

2. Шутько Н. П. Передача текстовой информации на основе изменения апроша с использованием особенностей формата XML / Н. П. Шутько // Технические средства защиты информации: тез. докл. XXI Белорусско-российской науч.-техн. конф., Минск, 6 июня 2023 года. – Минск: БГУИР, 2023. – С. 102–103.

3. Стеганография. Общий обзор [Электронный ресурс]. – Режим доступа: [https://m.vk.com/@fractal\\_utmn-steganografiya-obschii-obzor?context=author\\_page\\_date&ref=author\\_page](https://m.vk.com/@fractal_utmn-steganografiya-obschii-obzor?context=author_page_date&ref=author_page). – Дата доступа: 01.11.2023.

4. Использование особенностей формата XML в методах текстовой стеганографии / П. П. Урбанович, О. А. Нистюк, М. Г. Савельева, Н. П. Шутько, А. Н. Николайчук // Информационные системы и технологии: материалы международного научного конгресса по информатике. – Ч. 1. – Минск: БГУ, 2022. – С. 120–126.

УДК 004.421.2

**А.Н. Щербакова, Д.М. Романенко**

Белорусский государственный технологический университет  
Минск, Беларусь

## **ОСОБЕННОСТИ ФОРМИРОВАНИЯ КЛЮЧА ДЛЯ КОДИРОВАНИЯ ВЕКТОРНЫХ ИЗОБРАЖЕНИЙ**

*Аннотация.* В статье рассмотрены особенности формирования ключа для кодирования векторных изображений с целью внедрения в электронные документы, содержащие векторные изображения, при этом построение графического ключа должно быть индивидуально для кодированного сообщения.

**A.N. Shcherbakova, D.M. Romanenko**

Belarusian State Technological University  
Minsk, Belarus

## **FEATURES OF KEY GENERATION FOR ENCODING VECTOR IMAGES**

*Abstract.* The article discusses the features of the formation of a key for encoding vector images for the purpose of embedding in electronic documents containing vector images, while the construction of a graphic key should be individual for the encoded message.

Кодирование – перевод данных, отображенных в виде первичного алфавита, в последовательность кодов [1].

Кодирование авторской информации в векторных изображениях может осуществляться в виде набора линий или простых геометрических фигур с разными параметрами (тип линии, толщина линии, расстояние между линиями). Защита любых документов строится на внедрении защитного ключа.

Методы генерации векторных защитных изображений описаны в [2]. При этом параметры генерации векторного изображения будут определяться секретным авторским ключом.

Рассмотрим общий вид ключа ( $K$ ), который будет состоять как минимум из следующих блоков, представленных в десятичном виде.

1. Блок  $K^1$ , определяющий количество и параметры линий (тип линии, толщина линии, расстояние между линиями), соответствующий каждому символу сообщений используемого алфавита (каждый отдельный параметр разделяется символом «;»). Для определения типа линии целесообразно ввести цифровые идентификаторы, например, 1 – сплошная линия (SL), 2 – штриховая (DL), 3 – длинный штрих (LDL), 4 – штрих-пунктир (DDL), 5 – длинный штрих-пунктир (LDDL). Отметим, что количество типов линий может быть увеличено, т. е. добавлены и другие типы. Толщина линий и расстояние между линиями будет задаваться в абсолютных величинах, а именно в «пт», т. е. пунктах.

Таким образом, например, если блок  $K^1$  будет иметь следующий вид А–2;1;1;0,5, то это означает, буква «А» в векторном изображении будет представлена двумя сплошными линиями толщиной 1 пт и расстоянием между ними 0,5 пт.

Важно учесть при формировании ключа, что какая-либо комбинация линий, соответствующая определенному символу используемого алфавита, не должна быть началом другой комбинации линий, соответствующих другому символу алфавита. В противном случае на стадии декодирования их будет не различить.

