УДК 004.056.53 **DOI: 10.20998/2411-0558.2016.44.12**

К.В. ЗАЩЕЛКИН, канд. техн. наук, доц., ОНПУ, Одесса,
Е.Н. ИВАНОВА, ст. преп., ОНПУ, Одесса

АДАПТАЦИЯ МЕТОДА ВНЕДРЕНИЯ ЦИФРОВЫХ ВОДЯНЫХ ЗНАКОВ F5 К СРЕДЕ LUT-ОРИЕНТИРОВАННЫХ ИНФОРМАЦИОННЫХ КОНТЕЙНЕРОВ

Рассмотрены подходы к контролю информационных объектов при помощи технологии цифровых водяных знаков. Предложен метод внедрения цифрового водяного знака в информационные контейнеры с LUT-ориентированной архитектурой. Метод основан на использовании композиции известного метода F5, применяемого для контроля мультимедийных информационных объектов, и подходов к внедрению дополнительной информации в LUT-ориентированные контейнеры. Ил. 3. Библиогр. 17 назв.

Ключевые слова: контроль информационных объектов, цифровые водяные знаки, метод F5, LUT-ориентированные контейнеры.

Постановка проблемы. Технология цифровых водяных знаков (ЦВЗ) представляет собой подход к скрытому внедрению данных в информационный контейнер с целью контроля его целостности или легитимности использования. Технологии ЦВЗ основаны на применении стеганографических приемов [1], в рамках которых скрывается факт наличия ЦВЗ в информационном объекте (контейнере ЦВЗ). ЦВЗ может быть считан из контейнера при наличии стеганографического ключа (стего-ключа), определяющего правила доступа к элементам ЦВЗ.

В современных информационных системах ЦВЗ получили широкое распространение для контроля мультимедийного контента [2]: растровых графических файлов, видеофайлов, оцифрованного звука. Существенная особенность файлов-контейнеров для такого контента состоит в том, что все они являются *пассивными* информационными объектами, выполняющими только функцию хранения данных. Очевидно, что необходимость в подобном контроле использования и целостности информационных объектов не ограничивается только контейнерами данного вида. Такая необходимость имеет место и для *активных* информационных объектов, выполняющих некоторую вычислительную или управляющую функцию.

Исследования подходов к применению технологий ЦВЗ в отношении немультимедийных *активных* стего-контейнеров – информационных объектов, выполняющих определенную вычислительную или управляющую функцию, находятся сейчас на начальной стадии. В рамках таких подходов, например, предлагается

использовать в качестве стего-контейнеров для внедрения ЦВЗ, а также для задач скрытого хранения и пересылки защищенной информации, исполняемые файлы [3, 4] или исходные коды программ [5, 6] для вычислительных систем.

Анализ литературы. В работах [7 – 9] были предложены и развиты методы внедрения данных в стего-контейнеры с LUT-ориентированной архитектурой (LUT – Look Up Table – таблица поиска), отличающиеся от традиционных контейнеров тем, что является *активными* информационными объектами, состоящими из *неавтономных* элементарных единиц, данные в которых представлены *точно*. К таким контейнерам относятся, например, микросхемы FPGA (Field Programmable Gate Array) [10], являющиеся, на текущий момент, весьма используемой элементной базой для построения компьютерных и управляющих систем. Основным элементом таких контейнеров выступают блоки LUT, которые представляют собой структуру данных, используемую с целью замены вычислений на операции поиска заготовленных данных [11]. Блоки LUT в FPGA обычно представлены в виде одноразрядной оперативной памяти. Входы блока LUT при этом являются адресными входами такой памяти. При количестве входов, равном n, блок LUT хранит в себе 2^n бит информации и способен выполнить вычисление значения одной n-аргументной булевой функции.

Одним из эффективных подходов к внедрению ЦВЗ в мультимедийные информационные контейнеры является использование совместно с классическим стеганограическими приемами теории помехоустойчивого кодирования [12]. Методы внедрения ЦВЗ, использующие такой подход отличаются низкими значениями отношения количества искаженных (в результате внедрения) элементарных единиц контейнера к общему количеству элементарных единиц. Наиболее широкое применение среди таких методов, получил метод F5 [12 – 14] и его модификации. Этот метод сочетает технику синдромного декодирования вместе с традиционными подходами к внедрению ЦВЗ. Однако указанный подход пока не распространен на существующие методы встраивания ЦВЗ в информационные контейнеры с LUT-ориентированной архитектурой.

Цель данной работы состоит в развитии методов встраивания ЦВЗ в информационные контейнеры с LUT-ориентированной архитектурой путем адаптации метода F5 к среде указанных контейнеров.

