УДК 004.056.53

ПРОГРАММНАЯ СИСТЕМА КОНТРОЛЯ ИНФОРМАЦИОННОЙ ЦЕЛОСТНОСТИ НА ОСНОВЕ ВОССТАНАВЛИВАЮЩЕГО ЦИФРОВОГО ВОДЯНОГО ЗНАКА

К. т. н. К. В. Защелкин, к. т. н. И. Г. Милейко, А. А. Ищенко

Одесский национальный политехнический университет
Украина, г. Одесса
const-z@te.net.ua

Рассмотрены подходы к контролю целостности информационных объектов. Предлагается программная система контроля целостности растровых изображений, основанная на внедрении в них восстанавливающего цифрового водяного знака.

Ключевые слова: контроль целостности данных, цифровой водяной знак, хеш-сумма.

Одним из важных аспектов обеспечения информационной безопасности выступает задача контроля целостности данных, т. е. обнаружения их несанкционированного изменения. Большинство существующих подходов к контролю целостности основаны на добавлении к контролируемому информационному объекту некоторого блока данных (контрольной суммы, дайджеста, цифровой подписи), обеспечивающего контроль [1]. Этот блок данных присоединяется к информационному объекту, в результате чего становится его частью, либо хранится и передается вместе с объектом. Недостатком таких подходов является открытость факта наличия контрольного блока и возможность определенных манипуляций с этим блоком. В данной работе рассматривается подход, в рамках которого контрольный блок внедряется в информационный объект в виде цифрового водяного знака (ЦВЗ). В этом случае скрывается факт наличия контрольного блока. При этом извлечение контрольного блока из информационного объекта возможно только при помощи специального стегоключа, определяющего правила размещения ЦВЗ в информационном объекте.

В работе предлагается программная система, выполняющая контроль целостности растрового изображения при помощи внедрения в него восстанавливающего цифрового водяного знака. Восстанавливающие ЦВЗ [2] обладают свойствами, дающими возможность восстановить исходное состояние контейнера ЦВЗ (то состояние, которое контейнер имел до внедрения ЦВЗ в него) после считывания ЦВЗ из контейнера. В этих условиях появляется возможность выполнить контроль целостности изображения при помощи встроенного в него ЦВЗ. Предлагаемая программная система выполняет внедрение ЦВЗ в изображение-контейнер в соответствии с методом Фридрих—Голджан—Ду (далее — метод Фридрих) [3]. В рамках данной системы ЦВЗ предлагается составлять из двух компонентов: хеш-суммы исходного изображения и опционального текстового сообщения.

На этапе внедрения ЦВЗ в изображение предлагаемая программная система выполняет следующие действия: 1) при помощи наперед заданной хеш-функции вычисляется хеш-сума исходного изображения, для которого необходимо обеспечить контроль целостности; 2) у пользователя системы запрашивается дополнительная текстовая информация, которая, в частности, может содержать данные, идентифицирующие пользователя; 3) вычисленная хеш-функция и текстовая информация образуют внедряемые в виде ЦВЗ данные; 4) данные, полученные на предыдущем этапе, внедряется в исходное изображение в соответствии с методом Фридрих. Параметрами встраивания (стего-ключом) при этом являются: амплитуда Flipping-функции метода Фридрих; размерность блоков, на которые делится изображение; цветовой канал, в который производится встраивание ЦВЗ.

На этапе контроля целостности изображения предлагаемая программная система выполняет следующие действия: 1) в соответствии с методом Фридрих ЦВЗ извлекается из изображения. При этом программная система выполняет возвращение изображения в состояние, которое оно имело до внедрения ЦВЗ; 2) из извлеченного ЦВЗ выделяется хеш-сумма; 3) выполняется подсчет хеш-суммы

для изображения, полученного на этапе 1; 4) хеш-сума, полученная на этапе 1 из извлеченного ЦВЗ, и хеш-сума, полученная на этапе 3, сравниваются. Если хеш-суммы совпадают, то целостность изображения, в которое был внедрен ЦВЗ, считается не нарушенной.