2. Блок  $K^2$ , определяющий количество повторов сообщения в генерируемом векторном защитном изображении. Данный блок должен учитывать параметр емкости векторного изображения, выражаемый в максимальном количестве сообщений, состоящих из самого «тяжелого» символа, который может быть закодирован в изображении. Он зависит также и от геометрического размера изображения.

3. Блок  $K^3$ , определяющий вспомогательные символы, например, символ начала сообщения, конца сообщения, повтора сообщения и т. д. Данные символы необходимы для того, чтобы определить, от какой линии и до какой необходимо выполнять декодирование.

4. В блок  $K^4$  можно определить некоторое число комбинаций линий и их параметров, использующихся для заполнения изображения до и после закодированного изображения. Назовем эти символы маскирующими, они должны будут выполнять задачу дорисовки защитного изображения и при этом решать проблему повышения сложности взлома ключа.

Далее перейдем к описанию алгоритма формирования графических кодов символов. Для этого надо определиться с двумя входными параметрами:

– количество символов в используемом алфавите ( $N$ ) – если использовать все символы таблицы ASCII, то  $N = 256$ ;

– количество используемых линий ( $k$ ). При этом под  $k_1, k_2, k_3, k_4$  и т.д. будем понимать определённый код линии. Так, если предполагается выполнять кодирование посредством черных линий толщиной 1 пиксель, то получим следующее (представлено в таблице 1).

Реализовать формирование графических кодов символов удобно через построение дерева, в котором у узла может быть не более  $k$  потомков. При этом на первом уровне будет  $k$  узлов, на втором уровне  $k_2$  узлов, на третьем уровне –  $k_3$  узлов и так до тех пор, пока на соответствующем уровне не будет достаточное количество узлов, т.е. пока не выполнится неравенство следующего вида:

$$k^i \geq N, \quad (1)$$

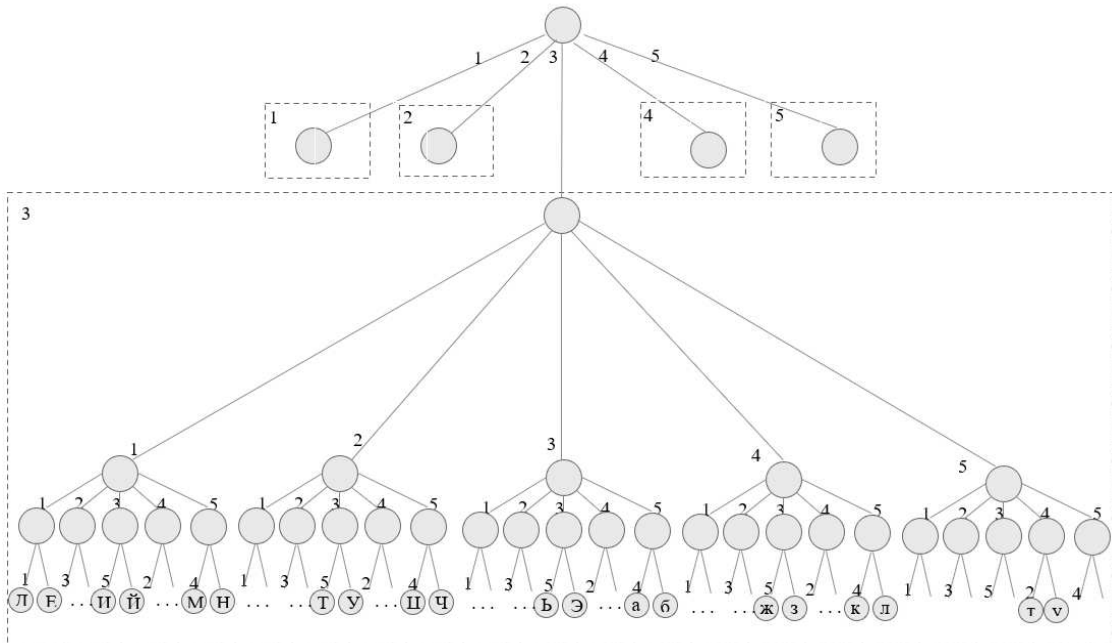
где  $i$  – количество потомков узла,  $i = 1, 2, 3$  и т. д. Узлы последнего уровня будут листьями дерева.