Основная часть. Предлагаемый в данной работе метод является композицией подходов известного метода F5 [12 – 14] в части использования синдромного декодирования и методов, предложенных

авторами данной статьи в работах [7 – 9] в части встраивания ЦВЗ в LUT-ориентированные контейнеры.

Последовательность действий классического метода F5 [14] состоит в следующем. Встраиваемая в контейнер двоичная последовательность разрядов ЦВЗ разделяется на *m*-разрядные двоичные векторы. Каждый из таких векторов рассматривается как синдром ошибки S^*_i для (n, k)-кода Хемминга с заданной проверочной матрицей **H**. Рабочая область контейнера ЦВЗ интерпретируется как последовательность *n*-разрядных векторов a_i, обладающих синдромом ошибки S_i, полученным по той же проверочной матрице **H**. Процедура встраивания ЦВЗ в соответствии с методом F5 заключается в замене синдромов векторов контейнера S_i на синдромы S^*_i ЦВЗ.

Предлагаемая адаптация данного метода к среде LUT-контейнеров предполагает следующие исходные данные, для реализации метода:

1) $M = (m_1, m_2, \ldots, m_r)$ – двоичная последовательность ЦВЗ, подлежащая встраиванию в контейнер;

2) *LC* – контейнер с LUT-ориентированной архитектурой, реализующий некоторую вычислительную или управляющую функцию;

3) *key* = (*set, coding, order*) – ключ для внедрения и извлечения ЦВЗ, где *set* – номер (адрес) разряда LUT, в который выполняется внедрение бита последовательности ЦВЗ;

coding = (*n, k, ERule*) – формальное описание принципов используемого помехоустойчивого кодирования: *n, k* – параметры помехоустойчивого (*n, k*)-кода, *n* – длина кодового слова, *k* – количество информационных разрядов в кодовом слове; *ERule* – правило выполнения синдромного декодирования (может быть представлено проверочной матрицей **H** блочного кода или иным описанием процедуры получения синдрома ошибки);

order – правило, задающее стего-путь – порядок обхода блоков LUT в контейнере для выполнения встраивания или извлечения информации.

Результат применения предлагаемого метода – LUT-контейнер *LC**, в который внедрена последовательность *M*. При этом функционирование контейнера *LC**, выражающееся в выполнении его целевой функции, не должно отличаться от функционирования контейнера *LC*, структура и параметры у контейнеров *LC** и *LC* также не должны отличаться.

Последовательность действий предлагаемого метода внедрения данных в LUT-ориентированный контейнер состоит из семи этапов.

Этап 1. Рабочая область LUT-контейнера определяется стего-путем, образованным парами последовательно соединенных блоков LUT, что определяется принципами, предложенными в работах [7 – 9].

Этап 2. Полученный стего-путь разделяется на сегменты по *n* пар блоков LUT (параметр *n* является одним из компонентов стего-ключа *key*).

Этап 3. Для каждой из полученных на предыдущем этапе пар блоков выполняется считывание значения разряда с номером *set* из внутреннего двоичного кода первых блоков пар. Совокупность полученных таким образом n разрядов каждого из сегментов стего-пути рассматривается как последовательность n-разрядных векторов a_i.

Этап 4. Для каждого из двоичных векторов a_i, полученных на предыдущем этапе, определяется синдром ошибки S_i. Синдром определяется в соответствии с правилом синдромного декодирования *ERule*, заданным стего-ключем.

Этап 5. Определяется количество контрольных разрядов используемого кода $m = n - k$. Двоичная последовательность, подлежащая встраиванию в контейнер $M = (m_1, m_2, \dots, m_r)$ разбивается на сегменты по m разрядов. Каждый из полученных сегментов интерпретируется как синдром ошибки S^*_i для (n, k)-кода, который (синдром) определяется по тем же правилам *ERule*, что и на этапе 4.

Этап 6. Непосредственно процедура встраивания информации ЦВЗ представляет собой последовательное внедрение сегментов скрываемой двоичной последовательности S^*_i (полученных на этапе 5) в двоичные вектора a_i, полученные на этапе 3. Такое внедрение достигается заменой каждого вектора a_i, имеющего синдром S_i, на вектор a^*_i, имеющий синдром S^*_i. Для выполнения указанной замены векторов текущий (S_i) и требуемый (S^*_i) синдромы суммируются по модулю два: $P_i = S_i \oplus S^*_i$. Полученное значение P_i задает вектор ошибки e_i с весом Хемминга $w(e_i) = 1$. Для кода Хемминга P_i соответствует единственному столбцу проверочной матрицы **H** кода, порядковый номер которого задает номер единичного разряда в векторе ошибки e_i.

