© Kovtun V., Kinzeryavyy O. Experimental research of bitwise method for information hiding in vector images // Ukrainian Scientific Journal of Information Security, 2014, vol. 20, issue 1, p. 66-70.

ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ МЕТОДУ ПОБІТОВОГО ПРИХОВУВАННЯ ДАНИХ У ВЕКТОРНІ ЗОБРАЖЕННЯ

Владислав Ковтун, Олексій Кінзерявий

Національний авіаційний університет, Україна

КОВТУН Владислав Юрійович, к.т.н

Рік та місце народження: 1978 рік, м. Кіровоград, Україна.
Освіта: Харківський військовий університет, 2000 рік.
Посада: доцент кафедри безпеки інформаційних технологій з 2009 року.
Наукові інтереси: інформаційна безпека, швидкі арифметичні перетворення у полях Галуа, криптографічні системи з відкритим ключем, криптоаналіз криптографічних перетворень з відкритим ключем.
Публікації: більше 50 наукових публікацій, серед яких наукові статті у міжнародних та вітчизняних фахових журналах, патенти і т.д.
E-mail: vladislav.kovtun@gmail.com

КІНЗЕРЯВИЙ Олексій Миколайович

Рік та місце народження: 1989 рік, м. Кам'янець-Подільський, Хмельницька область, Україна.
Освіта: Кам'янець-Подільський національний університет ім. Івана Огієнка, 2011 рік.
Посада: аспірант кафедри безпеки інформаційних технологій з 2011 року.
Наукові інтереси: інформаційна безпека, криптографія, стеганографія.
Публікації: більш ніж 15 наукових публікацій, серед яких наукові статті у вітчизняних фахових журналах, тези доповідей та авторські свідоцтва на програмні продукти.
E-mail: oleksiykinzeryavyy@gmail.com

Анотація. *В роботі авторами пропонується експериментальне дослідження побітового методу приховування інформації у векторні зображення. Приховування даних за розглянутим методом відбувається при поступовому поділі кривих Без'є на візуально однакові ланцюжки кривих, які, за рахунок своїх властивостей, забезпечують стійкість до атак на основі афінних перетворень. Виділено основні етапи реалізації розглянутої стеганосистеми, за якими було проведено дослідження програмної реалізації даного методу з приховування інформації у криві Без'є векторного зображення формату SVG. В ході досліджень були встановленні і описанні необхідні умови з підбору стеганоконтейнера, ключові структурні елементи SVG зображення, основні характеристики та критичні параметри даної стеганосистеми. Отримані результати експерименту демонструють ефективність та пропускну здатність запропонованого методу.*

Ключові слова*: захист інформації, стеганографія, векторні зображення, метод побітового приховування даних, SVG зображення, критичні параметри стеганосистеми, криві Без'є, алгоритм де Кастельжо.*

Вступ

З розвитком та збільшенням обсягів інформації, що циркулює у інформаційних системах, постійно збільшується потреба в захисті інформації від зловмисників. Забезпечення безпеки інформаційних ресурсів може здійснюватися стеганографічними методами захисту інформації, які дозволяють приховати сам факт існування секретного повідомлення. Найпоширенішим типом контейнеру для приховування інформації є зображення, серед яких активно досліджуються і використовуються зображення растрового та фрактального типу [1]. Однак, існують ще векторні зображення над якими не проводилися дослідження по можливості приховування даних в них. У роботі [2] запропоновано метод побітового приховування інформації у криві Без'є векторного зображення, що використовуються для подання різних векторних фігур. Використання саме кривих Без'є обумовлене їх властивостями представляти фігури у вигляді ланцюжка кривих Без'є зі збільшеною кількістю

точок, що візуально не відрізнити. У кожному такому ланцюжку кривої може приховуватися секретна інформація.

У зв'язку з цим, **актуальною** науково-технічною задачею є формалізація вимог до стеганоконтейнера, дослідження характеристик методу побітового приховування інформації у векторні зображення (описаного в роботі [2]) та визначення критичних параметрів даної стеганосистеми.

