УДК 003.26 004.7 004.9

Исследование и разработка алгоритма защиты проектной документации в CAD/CAM/CAE от несанкционированного доступа

© Т.М. Волосатова, Н.В. Чичварин

МГТУ им. Н.Э. Баумана, Москва, 105005, Россия

Приведены результаты анализа основных методов стеганографической защиты документов (цифровых изображений). Предложены метод и алгоритм стеганографического сокрытия данных, приемлемые для решения задачи защиты проектной документации от несанкционированного доступа. Описан разработанный и реализованный метод стеганографической защиты документации в QR-кодах. Проведены анализ и оценка читаемости QR-кода при большом объеме сообщения с использованием двух стеганографических алгоритмов: алгоритма Коча и алгоритма Бенхама.

Ключевые слова: *проектный документ, защита, несанкционированный доступ, QR-код, стеганография, алгоритм Коча, алгоритм Бенхама.*

Введение. В настоящее время, когда происходит интеграция различных видов систем автоматизированного проектирования (САПР), таких как CAD (Computer Aided Design) — система конструкторского проектирования), CAM (Computer Aided Manufacturin) — автоматизированные системы проектирования технологических процессов и CAE (Computer Aided Engineering — системы расчетов и инженерного анализа), возникает проблема защиты проектной документации от несанкционированного доступа (НСД). Как показывает обзор доступных публикаций [1–3], задача защиты проектной документации от НСД становится все более актуальной, особенно в связи с нарастающей популярностью САПР, опирающихся на PLM-технологии (Product Lifecycle Management — управление жизненным циклом изделия). Тем не менее число разработок в этой области невелико. Известные аппаратно-программные средства [3] не поддерживают CALS-технологии (Continuous Acquisition and Lifecycle Support — непрерывная информационная поддержка поставок и жизненного цикла изделий). Аналитический обзор публикаций показывает, что сегодня вопросам защиты проектной документации в САПР не уделяется достаточного внимания. В то же время, вероятность угрозы информационной безопасности проектных организации неуклонно возрастает. Например, эксперты компании SDRC установили, что любой пользователь мог получить доступ к ядру системы с правами администратора. Известно также, что у американской аэрокосмической корпорации Lockheed Martin в 1997 г. была украдена проектная документация самолета-невидимки Stealth.

К основным угрозам при работе с CALS-продуктами можно отнести утечку конфиденциальной информации и нарушение работоспособности системы. Последняя угроза может быть реализована с помощью и атак, рассчитанных на отказ в обслуживании и выводящих из строя отдельные элементы CAD-систем, и с помощью вирусов и «червей», заражающих САПР. Так, еще в 2000 г. был обнаружен первый вирус ACAD.Star для AutoCAD. Производители CALS-продуктов не уделяют должного внимания вопросам безопасности их использования, особенно на этапе опытной эксплуатации, что не позволяет задействовать встроенные возможности самих систем.

В настоящей работе предлагаются метод и алгоритм стеганографического сокрытия данных, которые, на наш взгляд, приемлемы для решения задачи защиты проектной документации от НСД. Исследования, проведенные с учетом обзоров в публикациях [4–13], позволяют подтвердить это положение.

Алгоритм реализуется за два этапа:

• формируется массив данных, подлежащих сокрытию. Массив может быть небольшой размерности и содержать текстовые данные о конструктивных параметрах «слепого чертежа», например, обозначение и распиновка микросхемы;

• в контейнер в виде QR-кода (*quick response* — быстрый отклик), доступного злоумышленнику текста (либо в открытую документацию) заносится текстовой массив любым из известных стеганографическим методом.