Этап 7. Изменение значения двоичного вектора i-го сегмента стего-пути ($a^*_i = a_i \oplus e_i$) производится в соответствии с правилами, изложенными в работах [7 – 9], которые базируются на принципах эквивалентных преобразований на множестве пар последовательно подключенных блоков LUT, предложенных в работах [15, 16]. Такое изменение достигается путем выполнения двух действий в отношении блока LUT, содержащего разряд вектора a_i который подвергается изменению: а) инвертирования внутреннего кода текущего обрабатываемого блока LUT; б) выполнения распространения инверсии на входы всех блоков LUT, подключенных к выходу текущего блока. Второе из указанных действий компенсирует изменения кода блока LUT, выполненные в ходе первого действия.

Рассмотрим пример выполнения указанных этапов предлагаемого метода внедрения ЦВЗ в LUT-ориентированный контейнер. Исходные данные примера:

– двоичная последовательность ЦВЗ, которую необходимо внедрить в контейнере $M = 011001010\ldots$;

– номер (адрес) разряда LUT, в который выполняется внедрение битов последовательности ЦВЗ *set* = 5;

– параметры применяемого помехоустойчивого кода: $n = 7$, $k = 4$;

– правило выполнения синдромного декодирования представлено в виде проверочной матрицы **H** кода Хемминга:

$$ERule = \mathbf{H} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

Для указанных исходных данных, процесс выполнения предложенного метода состоит в следующем.

Этап 1. Пусть для данного примера стего-путь, определяемый компонентом стего-ключа *order* устанавливает последовательность блоков LUT, представленную на рис. 1. Для компактности изложения показано начало этой последовательности, состоящее из восьми блоков LUT первого уровня: $LUT_{1.1}$ – $LUT_{1.8}$. Каждый из блоков LUT первого уровня подключен к одному из блоков LUT второго уровня: $LUT_{2.1}$ – $LUT_{2.4}$ (требование ограничений, определенных в работах [7, 9]).

Этап 2. Поскольку по условиям данного примера для внедрения ЦВЗ используется (n, k)-кода Хемминга с параметром $n = 7$, разбиваем последовательность блоков LUT, входящих в стего-путь на сегменты по семь пар блоков LUT в каждом. Для данного примера в первый сегмент входят следующие семь пар блоков:

$$\text{Сегмент 1} = ([LUT_{1.1}, LUT_{2.1}], ([LUT_{1.2}, LUT_{2.2}], ([LUT_{1.3}, LUT_{2.2}], [LUT_{1.4}, LUT_{2.2}], [LUT_{1.5}, LUT_{2.3}], [LUT_{1.6}, LUT_{2.4}], [LUT_{1.7}, LUT_{2.4}]).$$

Этап 3. Из кода первого блока, каждой из полученных семи пар блоков считываем значение разряда с номером *set*, который по условиям данного примера равен 5. На рис. 1 все разряды с порядковым номером 5 в кодах блоков LUT первого уровня показаны выделением. Совокупность выделенных разрядов образует двоичный вектор $a_1 = 0101110$.

Этап 4. Используя проверочную матрицу **H**, заданную компонентом стего-ключа *ERule* находим синдром ошибки S_i для вектора a_1, полученного на предыдущем этапе: $S_1 = a_1 \cdot \mathbf{H}^T = 101$.

Этап 5. Определяем количество контрольных разрядов используемого кода $m = n - k$. Для заданных по условиям примера параметров $n = 7$, $k = 4$ параметр m имеет значение 3. Двоичную последовательность ЦВЗ $M = 011001010\ldots$ разбиваем на сегменты по m разрядов: $M = 011|001|010|\ldots$. Далее из-за ограниченности объема

примера, покажем встраивание в LUT-контейнер только первого трехразрядного сегмента ЦВЗ "011".