**Таблица 1 – Пример штрихового векторного кодирования**

| Формальное описание узла | Условное обозначение кода линии | Формальное описание кода |
|--------------------------|---------------------------------|--------------------------|
| $k_1$                    | 1                               | SL:1:(0,0,0)             |
| $k_2$                    | 2                               | DL:1:(0,0,0)             |
| $k_3$                    | 3                               | LDL:1:(0,0,0)            |
| $k_4$                    | 4                               | DDL:1:(0,0,0)            |
| $k_5$                    | 5                               | LDDL:1:(0,0,0)           |

Так, например, при необходимости кодирования символов таблицы ASCII 5-ю вариантами линий, представленными в таблице выше, получим, что дерево будет состоять из 4 уровней. Фрагмент одной ветви (отходящей от корня) дерева со всеми потомками на всех уровнях представлен на рис. 1.

Отметим, что на последнем, четвертом, уровне можно сделать по 3 потомка (суммарное возможное количество листьев (символов) в дереве будет равно  $5 \times 5 \times 5 \times 3 = 225$  – этого будет достаточно для кодирования таблицы ASCII с кодами от 32 («пробел») до 255 («я»)).



**Рис. 1- Пример построения графического кода (фрагмент ветви)**

Таким образом, каждый символ кодируемой последовательности символов будет присутствовать на генерируемом изображении в виде последовательности линий. Так, например, слово КОД должно быть представлено тремя квартетами линий:

К – 3142 (LDL:1:(0,0,0); SL:1:(0,0,0); DDL:1:(0,0,0); DL:1:(0,0,0));

О – 3211 (LDL:1:(0,0,0); DL:1:(0,0,0); SL:1:(0,0,0); SL:1:(0,0,0));

Д – 3111 (LDL:1:(0,0,0); SL:1:(0,0,0); SL:1:(0,0,0); SL:1:(0,0,0)).

Важнейшую роль во всем алгоритме кодирования играет построенное дерево – именно оно гарантирует то, что коды не повторяются, и то, что любой код не будет являться начальной комбинацией другого кода.

Из рисунка 1 примера видно, что все символы кодируются одинаковым количеством линий. Однако это не является обязательным условием предлагаемого алгоритма кодирования – можно построить дерево таким образом, чтобы листья, т.е. чтобы символы, располагались на всех уровнях дерева.

Приведем пример формирования графических кодов символов. Будем использовать символы таблицы ASCII от 97 до 122 и символ точка, то есть  $N = 27$ , каждая линия будет описываться тремя

параметрами. Первый параметр – тип линии. В качестве основных выберем следующие 3 типа (сплошная – Solid line (SL), штриховая – Dashed line (DL), штрих-пунктир – Dot-Dash line (DDL)). Вторым параметром является толщина линии, толщину будем задавать в пикселях, например, 1, 2 и 3. Третий параметр – расстояние между линиями, также 1, 2 и 3. Дерево представлено на рисунке 2.

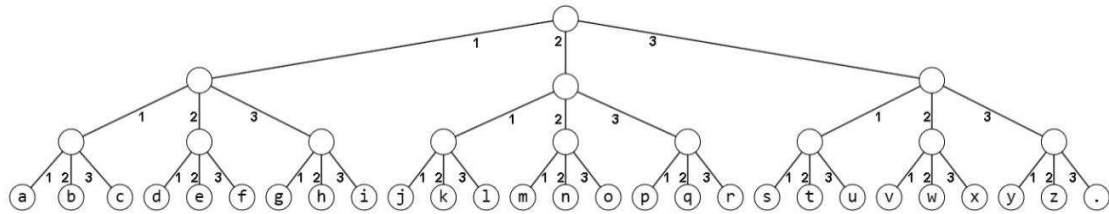


Рис. 2. Пример построения графического кода

Так, например, слово *vector* должно быть представлено шестью линиями:

v – 321 (DDL:2:1); e – 122 (SL:2:2); c – 113 (SL:1:3); t – 312 (DDL:1:2); o – 223 (DL:2:3); r – 233 (SL:3:3).