1. Описание предлагаемого алгоритма. ***Описание контейнера и обсуждение возможности его применения в качестве контейнера.*** Как известно, QR-код — матричный код, разработанный и представленный японской компанией Denso-Wave в 1994 г. Самый большой (версия 40) QR-код составляет 177×177 пикселов. Алфавитно-цифровая кодировка этих кодов поддерживает 10 цифр, буквы от *A* до *Z* и несколько специальных символов (11 бит на два символа, до 4296 символов), а байтовая, когда данные в любой подходящей кодировке (по умолчанию ISO 8859-1), — до 2953 байт. Кроме того, существуют «псевдокодировки»: задание способа кодировки в данных, разбиение длинного сообщения на несколько кодов и т. д. Для исправления ошибок применяется код Рида — Соломона с восьмибитным кодовым словом. Есть четыре уровня избыточности: 7, 15, 25 и 30 %. Благодаря исправлению ошибок удается нанести на QR-код рисунок и все равно оставить его читаемым. Эти сведения были проверены экспериментально. Область данных защищена кодером, реализующим оператор XOR со специальной маской. Кодер перебирает все варианты масок, подсчитывая штрафные очки для каждой по особым правилам, и выбирает самую удачную. На рис. 1 приведен пример искаженного изображения кода. Оценка читаемости QR-кода при больших объемах сообщения показала, что серьезного влияния

объема кодируемых данных на распознавание QR-кода не отмечено. Он распознавался с помощью мобильного телефона Galaxy S3 на базе Android.

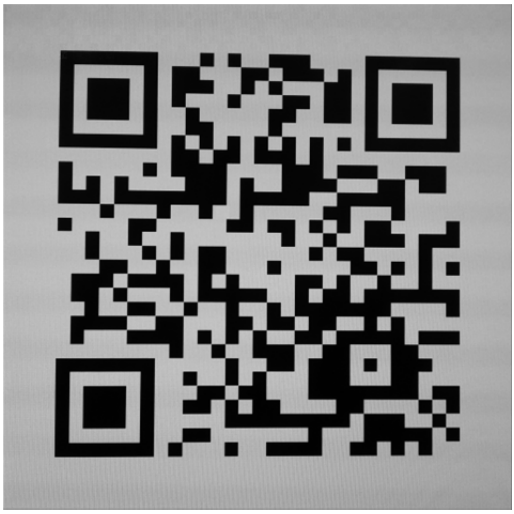


Рис. 1. QR-код с сообщением (“123123 PROVERKAPROVERKA123123QWERTY1213 QWERTY124”), который полностью восстанавливается

На рис. 2 приведены результаты оценки влияния искажений на устойчивость QR-кода.

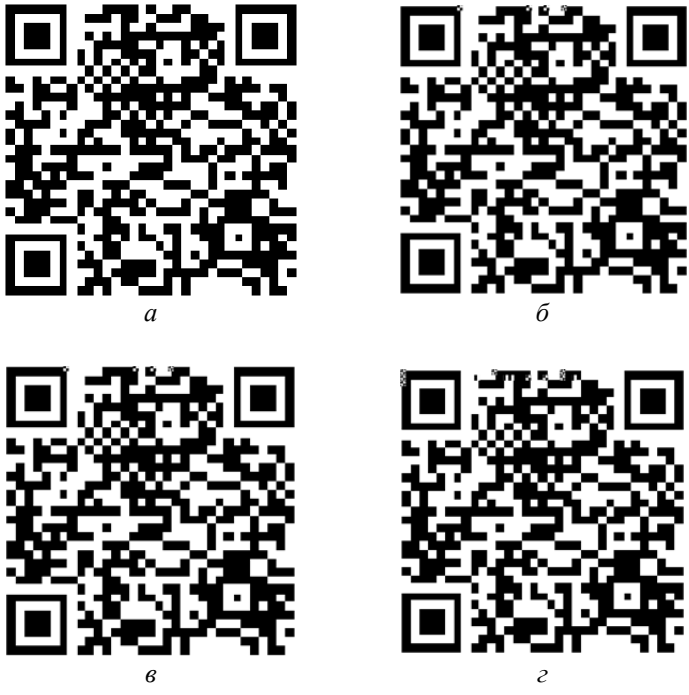
На рис. 3 показана схема подключения накопителя FLASH на микроконтроллере 74HC244.