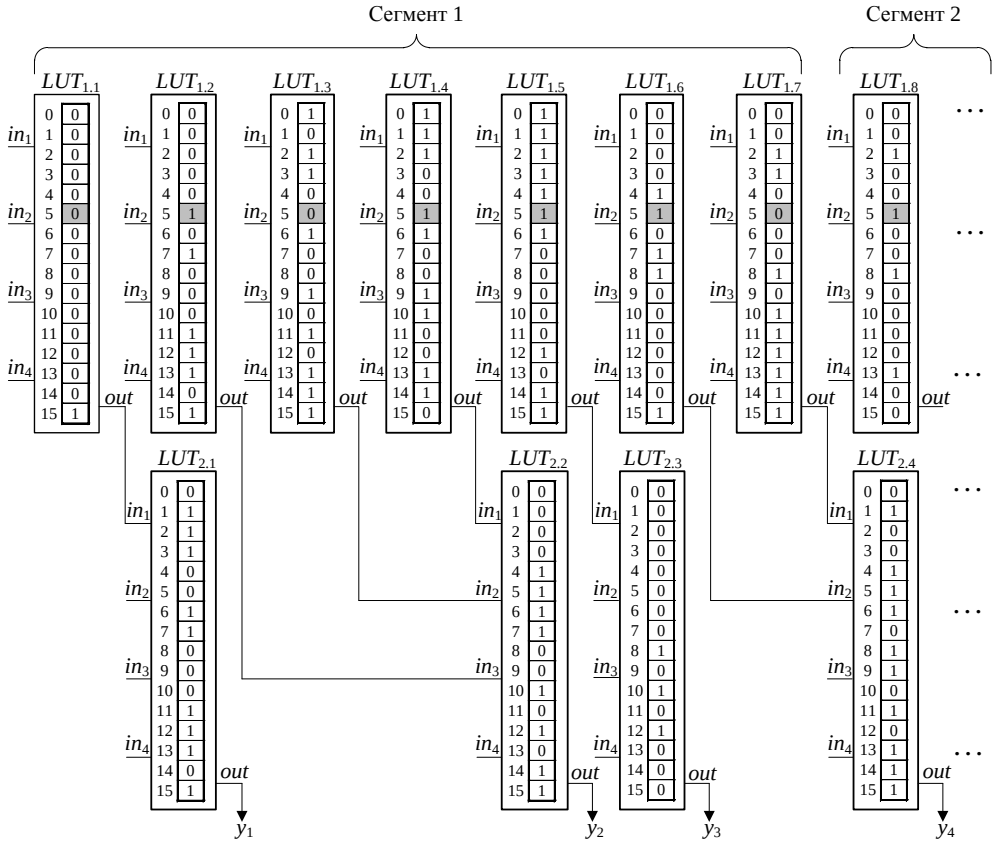

Рис. 1. Пример фрагмента LUT-контейнера до встраивания ЦВЗ

Этап 6. Для встраивания первого сегмента ЦВЗ "011" в контейнер необходимо в векторе $a_1 = 0101110$ заменить текущий синдром $S_1 = 101$ на синдром, определяемый значением сегмента ЦВЗ $S^*_1 = 011$. Для этого следует получить вектор ошибки e_1, который, будучи наложенным на вектор a_1, изменит его синдром с S_1 на S^*_1. Суммируем текущий и требуемый синдромы поразрядно по модулю два $P_1 = S_1 \oplus S^*_1 = 110$. Полученное значение P_1 задает вектор ошибки e_1 с весом Хемминга $w(e_i) = 1$. P_1 соответствует единственному столбцу проверочной матрицы **H** кода, порядковый номер которого задает номер единичного разряда в векторе ошибки e_1. Значение $P_1 = 110$ совпадает со значением предпоследнего (6-го) столбца проверочной матрицы **H** (вес разрядов в столбце матрицы возрастает в направлении сверху-вниз). Это означает, что вектор ошибки содержит единицу в предпоследнем разряде и имеет вид $e_1 = 0000010$. Таким образом, для внедрения сегмента ЦВЗ "011" в

контейнер необходимо инвертировать предпоследний разряд вектора $a_1 = 0101110$ с тем, чтобы получить вектор $a^*_1 = a_1 \oplus e_1 = 0101100$. Действительно, вектор $a^*_1 = 0101100$, который отличается от исходного вектора $a_1 = 0101110$ только в одном (предпоследнем) разряде, имеет синдром ошибки $S_1 = a^*_1 \cdot \mathbf{H}^T = 011$, совпадающий со значением сегмента ЦВЗ, который необходимо встроить в контейнер.

