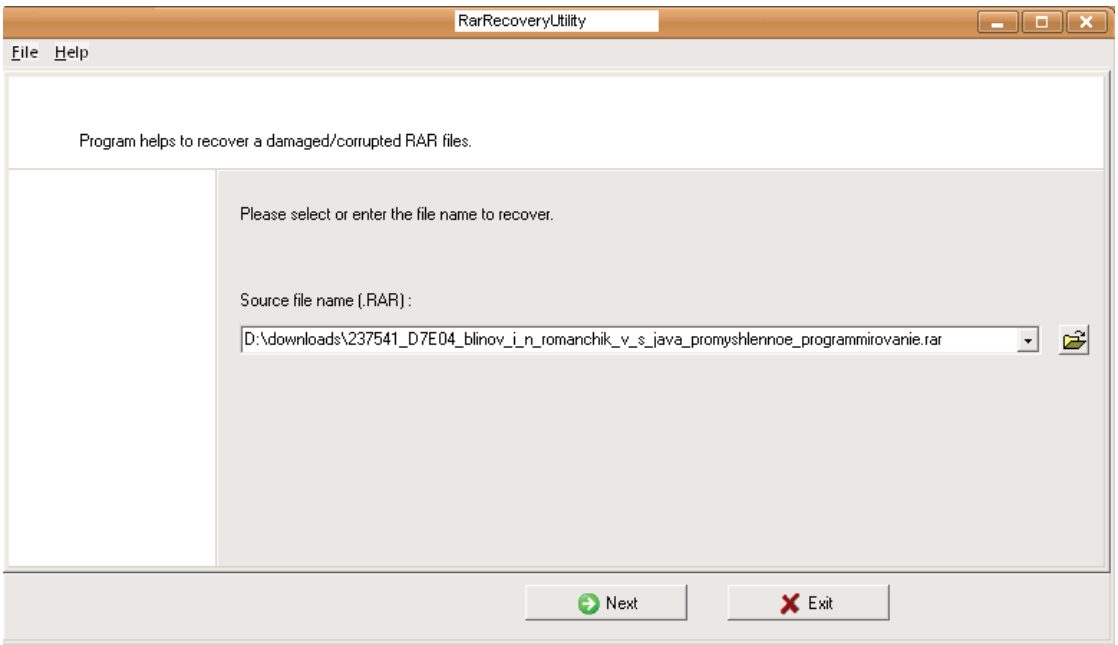

Рис. 1. Вид начального окна мастера учебной утилиты

На каждом этапе этого мастера пользователь (обучаемый) должен выполнить всего лишь одно действие, предусмотренное логикой алгоритма восстановления. В программе предусмотрено последовательное пошаговое выполнение таких действий, как:

1) анализ структуры архива, формирование списка папок и файлов;
2) контроль найденных файлов с целью поиска повреждений;
3) выбор поврежденных файлов для восстановления;
4) выбор директории, куда нужно сохранить восстановленный файл либо файлы;
5) выполнение восстановления файла;
6) формирование отчета о результатах работы.

Утилита проводит сканирование и анализ поврежденного архива, извлекая из него информацию о файлах и каталогах, включенных в него. Список найденных объектов предоставляется пользователю, который должен выбирать для восстановления необходимые ему файлы (рис. 2).