Рисунок 4 наглядно показывает, что «слепая» принципиальная электрическая схема с трудом поддается аппаратной реализации, а поскольку отсутствует обозначение USB, затруднено и определение назначения схемы. Таким образом, очевидно, что воспроизведение устройства по такой схеме почти невозможно. Задача восстановления еще сложнее, если защитить от НСД код программы, составляющий бóльшую часть коммерческой тайны.

Восстановление становится невозможным, если защитить от НСД код микропрограммы. Существует по крайней мере два способа сокрытия кода и соответственно защиты данных:

• путем установки специального *бита защиты* или нескольких бит в слове конфигурации микроконтроллера. Физически эти биты располагаются в специальных ячейках памяти на кристалле микроконтроллера;

• любым из известных методов стеганографии. Предпочтительно сокрытие совместно с QR-кодом с помощью алгоритма, описанного выше.



а *б* *в* *г*

Рис. 2. Результаты оценки влияния искажений на устойчивость QR-кода. Длина встраиваемых данных, символы: *а* — идеальный код; *б* — 100; *в* — 200; *г* — 400

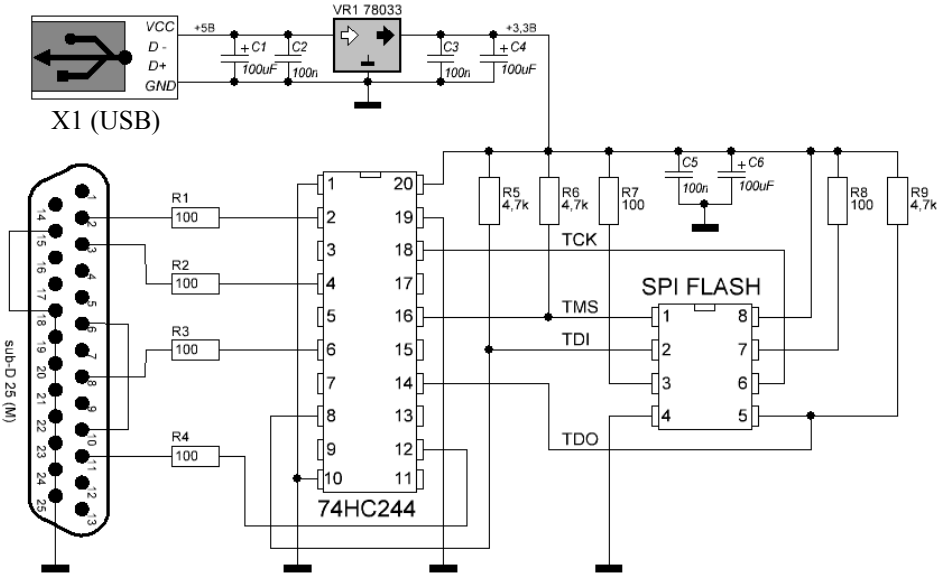


Рис. 3. Схема подключения накопителя FLASH на микроконтроллере 74HC244

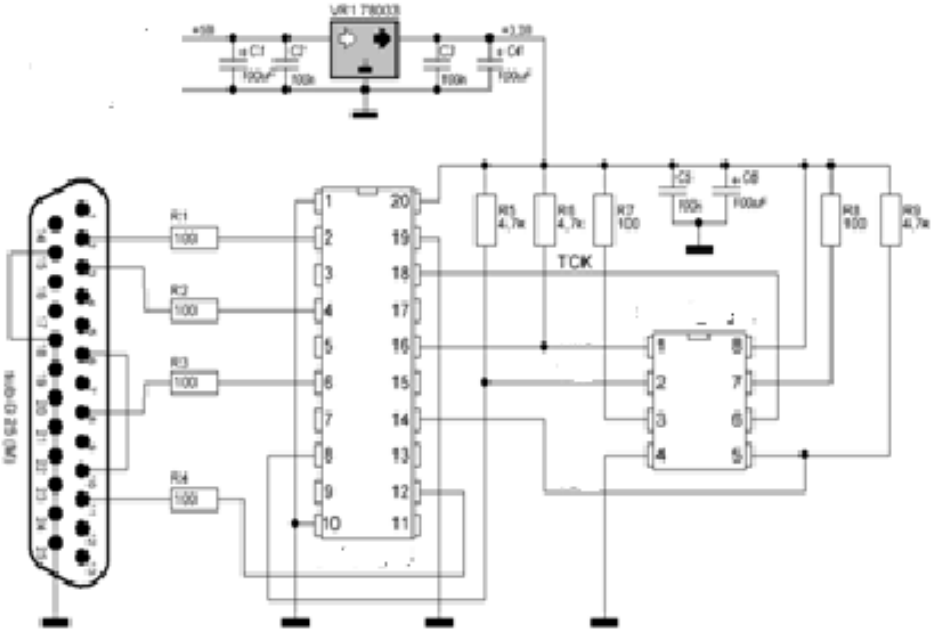


Рис. 4. «Слепая» схема, восстановленная после QR-кодирования

Примеры существенных фрагментов кода микропрограммы для устройства подключения FLASH к USB приведены ниже.

Текст кода на микроассемблере:

```
LIST p=16F84a include "P16F84A.INC"; Установка схемы
CBLOCK 0x0C
W_TEMP ;0x0C
STATUS_TEMP ;0x0D
FLAGS ;0x0E
COUNTER ;0x0F
ENDC; Установка флагов и клокера
```

Описание использованных стегоалгоритмов. ***Алгоритм Коча (Koch).*** Согласно алгоритму [4–6] в блок размером 8×8 встраивается 1 бит цифрового водяного знака (ЦВЗ). В указанных работах описано две реализации алгоритма: псевдослучайно могут выбираться два или три коэффициента дискретного косинусного преобразования Фурье (ДКП).