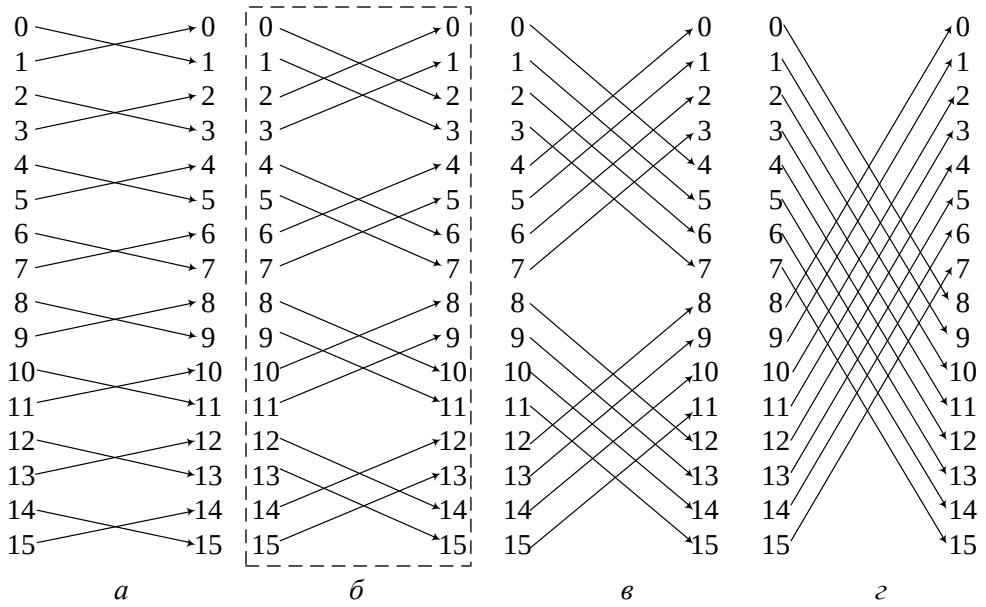

Рис. 2. Принцип выполнения перестановок значений разрядов кода в процессе распространения инверсии на входы 4-х входового блока LUT
а – распространение на вход с весом 1; *б* – распространение на вход с весом 2; *в* – распространение на вход с весом 4; *г* – распространение на вход с весом 8

Этап 7. На предыдущем этапе было установлено, что для встраивания сегмента ЦВЗ "011" в LUT-контейнер, необходимо в предпоследнем (6-м) разряде вектора a_1 значение 1 поменять на значение 0. Этот разряд находится в LUT-контейнере коде блока $LUT_{1.6}$ по адресу $set = 5$ (рис. 1). Изменение значения этого разряда в блоке LUT выполняем в соответствии с правилами, изложенными в работах [7 – 9] т.е. путем инвертирования внутреннего кода текущего обрабатываемого блока LUT и выполнения распространения инверсии на входы всех блоков LUT, подключенных к выходу текущего блока. В данном примере весь код блока $LUT_{1.6}$ инвертируется. Для компенсации этой инверсии разряды кода блока $LUT_{2.4}$, подключенного к выходу блока $LUT_{1.6}$ меняются местами в соответствии со следующим принципом. В схеме, изображенной на рис. 1, входы блоков LUT имеют двоичные веса,

пропорциональные величине номера входа, т.е. вход in_1 имеет вес 1, вход in_2 – вес 2, вход in_3 – вес 4 и вход in_4 – вес 8. Блок $LUT_{1.6}$ подключен к входу in_2 блока $LUT_{2.4}$, который имеет вес 2. Таким образом, необходимо распространить инверсию блока $LUT_{1.6}$ на вход блока $LUT_{2.4}$ с весом 2. Такое распространение инверсии выполняется путем перестановки между собой, рядом расположенных пар разрядов кода (рис. 2, *б*).

На рис. 2 показаны принципы перестановок разрядов кодов блока LUT в ходе распространения инверсии на различные входы блока. Вариант, рассматриваемый в данном примере, показан выделением.

Результирующий вид фрагмента LUT-контейнера с встроенным в него сегментом ЦВЗ "011" показан на рис. 3.