Важно отметить, что в каждой закодированной последовательности присутствует кодовое слово, которое различно для каждого пользователя. Например, в качестве кодового слова возьмем слово *code*. Тогда получим:

v – 321 (DDL:2:1); e – 122 (SL:2:2); c – 113 (SL:1:3); t – 312 (DDL:1:2); o – 223 (DL:2:3); r – 233 (SL:3:3); c – 113 (SL:1:3); o – 223 (DL:2:3); d – 121 (SL:2:1); e – 122 (SL:2:2)

Таким образом, блок  $K^1$  ключа влияет на построение дерева, блоки  $K^2$ ,  $K^3$ ,  $K^4$  необходимы для усложнения анализа изображения и потенциального взлома. Обеспечивать уникальность графических кодов при одинаковом количестве используемых для кодирования линий, а также одинаковом построенном дереве с позиции распределения кодов линий можно, варьируя для разных пользователей типы линий и их атрибуты, а также соответствие линий и символов. Сделать это удобно через хранение в ключевой информации пользовательского алфавита, т.е. набора используемых символов и знаков в определенной последовательности, не соответствующей таблице ASCII.

### Список использованных источников

1. Кузьмин, И.В. Теоретические основы информационные техники / И.В. Кузьмин // Сфера. – 2016. – 162 с.
2. Новосельская, О.А., Савчук, Н.А., Щербакова, А.Н., Романенко, Д.М. Алгоритмы и программное средство для генерации

защитных изображений печатных документов. / О.А. Новосельская // Труды БГТУ. Сер. 3, Физико-математические науки и информатика. – 2022. – № 1 (254). – С. 64-72.

3. Труфанова, Е.В., Кузнецова, Н.А., Макеев, П.В. Штрихкод. История создания, его разновидности и назначение / Е.В. Труфанова // Новая наука: теоретический и практический взгляд. – 2016. – № 8. – С. 186-188.

4. Novoselskaya, O.A., Kolesnikov, V.L., Solov'eva, T.V., Nagornova, I.V., Bablyuk, E.B., Trapeznikova, O.V. Scale Control and Quality Management of Printed Image Parameters / O.A. Novoselskaya // Journal of Physics: Conference Series. – Vol. 858. – 2017.

5. Романенко, Д.М., Алаа, Вахаб. Методы цифровой стеганографии на основе модификации цветовых параметров изображения / Д.М. Романенко // Труды БГТУ. – 2018. – № 1 (206). – С. 94-99.

УДК 336.741.24

**Н.И. Белодед, В.А. Абрамова, А.В. Федоренко**

Академия управления при Президенте Республики Беларусь  
Минск, Беларусь

## **ПРЕИМУЩЕСТВА И НЕДОСТАТКИ КРИПТОВАЛЮТЫ В УСЛОВИЯХ ГЛОБАЛИЗАЦИИ**

*Аннотация.* Едва ли, не главным признаком глобализационных тенденций современности выступает динамичное развитие всех сфер общества, в частности, различных инновационных технологий.

**N.I. Beloded, V.A. Abramova, A.V. Fedorenko**

Academy of Public Administration under the aegis of the President  
of the Republic of Belarus  
Minsk, Belarus

## **ADVANTAGES AND DISADVANTAGES OF CRYPTOCURRENCIES IN THE CONTEXT OF GLOBALIZATION**

*Abstract.* Perhaps the main sign of the globalization trends of our time is the dynamic development of all spheres of society, in particular, various innovative technologies.