Рассмотрим вариацию алгоритма с двумя выбираемыми коэффициентами.

Встраивание информации осуществляется следующим образом: для передачи бита 0 добиваются того, чтобы разность абсолютных значений коэффициентов была бы больше некоторой положительной

величины, а для передачи бита 1 эта разность делается меньше некоторой отрицательной величины:

$$\left|c_b(j_{i,j},k_{i,1})\right|-\left|c_b(j_{i,2},k_{i,2})\right|>\varepsilon, \quad \text{если} \quad s_i=0,$$

$$\left|c_b(j_{i,j},k_{i,1})\right|-\left|c_b(j_{i,2},k_{i,2})\right|<-\varepsilon, \quad \text{если} \quad s_i=1.$$

Алгоритм Бенхама (Benham). Этот алгоритм [4, 6] является улучшенной версией алгоритма Коча. Улучшения проведены по двум направлениям: для встраивания используются не все блоки, а лишь пригодные для этого. Внутри блока для встраивания выбираются не два, а три коэффициента, что уменьшает искажения. Пригодными для встраивания информации считаются блоки изображения, не являющиеся слишком гладкими, а также не содержащие малое число контуров. Для первого типа блоков характерно равенство нулю высокочастотных коэффициентов, для второго — очень большие значения нескольких низкочастотных коэффициентов. Эти особенности и являются критерием отсечения непригодных блоков.

При встраивании бита ЦВЗ псевдослучайно выбираются три коэффициента ДКП блока. Если необходимо вложить 1, коэффициенты изменяются так (если требуется), чтобы третий коэффициент стал меньше каждого из первых двух; если нужно встроить 0, он делается больше других. В том случае, если такая модификация приведет к слишком большой деградации изображения, коэффициенты не изменяют, и этот блок просто не используется. Изменение трех коэффициентов вместо двух, а тем более отказ от изменений в случае неприемлемых искажений уменьшает вносимые ЦВЗ погрешности. Декодер всегда сможет определить блоки, в которые ЦВЗ не встроен, повторив анализ, выполненный в кодере.