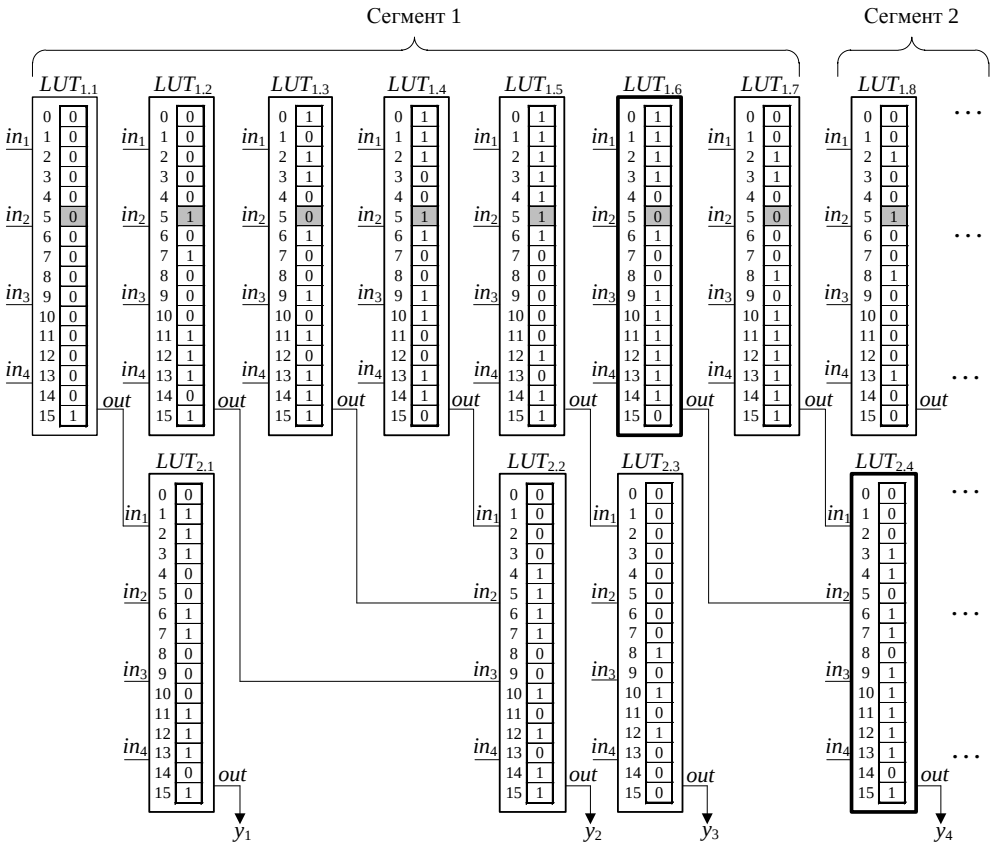

Рис. 3. Фрагмент LUT-контейнера после встраивания в него ЦВЗ

В результате встраивания ЦВЗ структура контейнера осталась неизменной. Были заменены коды только одной из семи пар блоков LUT, находящихся в первом сегменте стего-пути (указанная пара блоков показана на рис. 3 выделением). Это позволило скрыть в данный сегмент трехразрядную часть кода ЦВЗ. Функционирование контейнера при этом

не претерпело изменений. Модификация кодов блоков $LUT_{1.6}$ и $LUT_{2.4}$ не привела к изменению целевой функции контейнера и, в частности, к изменению поведения контейнера по выходу y_4.

Выводы. Предложенный в работе метод позволяет выполнять внедрение ЦВЗ в контейнеры с LUT-ориентированной архитектурой. Метод основан на композиции методов внедрения ЦВЗ в LUT-контейнеры, предложенных авторами данной работы и известного метода F5, ориентированного на работу с пассивными мультимедийными контейнерами. Предложенный метод не меняет связи блоков LUT между собой в пределах контейнера, однако выполняет эквивалентные преобразования значений кодов отдельных блоков. Такие эквивалентные изменения кодов не изменяют целевую функцию LUT-контейнера и, как было показано в работе [17], не влияют на его основные параметры. Предложенный метод отличается от существующих методов встраивания ЦВЗ в LUT-контейнеры низким значением показателя отношения количества изменяемых (в результате встраивания) блоков LUT контейнера к общему количеству блоков. Так в рассмотренном в работе примере для внедрения в контейнер трехразрядного сегмента ЦВЗ понадобилось внести изменения только в одну из семи пар блоков LUT, входящих в сегмент стего-пути.

Предложенный метод может найти применение при разработке аппаратно-программного обеспечения, реализующего внедрение цифровых водяных знаков в вычислительные и управляющие устройства, построенные на основе LUT-ориентированной элементной базы (например, FPGA или программируемых логических интегральных схем со схожими архитектурами).

Список литературы: **1.** *Shih F.* Multimedia Security: Watermarking, Steganography, and Forensics / *F. Shih.* – CRC Press, 2013. – 424 p. **2.** *Cox I.* Digital Watermarking and Steganography / *I. Cox, M. Miller, J. Bloom, J. Fridrich*. – Burlington: Morgan Kaufmann Publishers, 2008. – 592 p. **3.** *Skoudis E.* Malware: Fighting Malicious Code / *E. Skoudis, L. Zeltser*. – New Jersey: Prentice Hall, 2004. – 672 p. **4.** *Hamilton A.* Survey of Static Software Watermarking / *A. Hamilton, S. Danicic* // Proceedings of Internet Security World Congress (WorldCIS-2011). – London, 2011. – P. 100-107. **5.** *Hakun L.* New approaches for software watermarking by register allocation / *L. Hakun, K. Keiichi* // Proceedings of the ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel Distributed Computing. – 2008. – P. 63-68. **6.** *Xiao Cheng L.* Software Watermarking Algorithm Based on Register Allocation / *L. Xiao Cheng, C. Zhiming* // Proceedings of International Symposium Distributed Computing and Applications to Business Engineering and Science (DCABES). – Hong Kong, 2010. – P. 539-543. **7.** *Защелкин К.В.* Метод внедрения цифровых водяных знаков в аппаратные контейнеры с LUT-ориентированной архитектурой / *К.В. Защелкин, Е.Н. Иванова* // Информатика и математические методы в моделировании. – Одесса, 2013. – Т. 3. – № 4. – С. 369-384. **8.** *Защелкин К.В.* Метод стеганографического скрытия данных в LUT-ориентированных аппаратных контейнерах / *К.В. Защелкин, Е.Н. Иванова* // Електротехнічні та комп'ютерні системи. – Київ, 2013. – Вип. 12 (88). – С. 83-90. **9.** *Защелкин К.В.* Развитие