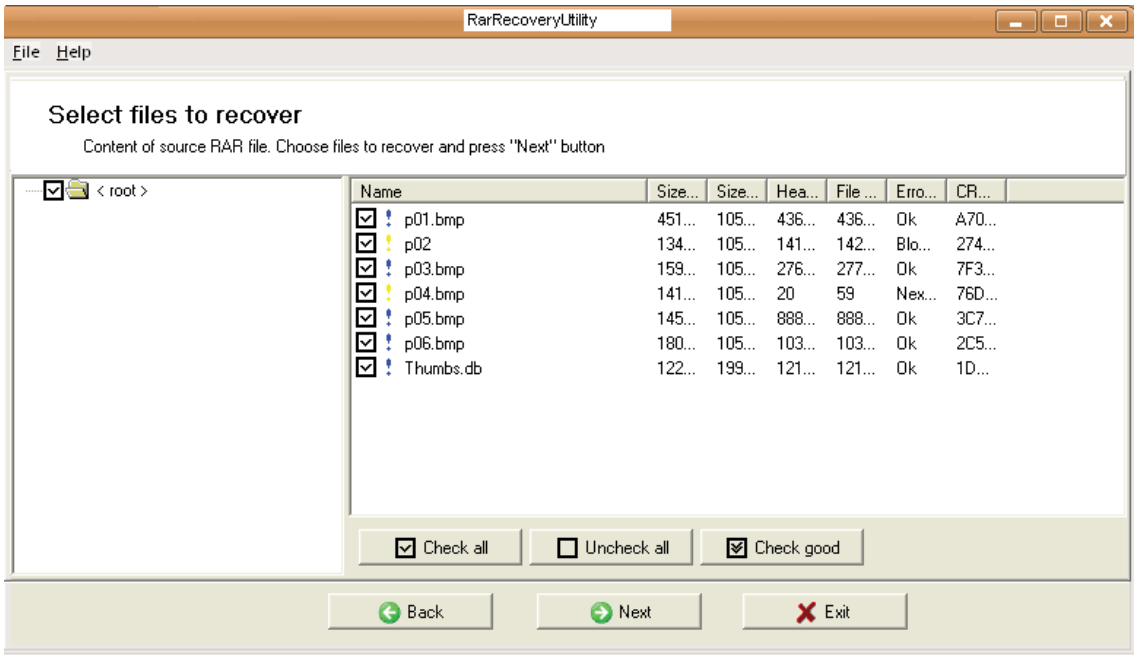

Рис. 2. Пример интерфейса мастера учебной утилиты. Выбор файлов для восстановления

Утилита может полностью или частично восстановить выбранные файлы или не восстановить их вообще, в зависимости от степени повреждения архива. Извлеченные файлы и папки сохраняются в указанном пользователем месте, после чего они доступны для использования и дальнейшего анализа. Исходный архив остается неизмененным и никак не модифицируется.

Работа учебной утилиты сопровождается выводом подробных комментариев, которые отражают особенности работы алгоритмов анализа структуры файла архива и восстановления поврежденной информации. Комментируются также критерии, по которым была определена ситуация наличия повреждения данных, анализ степени повреждения, работа алгоритма восстановления поврежденных файлов.

Литература

1. WinRAR Официальный сайт в России [Электронный ресурс]. Режим доступа: http://www.win-rar.ru. Дата доступа: 05.04.2011.

2. *John, G.* Cleary and Ian H. Witten. Data Compression Using Adaptive Coding and Partial String Matching / John G. Cleary, Ian H. Witten // IEEE Transactions on Communications, Vol. COM-32, No. 4, April 1984. P. 396–402.

3. *Фомин, А. А.* Основы сжатия информации / А. А. Фомин; СПБ ГТУ. – Санкт-Петербург, 1998.

4. *Storer, J. A.* Data compression via textual substitution / J. A. Storer, T. G. Szymanski . J.ACM 29,4(Oct. 1982). P. 928–951.

5. *Ватолин, Д.* Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео / Д. Ватолин, А. Ратушняк, М. Смирнов, В. Юкин. Москва. М.: ДИАЛОГ-МИФИ, 2002.

Кадан Александр Михайлович, заведующий кафедрой системного программирования и компьютерной безопасности Гродненского государственного университета имени Янки Купалы, кандидат технических наук,доцент, alexander.kadan@gmail.com

Баранов Андрей Владимирович, студент факультета математики и информатики Гродненского государственного университета имени Янки Купалы, zer0sandrew@gmail.com

Шагун Андрей Александрович, студент факультета математики и информатики Гродненского государственного университета имени Янки Купалы, shahunaa@gmail.com