Мета роботи полягає у формалізації вимог до стеганоконтейнера, дослідження характеристик методу побітового приховування інформації у криві Без'є векторного зображення та визначення критичних параметрів даної стеганосистеми.

Основна частина

Процес приховування та вилучення інформації з векторних зображень, згідно методу [2], можна подати у вигляді наступних етапів:

1. Проведення аналізу векторних фігур векторного зображення (стеганоконтейнера) на можливість приховування в ньому секретної інформації.
2. Приховування даних, шляхом представлення векторних фігур у вигляді кривих Без'є з подальшим їх розбиттям на сегменти з секретною інформацією.
3. Вилучення секретної інформації з обробленого векторного зображення шляхом аналізу сукупності ланцюжків кривих Без'є.

Перед проведенням аналізу векторних фігур конкретного векторного зображення, проведемо огляд відомих форматів зберігання векторних зображень та коротко зупинимося на одному з них.

Векторне подання зображень використовується при побудові повнокольорових ілюстрацій, складних креслень, логотипів, емблем тощо, де потрібне виконання афінних перетворень без втрати якості. Найпопулярнішими графічними пакетами, для створення та обробки векторних зображень, є: *Adobe Illustrator, CorelDRAW, Adobe Flash Professional, AutoCAD, ArchiCAD, KOMPAS-3D* та інші. Вони підтримують велику кількість векторних форматів представлення зображень: *AI, CDR, CMX, CDW, CDT, DWG, DXF, WMF, EMF, EPS, FLA, FH, SVG, SWF, CGM, IGS*. В роботі, автори, для наочного прикладу, використовують формат *SVG* (*Scalable Vector Graphics*), який є відкритим і не має патентних обмежень. *SVG* формат володіє наступними властивостями [3]:

− формат читається і модифікується за допомогою звичайного текстового редактору, крім того, він зазвичай має менший розмір ніж растрові формати (*JPEG*, *PNG* або *GIF*), а також добре піддається стисненню звичайними архіваторами;

− існує можливість масштабування будь-якої частини векторного зображення без втрати якості;

− існує можливість застосування афінних перетворень, деформацій та ефектів (розмиття, витискування, заливку, штрихування тощо);

− підтримує можливість використання елементів растрової графіки (*JPEG, PNG, GIF*);

− існує можливість використання кубічних та квадратичних кривих Без'є;

− існує можливість створення анімації за допомогою мови *SMIL*, скриптів *JavaScript*, каскадних таблиць стилів *CSS* та відстеження подій;

− *SVG* документ легко інтегрується з *HTML* і *XHTML* документами через тег <embed>.

Для побудови фігур замкненого чи відкритого типу в структурі *SVG* формату використовується об'єкт *Path.* Даний об'єкт за допомогою параметра *d* задає ряд символьних команд, кожна з яких задається типом команди та її числовими параметрами (див. табл. 1) [3, 4]. На основі даних команд поступово будуються відрізки прямих або дуги заданої фігури. При відносній формі запису кінцеві координати кожної попередньої команди стають точкою відліку для наступної команди, а при абсолютній формі – координати кожної команди обраховуються відносно початку координат.

Команди параметру *d* об'єкту *Path* Таблиця 1

Команда (Назва)	Опис команди
Z / z (closepath)	Завершення шляху та побудова ліній від поточної точки до першої позиції
M / m (moveto)	Встановлення опорної точки відліку
L / l (lineto)	Побудова ліній від поточної точки до заданої
H / h (horizontal lineto)	Побудова горизонтальної ліній від поточної точки до заданої, а координата Y буде дорівнювати ординаті поточної точки
V / v (vertical lineto)	Побудова вертикальної лінії від поточної точки до заданої, а координата X буде дорівнювати абсцисі поточної точки
C / c (curveto)	Побудова кубічної кривої Без'є від поточної точки до заданої (X,Y) з початковою (X_1,Y_1) і кінцевою (X_2,Y_2) контрольною точкою
S / s (smooth curveto)	Побудова кубічної кривої Без'є від поточної точки до заданої (X,Y), де (X_2,Y_2) остання контрольна точка, а перша контрольна точка передбачається як кінцева контрольна точка попередньої кривої
Q / q (quadratic Bezier curveto)	Побудова кубічної кривої Без'є від поточної точки до заданої (X,Y) з точкою управління (X_1,Y_1), що керує вигином кривої
T / t (smooth quadratic Bezier curveto)	Побудова кубічної кривої Без'є від поточної точки до заданої (X,Y), де контрольна точка цієї команди являє собою відображення контрольної точки попередньої команди