метода стеганографического скрытия данных в LUT-ориентированных аппаратных контейнерах / *К.В. Защелкин, Е.Н. Иванова* // Електротехнічні та комп'ютерні системи. – Київ, 2014. – Вип. 13 (89). – С. 231-239. **10.** *Грушвицкий Р.И.* Проектирование систем на микросхемах с программируемой структурой / *Р.И. Грушвицкий, А.Х. Мурсаев, Е.П. Угрюмов.* – СПб.: БХВ, 2010. – 650 с. **11.** *Paul S.* Reconfigurable Computing Using Content Addressable Memory for Improved Performance and Resource Usage / *S. Paul, S. Bhunia* // Proceedings of Design Automation Conference ACM/IEEE (DAC-2008). – Anaheim, 2008. – P. 786-791. **12.** *Fridrich J.* Steganography in Digital Media / *J. Fridrich*. – Cambridge University Press, 2010. – 438 p. **13.** *Беззатеев С.В.* Специальные классы кодов для стеганографических систем / *С.В. Беззатеев, Н.В. Волошина, К.А. Жиданов* // Доклады Томского государственного университета систем управления и радиоэлектроники. – Томск, 2012. – № 1-2 (25). – С. 112-118. **14.** *Westfeld A.* F5 – A Steganographic Algorithm. High Capacity Despite Better Steganalysis / *A. Westfeld* // Information Hiding. 4-th International Workshop. – Berlin: Springer-Verlag, 2001. – Vol. 2137. – P. 289-302. **15.** *Drozd A.V.* Use of Natural LUT Redundancy to Improve Trustworthiness of FPGA Design / *A.V. Drozd, M.A. Drozd, M.A. Kuznietsov* // Proceedings of the 12th International Conference ICTERI-2016. – Kiev, Ukraine, 2016. – P. 322-331. **16.** *Дрозд Ю.В.* Естественные ресурсы компьютерных систем на базе FPGA / *Ю.В. Дрозд, А.В. Дрозд, Н.А. Кузнецов* // Труды Одесского политехнического университета. – Одесса, 2013. – Вып. 2 (41). – С. 223-226. **17.** *Защелкин К.В.* Исследование основных характеристик FPGA-проектов при изменении кодов в блоках LUT // *К.В. Защелкин, Кузнецов Н.А., Дрозд А.В.* // Науковий вісник Чернівецького університету. Серія "Комп'ютерні системи та компоненти". – Чернівці, 2014. – Т. 5. – Вип. 2. – С. 41-45.

References:

1. Shih, F. (2013), *Multimedia Security: Watermarking, Steganography, and Forensics*, CRC Press, Boston, 424 p.

2. Cox, I., Miller, M., Bloom, J. and Fridrich, J. (2008), *Digital Watermarking and Steganography*, Morgan Kaufmann Publishers, Burlington, 592 p.

3. Skoudis, E. and Zeltser, L. (2004), *Malware: Fighting Malicious Code*, Prentice Hall, New Jersey, 672 p.

4. Hamilton, A. and Danicic, S. (2011), "Survey of Static Software Watermarking", *Proceedings of Internet Security World Congress (WorldCIS-2011)*, 2011, London, pp. 100-107.

5. Hakun, L. and Keiichi, K. (2008) "New approaches for software watermarking by register allocation", *Proceedings of the ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel Distributed Computing*, 2008, Washington, DC, USA, pp. 63-68.

6. Xiao Cheng, L. and Zhiming, C. (2010), "Software Watermarking Algorithm Based on Register Allocation", *Proceedings of International Symposium Distributed Computing and Applications to Business Engineering and Science (DCABES)*, 2010, Hong Kong, pp. 539-543.

7. Zashchelkin, K.V. and Ivanova, E.N. (2013), "Method of embedding digital watermarks in hardware containers with LUT-oriented architecture", *Informatics and Mathematical Methods in Simulation*, Vol. 3, No 4, pp. 369-384.

8. Zashchelkin, K.V. and Ivanova, E.N. (2013), "Method of steganographical data hiding in LUT-oriented hardware containers", *Electrotechnic and Computer Systems*, Vol. 12 (88), pp. 83-90.