Особенностью описанных алгоритмов является то, что они позволяют встраивать в контейнер до стего объемом до 30 % объема контейнера.

Описание программной реализации алгоритма для проведения численного эксперимента. ***Описание хода исследований.*** Исследования стегоалгоритма проведены с помощью программы, реализованной в среде MathCAD:

• в среде MathCAD проводится загрузка монохромного изображения созданного QR-кода формата BMP с помощью оператора READBMP. В результате формируется матрица, элементами которой являются два значения — 0 и 255;

• осуществляется разбивка на блоки матрицы исходного изображения.

Существенный фрагмент программы деления матрицы приведен ниже:

```
DELC(p) := | c1 ← 0
           | c2 ← N − 1
           | for b ∈ 0..NC − 1
           |     r1 ← mod(N · b, X)
           |     r2 ← r1 + N − 1
           |     C_b ← submatrix(p, r1, r2, c1, c2)
           |     c1 ← c1 + N  if r2 = X − 1
           |     c2 ← c2 + N  if r2 = X − 1
           | C

C := DELC(p1)
```

В этом фрагменте реализован цикл деления размером 8×8. Процедура mod(.) возвращает остаток от деления x на y (x модулю y). Процедура submatrix(.) возвращает подматрицу массива A, состоящую из b элементов. Каждый блок массива C предназначен для сокрытия одного бита конфиденциального сообщения:

• производится вычисление ДКП Фурье применительно к каждому из полученных блоков. Существенные фрагменты программы ДКП с комментариями приведены ниже.

1. Вычисляются коэффициенты ДКП:

$$\xi(\chi) := \begin{cases} \dfrac{1}{\sqrt{2}} & \text{if} \quad \chi=0, \\ 1 & \text{if} \quad \chi>0. \end{cases}$$

2. К каждому из полученных блоков применяется прямое ДКП:

$$\Omega := DKP(C) :=$$

$$DKP(C) \left| \begin{array}{l} \text{for } b \in 0..NC-1 \\ \quad \text{for } \upsilon \in 0..N-1 \\ \quad\quad \text{for } \nu \in 0..N-1 \\ \quad\quad\quad \Omega r_{\upsilon,\nu} \leftarrow \zeta(\nu)\dfrac{\zeta(\upsilon)}{\sqrt{2N}} \displaystyle\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}\left[(C_b)_{x,y}\cos\left[\dfrac{\pi\upsilon(2x+1)}{2N}\right]\cos\left[\dfrac{\pi\nu(2y+1)}{2N}\right]\right] \\ \quad \Omega_b \leftarrow \Omega r \\ \Omega \end{array} \right.$$

$$\Omega := DKP(C)$$

• задаются параметры стегозаписи в контейнер — выбирается ключ и сообщение. В качестве ключа в выбранном алгоритме используются две позиции коэффициентов в матрице ДКП, которые будут использоваться при встраивании и извлечении сообщения в (из) контейнера.

При проведении эксперимента в качестве сообщения изначально было выбрано слово «1». Впоследствии этот параметр будет изменяться для оценки влияния постоянно растущего объема стего на восстановленное стего и расшифровку QR-кода. Для выбранного алгоритма также необходимо задать значение порога, с которым будут сравниваться результаты разности модулей коэффициентов ДКП, выбранных в качестве ключа. Сначала выберем его равным 25 ($P = 25$). Впоследствии этот параметр также будет изменяться для оценки влияния постоянно растущего объема стего на восстановленное стего и расшифровку QR-кода;