Закінчення таблиці 1

A / a (elliptical arc)	Побудова еліптичної кривої від опорної точки до заданої (X,Y), де розмір і орієнтація еліпса задається двома радіусами (R_x,R_y) та параметром *x-axis-rotation*, який визначає розташування еліпса по осі X. Центр еліпса обчислюється автоматично на підставі заданих параметрів, а параметри *large-arc-flag* і *sweep-flag* задають зовнішній вигляд еліпса

Отже, необхідною **вимогою** для приховування даних в векторний формат *SVG* є наявність в структурі даного файлу об'єкта *Path* та параметра *d* з командами за якими будуються криві Без'є. При позитивному результаті структурної перевірки *SVG* файлу слід переходити до наступного етапу. У противному випадку потрібно перетворити та подати все векторне зображення у вигляді сукупностей ланцюжків кривих Без'є.

В даній роботі не розглядається сам метод, а проводиться його експериментальне дослідження по приховуванню даних в структурі *SVG* зображення.

Процес приховування даних у криві Без'є векторного зображення за методом [2] полягає в поступовому поділі даних кривих на сегменти. Розбиття кривої Без'є будь-якого порядку на ланцюжок кривих того ж порядку буде здійснюватися за алгоритм де Кастельжо (рис. 1, 2) [5], причому отриманий ланцюжок кривих буде візуально подібний до початкової кривої. Така подібність може бути використана для зворотного обчислення початкових значень точок перетину між кривими Без'є.

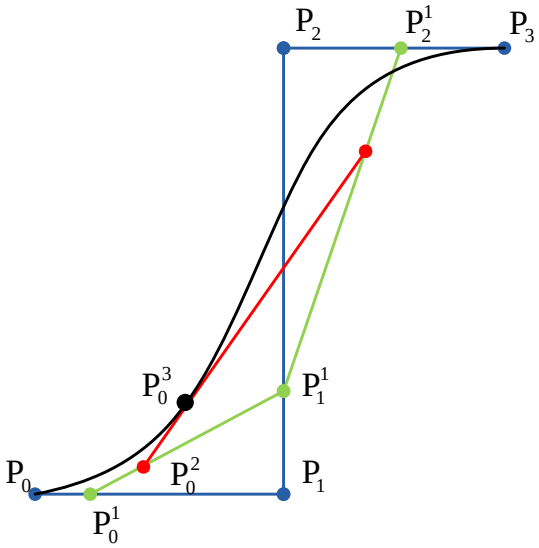

Рис. 1. Застосування алгоритму де Кастельжо

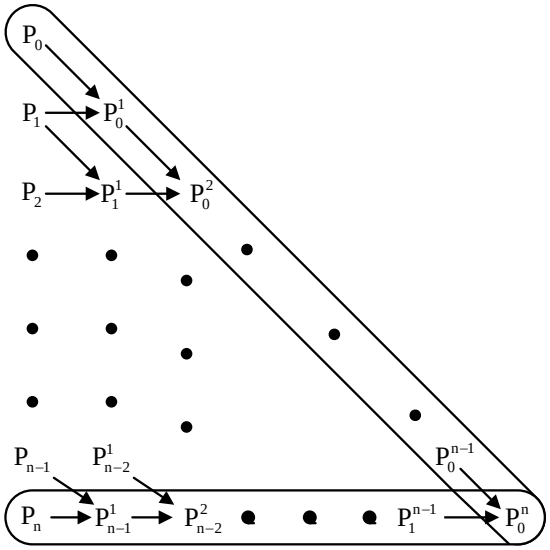

Рис. 2. Схематична ілюстрація алгоритму де Кастельжо

Для проведення експерименту по приховуванню та вилученню даних з однієї кривої Без'є (типу ***curveto***) кожного *SVG* зображення були використані наступні критичні параметри:

– **Критичний параметр 1.** Розбиття кривої Без'є при надто малих значеннях параметру побудови кривої $t = t + \Delta t$, $t \in [0,1]$, де Δt – крок зміни параметра t, призводило до отримування координати точок утворених частин кривої з дуже великою дробовою частиною (більше 20 десяткових знаків). Тому, був введений параметр CP_1, що обмежував максимально допустиму кількість десяткових знаків дробової частини координат опорних точок отримуваного ланцюжка кривої Без'є.