9. Zashchelkin, K.V. and Ivanova, E.N. (2014), "Method development for implementing the steganography data hiding in LUT-oriented hardware containers", *Electrotechnic and Computer Systems*, Vol. 13 (89), pp. 231-239.
10. Grushvitsky, R.I., Mursaev, A.H. and Ugryumov, E.P. (2010), *Designing systems on chips with programmable structure, BHV*, St. Petersburg, Russian Federation, 650 p.
11. Paul S. and Bhunia S. (2008), "Reconfigurable Computing Using Content Addressable Memory for Improved Performance and Resource Usage", *Proceedings of Design Automation Conference ACM/IEEE (DAC-2008)*, Anaheim, pp. 786-791.
12. Fridrich J. (2010), *Steganography in Digital Media*, Cambridge University Press, New York, 438 p.
13. Bezzateev, S.V., Voloshina, N.V. and Zhidanov, K.A. (2012), "Special class of error correcting codes for steganographic systems", *Reports of the Tomsk State University of Control Systems and Radio Electronics*, No 1-2 (25), pp. 112-118.
14. Westfeld, A. (2001), "F5 – A Steganographic Algorithm. High Capacity Despite Better Steganalysis", *Information Hiding. 4-th International Workshop*, 2001, Berlin, Vol. 2137, pp. 289-302.
15. Drozd, A.V., Drozd, M.A. and Kuznietsov, M.A. (2016), "Use of Natural LUT Redundancy to Improve Trustworthiness of FPGA Design", *Proceedings of the 12th International Conference ICTERI-2016*, 2016, Kiev, Ukraine, pp. 322-331.
16. Drozd, J.V., Drozd, A.V. and Kuznetsov, N.A. (2013), "Natural resources of computer systems based on FPGA", *Proceedings of Odessa Polytechnic University*, Vol. 2 (41), pp. 223-226.
17. Zashchelkin, K.V., Kuznetsov, N.A. and Drozd, A.V. (2014), "Research in key features of FPGA-projects in case of changing the codes in LUT-blocks", *Scientific Herald of Chernivtsy University, Series: Computer systems and components*, Vol. 5, No 2, pp. 41-45.

Статью представил д-р техн. наук, проф. Одесского национального политехнического университета Ситников В.С.

Поступила (received) 25.08.2016

Zashcholkin Konstantin, PhD, Tech., Associate Professor
Odessa National Polytechnic University
Ave. Shevchenko, 1, Odessa, Ukraine, 65044
Tel.: (048) 734-83-22, e-mail: const-z@te.net.ua
ORCID ID: 0000-0003-0427-9005

Ivanova Elena, Senior Lecturer
Odessa National Polytechnic University
Ave. Shevchenko, 1, Odessa, Ukraine, 65044
Tel.: (048) 734-83-91, e-mail: enivanova@ukr.net
ORCID ID: 0000-0002-4743-6931

УДК 004.056.53

Адаптація метода вбудовування цифрових водяних знаків F5 до середовища LUT-орієнтованих інформаційних контейнерів / Защолкін К.В., Іванова О.М. // Вісник НТУ "ХПІ". Серія: Інформатика та моделювання. – Харків: НТУ "ХПІ". – 2016. – № 44 (1216). – С. 135 – 146.

Розглянуто підходи до контролю інформаційних об'єктів за допомогою технології цифрових водяних знаків. Запропоновано метод вбудовування цифрового водяного знака в інформаційні контейнери з LUT-орієнтованою архітектурою. Метод базується на використанні композиції відомого методу F5, що застосовується для контролю мультимедійних інформаційних об'єктів, і підходів до вбудовування додаткової інформації в LUT-орієнтовані контейнери. Іл. 3. Бібліогр. 17 назв.

Ключові слова: контроль інформаційних об'єктів, цифрові водяні знаки, метод F5, LUT-орієнтовані контейнери.

УДК 004.056.53

Адаптация метода внедрения цифровых водяных знаков F5 к среде LUT-ориентированных информационных контейнеров / Защелкин К.В., Иванова Е.Н. // Вестник НТУ "ХПИ". Серия: Информатика и моделирование. – Харьков: НТУ "ХПИ". – 2016. – № 44 (1216). – С. 135 – 146.

Рассмотрены подходы к контролю информационных объектов при помощи технологии цифровых водяных знаков. Предложен метод внедрения цифрового водяного знака в информационные контейнеры с LUT-ориентированной архитектурой. Метод основан на использовании композиции известного метода F5, применяемого для контроля мультимедийных информационных объектов, и подходов к внедрению дополнительной информации в LUT-ориентированные контейнеры. Ил. 3. Библиогр. 17 назв.

Ключевые слова: контроль информационных объектов, цифровые водяные знаки, метод F5, LUT-ориентированные контейнеры.

UDC 004.056.53

Adaptation of the F5 method for embedding digital watermarks into the environment of LUT-oriented information containers // Zashcholkin K.V., Ivanova E.N. // Herald of the National Technical University "KhPI". Subject issue: Informatics and Modelling. – Kharkov: NTU "KhPI". – 2016. – № 44 (1216). – P. 135 – 146.

The approaches to the information objects control have been considered by means of digital watermarking technology. We propose the method for embedding the digital watermark into containers with LUT-oriented architecture. The method is based on composition of well-known F5 method, which is implemented to control multimedia information objects, and approaches to embedding additional information into LUT-oriented containers. Figs.: 3. Refs.: 17 titles.

Keywords: information objects control, digital watermarks, F5 method, LUT-oriented containers.