• проводится встраивание стего. Фрагмент программы встраиваниястего с комментариями приведен ниже:

$$\begin{array}{l} \text{D2B}(x) := \left| \begin{array}{l} \text{for } i \in 0..7 \\ \quad \left| \begin{array}{l} v_i \leftarrow \text{mod}(x, 2) \\ x \leftarrow \text{floor}\left(\frac{x}{2}\right) \end{array} \right. \\ v \end{array} \right. \end{array}$$

Процедура mod(.) возвращает остаток от деления x на y (x модулю y). Результат имеет тот же знак, что и x. Процедура floor(.) возвращает наибольшее целое, меньше или равное r. Производится восстановление блоков, путем вычисления обратного ДКП,

• восстанавливается исходная матрица;

$$
\begin{aligned}
\varepsilon := \;& \varepsilon \leftarrow \Omega \\
& M \leftarrow \mathrm{str2vec}(M) \\
& b \leftarrow 0 \\
& \text{for } \mu \in 0..\mathrm{rows}(M) - 1 \\
& \quad m \leftarrow \mathrm{D2B}\left(M_{\mu}\right) \\
& \quad \text{for } i \in 0..7 \\
& \quad\quad \Omega r \leftarrow \Omega_{i+N\mu} \\
& \quad\quad \omega_1 \leftarrow \left|\Omega r_{\upsilon_1,\nu_1}\right| \\
& \quad\quad \omega_2 \leftarrow \left|\Omega r_{\upsilon_2,\nu_2}\right| \\
& \quad\quad z_1 \leftarrow 1 \;\text{ if }\; \Omega r_{\upsilon_1,\nu_1} \geq 0 \\
& \quad\quad z_1 \leftarrow -1 \;\text{ if }\; \Omega r_{\upsilon_1,\nu_1} < 0 \\
& \quad\quad z_2 \leftarrow 1 \;\text{ if }\; \Omega r_{\upsilon_2,\nu_2} \geq 0 \\
& \quad\quad z_2 \leftarrow -1 \;\text{ if }\; \Omega r_{\upsilon_2,\nu_2} < 0 \\
& \quad\quad \omega_1 \leftarrow P + \omega_2 + 1 \;\text{ if }\; \left[\left(\omega_1 - \omega_2 \leq P\right) \wedge \left(m_i = 0\right)\right] \\
& \quad\quad \omega_2 \leftarrow P + \omega_1 + 1 \;\text{ if }\; \left[\left(\omega_1 - \omega_2 \geq -P\right) \wedge \left(m_i = 1\right)\right] \\
& \quad\quad \Omega r_{\upsilon_1,\nu_1} \leftarrow z_1\omega_1 \\
& \quad\quad \Omega r_{\upsilon_2,\nu_2} \leftarrow z_2\omega_2 \\
& \quad\quad \varepsilon_b \leftarrow \Omega r \\
& \quad\quad b \leftarrow b + 1 \\
& \varepsilon
\end{aligned}
$$

• полученый результат сохраняется в файл: WRITEBMP (qrcode KOCH1):=p2;

• далее для восстановления стего по известному ключу проводится загрузка сохраненного изображения из файла: p4:= WRITEBMP (qrcodeKOCH);

• по заданным функциям выполняется разбиение массива на блоки 8x8 и прямое ДКП: C2:=DELC(p4) и Ω2:=DKP(C2);

• затем восстанавливается стего по известному ключу. Фрагмент программы извлечения стего приведен ниже;

- определяется ключ:

$$B2D(x):=\sum_{i=0}^{7}\left(x_i 2^i\right);$$

- в тройном вложенном цикле проводится извлечение стего по известному ключу:

```
PPP := | j ← 0
       | for k ∈ 0.. DL − 1
       |     for i ∈ 0.. 7
       |         Ωr ← Ω2_j
       |         ω_1 ← |Ωr_{υ_1, ν_1}|
       |         ω_2 ← |Ωr_{υ_2, ν_2}|
       |         m_i ← 0  if ω_1 > ω_2
       |         m_i ← 1  if ω_1 < ω_2
       |         j ← j + 1
       |     PPP_k ← B2D(m)
       |     m ← 0
       | PPP
```

$$\text{vec2str(PPP)} = 1$$

Исследование влияния объема стего на восстановленное стего и расшифровка QR-кода показывает, что на восстановленное стего и расшифровку QR-кода могут влиять два параметра: объем встраиваемой информации и параметр порогового значения *P*.