В разработанной программной системе в качестве метода хеширования был использован алгоритм MD5. Данный алгоритм позволяет получить 128-битную хеш-сумму (дайджест) изображения. Система позволяет при необходимости задействовать другой метод хеширования или осуществить выбор одного из нескольких подобных методов. Кроме того, имеется возможность добавления псевдослучайного или фиксированного модификатора к хеш-сумме с последующим повторным хешированием полученной последовательности данных.

Использованный в работе метод Фридрих требует выполнения сжатия специального двоичного RS-вектора, в который занесена информация о классификации блоков изображения. На этапе внедрения ЦВЗ в изображение блоки изображения классифицируются, а RS-вектор, хранящий результаты классификации, подвергается сжатию и внедряется в изображение. На этапе извлечения этот вектор считывается из изображения в составе ЦВЗ и подвергается декомпрессии. Для реализации сжатия вектора рассматривались следующие методы: классический метод Хаффмана, GZip, метод LZF [4]. Использование метода Хаффмана вызвало трудности из-за того, что этот метод выполняет построение кодового дерева по сжимаемым данным. Далее это дерево используется как на этапе сжатия, так и на этапе декомпрессии. Это означает, что для выполнения разжатия необходимо передать методу то же кодовое дерево, что использовалось при сжатии. Данное обстоятельство вызывает затруднение из-за необходимости передачи или хранения кодового дерева вместе с изображением, проверяемым на целостность. Алгоритм GZip не имеет недостатка предыдущего описанного метода — для сжатия и разжатия он не требует никаких дополнительных данных. Но при работе с ним имела место следующая проблема. При нарушении целостности изображения с внедренным ЦВЗ, возникает возможность того, что это нарушение изменит один или несколько заголовков, используемых алгоритмом. Например, архив, сжатый алгоритмом GZip, всегда начинается с последовательности байт «31, 139, 8 ...». Если при внесении изменений в изображение набор этих байт меняется, архив уже не удается распаковать. В итоге для реализации был выбран метод LZF, не имеющий упомянутых выше недостатков.

Разработка рассмотренной систем произведена на основе программной платформы .Net. При разработке кроме стандартных классов .Net использовались собственные классы — класс, реализующий алгоритм сжатия LZF, и класс, содержащий набор статических вспомогательных методов: метод разбиения изображения на набор блоков; метод, выполняющий для блока изображения Flipping-функцию; метод, реализующий функцию дискриминации; метод, позволяющий выполнить классификацию блоков и получить вектор RS; метод, который позволяет записать (считать) вектор RS в изображение (из изображения); метод, позволяющий восстановить исходное состояние изображения по разжатому RS-вектору.

В среде разработанной программной системы проведена серия экспериментов по контролю целостности изображений. В экспериментах были задействованы около пятидесяти изображений различного размера и природы. Результаты экспериментов показали эффективность предлагаемой системы при обнаружении фактов искажения и подмены изображений.

ИСПОЛЬЗОВАННЫЕ ИСТОЧНИКИ

1. Stamp M. Information Security: Principles and Practice. – New Jersey: Wiley, 2011.

2. Cox I., Miller M., Bloom J., Fridrich J. Digital watermarking and steganography.– Amsterdam: Morgan Kaufmann Publishers, Amsterdam, 2008.

3. Fridrich J., Goljan M., Du R. Lossless data embedding – new paradigm in digital watermarking // EURASIP Journal on Advances in Signal Processing.– 2002.– Vol. 2.– P. 185–196.

4. Сэломон Д. Сжатие данных, изображений и звука.– Москва: Техносфера, 2004.

K. V. Zashcholkin, I. G. Mileyko, A. A. Ishchenko

A program system for data integrity control based on a digital restoring watermark.

Approaches to the information object integrity control are considered in the paper. The authors offer a software system for bitmap images integrity control. The system is based on the introduction of a restoring digital watermark into the images.

Keywords: *data integrity control, digital watermark, hash sum.*