– **Критичний параметр 2.** Розбиття надто малої кривої Без'є, де відстань між її опорними точками досить мала, призводило до отримання координат точок утворених частин кривої з дуже великою дробовою частиною, що при зворотному процесі вилучення даних потребувало більше десяткових знаків дробової частини ніж дозволено параметром CP_1. Тому, було введено обмеження стосовно мінімальної довжини кривої відносно відстаней між її опорними точками CP_2.

Приховування даних в криву Без'є (згідно методу [2]) при певному кроці t виконувалося так:

– якщо необхідно приховати біт з значенням «0», то при даному t крива Без'є не ділилася на частини, а здійснювався перехід до наступного біта приховуваної послідовності, що приховувався при наступному кроці $t = t + \Delta t$;

– якщо необхідно приховати біт з значенням «1», то при даному t виконувався поділ кривої Без'є на два сегмента. Причому, подальше внесення наступного біта приховуваної послідовності здійснювалося в отриманий другий сегмент кривої Без'є при наступному кроці $t = t + \Delta t$.

Розбиття кривої Без'є (типу ***curveto***) на два сегмента при певному кроці t проходило в три етапи, де на кожному етапі обраховувалися додаткові точки [5]:

– ***<u>На першому етапі</u>*** з початкових координат точок кривої Без'є P_0, P_1, P_2, P_3 обчислювалися

точки $P_0^1 = (1-t)P_0 + tP_1$, $P_1^1 = (1-t)P_1 + tP_2$ та $P_2^1 = (1-t)P_2 + tP_3$.

– ***На другому етапі*** з отриманих координат точок P_0^1, P_1^1, P_2^1 розраховувалися точки $P_0^2 = (1-t)P_0^1 + tP_1^1$ та $P_1^2 = (1-t)P_1^1 + tP_2^1$.

– ***На третьому етапі*** з отриманих координат точок P_0^2, P_1^2 отримувалися координати останньої точки $P_0^3 = (1-t)P_0^2 + tP_1^2$.

За даними точками початкова крива Без'є ділилася на два сегмента, з яких перший сегмент будувався по точкам P_0, P_0^1, P_0^2, P_0^3, а другий – по точкам P_0^3, P_1^2, P_2^1, P_3 [5].

Вилучення приховуваних даних з обробленого *SVG* зображення виконувалось шляхом збирання початкової кривої Без'є з ланцюжка кривих, що здійснювалося в процесі поступового об'єднання двох кінцевих кривих даної послідовності, наступним чином:

1) На кожному кроці t, що змінювався в зворотному проміжку $[1,0]$ при визначеному кроці Δt, обчислювалися координати точок P_1^1 за двома способами:

– ***Перший спосіб.*** Спочатку обраховувалися додаткові координати $P_1 = \left(P_0^1 - (1-t)P_0\right)/t$ та $P_2 = \left(P_2^1 - tP_3\right)/(1-t)$, за допомогою яких визначався $P_{1(1)}^1 = (1-t)P_1 + tP_2$.

– ***Другий спосіб.*** За координатами P_1^2 та P_2^1 обраховувався $P_{1(2)}^1 = \left(P_1^2 - tP_2^1\right)/(1-t)$.

2) При виконанні рівності $P_{1(1)}^1 \approx P_{1(2)}^1$ виконувалося об'єднання двох останніх кривих послідовності в одну криву Без'є, координатами якої ставали точки P_0, P_1, P_2, P_3. Об'єднання даних кривих вказувало на вилучення біта приховуваної послідовності з значенням «1», а отримана крива, надалі, поєднувалася з наступною кривою послідовності при наступному кроці $t = t - \Delta t$.