Для оценки влияния объема стего на восстановленное стего и расшифровку QR-кода при фиксированном параметре *P* (*P* = 25) повторим описанную в п. 1 последовательность действий для объема встраиваемой информации, равного 100, 75, 50, 25, и 10 % объема контейнера. Для объема встраиваемой информации, равного 10 % объема контейнера, рассмотрим два различных скрываемых сообщения. Все полученные при этом QR-коды сохраняются в файлы qrcode KOCH6.bmp, qrcodeKOCH5.bmp, qrcodeKOCH4.bmp, qrcodeKOCH3. bmp, qrcodeKOCH7.bmp, qrcodeKOCH8.bmp соответственно.

Оценка влияния параметра P на восстановленное стего и расшифровку QR-кода. В процессе исследования опытным путем

было установлено, что при объеме скрываемого сообщения около 10 % объема контейнера (и даже нсколько большем), изменение значения параметра P линейно воздействует на значения пискселов в результирующем изображении. При этом в результирующем изображении значение пикселов будет 0 и $255 - 2*\left\lfloor \frac{P}{10} \right\rfloor$ при $P > 20$.

При ($P < 20$ ^ $P > 10$) значение пикселов будет 0 и 252, при ($P < 10$ ^ $P > 5$) — 0 и 253, при ($P < 5$ ^ $P > 1$) — будет 0 и 254.

При увеличении объема скрываемого сообщения изменение значений пикселов носит нелинейный характер. Кроме того, установлено, что при объеме скрываемого сообщения, меньшем либо равном 10 % объема контейнера, при любых значениях параметра P QR-код расшифровывается. При объеме скрываемого сообщения более 10 % объема контейнера и $P < 210$ QR-код также поддается расшифровыванию, а при объеме скрываемого сообщения более 10 % объема контейнера и $P > 210$ QR-код не расшифровывается.

Выводы по результатам численного эксперимента. При исследовании стегозаписи в контейнер в виде файла с изображением QR-кода методом Коча были получены следующие результаты.

При любом значении параметра P происходит полное восстановление стего и расшифровка QR-кода.

При объеме скрываемого сообщения, меньшем либо равном 10 % объема контейнера: при $P \geq 20$ в результирующем изображении значение пикселов будет 0 и $255 - 2*\left\lfloor \frac{P}{10} \right\rfloor$; при ($P < 20$ и $P > 10$) значение 0 и 252, при ($P < 10$ и $P > 5$) — 0 и 253, при ($P < 5$ и $P > 1$) — 0 и 254.

Численный эксперимент позволил также определить допустимое соотношение объемов контейнера и стего. Установлено следующее:

• при объеме скрываемого сообщения более 10 % объема контейнера и $P =< 210$ QR-код поддается расшифровыванию;

• при любом значении объема скрываемого сообщения происходят полное восстановление стего и расшифровка QR-кода. При этом, начиная с объема в 10 %, сокрытие сообщения в QR-коде становится визуально заметным. Вид скрываемого сообщения (было рассмотрено два разных текста) не влияет на полученный результат.

Проведенные исследования и полученные результаты разработки, часть которых приведена в работе, позволяют сделать следующие выводы.

1. Получен оригинальный метод сокрытия данных, в частности проектной документации, разрабатываемой в САПР.

2. Проведено теоретическое обоснование предложенных методов сокрытия данных.

3. Разработана экспериментальная программа, позволяющая оценить эффективность практической реализации предложенного метода.

4. Проведены экспериментальные исследования, результаты которых подтвердили корректность теоретических посылок.

ЛИТЕРАТУРА

[1] НОУ «РНТЦ ЭКИБ». URL: рнтц.рф (дата обращения 3.12.2013).

[2] Грибунин В.Г., Оков И.Н., Туринцев И.В. *Цифровая стеганография*. Москва, СОЛОН-Пресс, 2002, 272 с.

[3] Joseph J.K. Oð Ruanaidh*, Thierry Pun. Rotation scale and translation invariant spread spectrum digital image watermarking. *Signal Processing*, 1998, vol. 66, pp. 303−317.

[4] Pereira S., Joseph J., Deguillaume F. Template Based Recovery of Fourier-Based Watermarks Using Log-Polar and Log-Log Maps. *IEEE Int. Conf. on Multimedia Computing and Systems*, 1999, pp. 5–15.

[5] Lin Ch.-Y., Chang Sh.-F. Distortion Modeling and Invariant Extraction for Digital Image Print-and Scan Process. *International Symposium on Multimedia Information Processing*, 1999, pp. 10–23.

[6] Lin Ch.-Y., Chang Sh.-F. Public Watermarking Surviving General Scaling and Cropping: An Application for Print-and-Scan Process. *Multimedia and Security Workshop at ACM Multimedia*, 1999, pp.13–35.

[7] Pereira S., Thierry P. Fine Robust Template Matching for Affine Resistant Image Watermarks. *IEEE Trans. on Image Processing*, 1999, pp. 12–37.

[8] Kutter M. Watermarking Resisting to Translation, Rotation, and Scaling. *Signal Processing Laboratory*, 1998, pp. 10–27.

[9] Kutter M. Digital Signature of Color Images using Amplitude Modulation. *Signal Processing Laboratory*, 1997, pp. 9–23.

[10] Thilaka S., Donald L. Image Reconstruction with the FFT [Book Section]. GPU Gems 2. Addison-Wesley, 2005 (дата обращения 12.12.2013).

[11] Волосатова Т.М., Денисов А.В., Чичварин Н.В. Комбинированные методы защиты данных в САПР. *Информационные технологии. Приложение*, 2012, № 5, с. 2–32.

[12] Чичварин Н.В. Стеганографический метод маскирования данных с использованием цифровых голограмм. *Сб. докладов Всероссийской научн.-техн. конф. «Безопасные информационные технологии»*. Москва, МГТУ им. Н.Э. Баумана, 2011, с. 11–12.

[13] URL://http://www.ess.ru/publications/articles/kravchenko/kravchenko.htm (дата обращения 3.12.2013).

Статья поступила в редакцию 11.02.2014

Ссылку на эту статью просим оформлять следующим образом:

Волосатова Т.М., Чичварин Н.В. Исследование и разработка алгоритма защиты проектной документации в CAD/CAM/CAE от несанкционированного доступа. *Инженерный журнал: наука и инновации*, 2014, вып. 2.
URL: http://engjournal.ru/catalog/it/hidden/1201.html

Волосатова Тамара Михайловна родилась в 1955 г., окончила МВТУ им. Н.Э. Баумана в 1978 г. Канд. техн. наук, доцент кафедры «Системы автоматизированного проектирования» МГТУ им. Н.Э. Баумана. Автор более 100 научных и учебно-методических публикаций в области автоматизированного проектирования оптико-электронных систем и систем преобразования сигналов и защиты проектной документации САПР. e-mail: tamaravol@gmail.com

Чичварин Николай Викторович родился в 1947 г., окончил МВТУ им. Н.Э. Баумана в 1970 г. Канд. техн. наук, доцент кафедры «Информационная безопасность» МГТУ им. Н.Э. Баумана. Автор более 90 научных и учебно-методических публикаций в области автоматизированного проектирования оптико-электронных систем и систем информационной безопасности. e-mail: genrix.gertz@gmail.com