3) Не виконання рівності $P_{1(1)}^1 \approx P_{1(2)}^1$ означало отримання біта з значення «0», а також те, що дані криві порівнювалися знову при наступному кроці $t = t - \Delta t$.

Критичний параметр 3. При обчисленні координат точок $P_{1(1)}^1$ та $P_{1(2)}^1$ виникала деяка цифрова неточність декількох останніх десяткових знаків дробової частини даних координат, що пов'язана з накладанням обмеження CP_1. Дана неточність суттєво впливала на результат порівняння даних точок і на сам процес вилучення даних, тому було введено максимально допустиму похибку в кількості останніх десяткових знаків дробової частини координат опорних точок CP_3.

На основі описаного вище був проведений експеримент по приховуванню та вилученню даних з 30-ти довільних *SVG* зображень, з наступними критичними параметрами: $CP_1 = 15$, $CP_2 = 4$, $CP_3 = 5$. Результати експерименту наведені в табл. 2.

Результати експерименту по приховуванню даних в *SVG* зображеннях Таблиця 2

Номер *SVG* файлу	Розмір прихованої інформації, байт	Розмір стеганоконтейнера «до», байт	Розмір стеганоконтейнера «після», байт	Збільшення розміру стеганоконтейнера, %
1	20	22243	33323	49,81
2	40	9198	28646	211,44
3	60	20910	43585	108,44
4	80	14231	51478	261,73
5	100	18949	66146	249,07
6	120	6509	61577	846,03
7	140	15661	79778	409,41
8	160	9037	82448	812,34
9	180	13902	97029	597,95
10	200	69428	160895	131,74
11	220	11655	112287	863,42
12	240	66158	175427	165,16
13	260	56360	173173	207,26
14	280	68047	193388	184,2
15	300	10019	147083	1368,04
16	320	23478	169818	623,31
17	340	64757	215667	233,04
18	360	8576	171772	1902,94
19	380	17276	190647	1003,54
20	400	8643	190844	2108,08
21	420	59466	251577	323,06
22	440	34407	235252	583,73
23	460	28935	237302	720,12
24	480	20903	239648	1046,48
25	500	63636	290914	357,15
26	520	33593	270823	706,19
27	540	86794	332014	282,53
28	560	38227	294161	669,51
29	580	9658	274500	2742,2
30	600	31106	304988	880,48

Висновки

З результатів експерименту встановлено, що в одну криву Без'є можна приховати дані досить значного розміру. Максимальний та мінімальний розмір приховуваної інформації при різних критичних параметрах стеганосистеми потребує подальшого ретельного дослідження. Крім того, кількість приховуваних даних можна збільшити в декілька разів якщо провести приховування їх в усі криві Без'є *SVG* зображення. Однак, недоліком за приховування великих об'ємів даних буде зростання розміру стеганоконтейнера в велику кількість разів.

З огляду на це, актуальним та необхідним є проведення дослідження стосовно пошуків шляхів оптимізації запропонованого методу побітового приховування інформації у криві Без'є векторного зображення.

Література

[1] Кінзерявий О.М. Систематизація сучасних методів комп'ютерної стеганографії / О.М. Кінзерявий, В. Ю. Ковтун, С. О. Гнатюк // Безпека інформації. — 2013. — №3. — С. 209-217.

[2] Кінзерявий О.М. Стеганографічний метод приховування даних у векторних зображеннях / О.М. Кінзерявий, В.Ю. Ковтун, С.О. Гнатюк, В.М. Кінзерявий // Вісник Інженерної академії України. — 2013. — №3-4. — С. 66-68.

[3] Dailey D. Building Web Applications with SVG / D. Dailey, J. Frost, D. Strazzullo. — O'Reilly Media. — 2012. — 268 с.

[4] Дунаев В.В. HTML, скрипты и стили : [3-е издание] / В. В. Дунаев. — СПб. : БХВ-Петербург, 2011. — 816 с.

[5] Кунву Ли. Основы САПР (CAO/CAM/CAE) / Ли Кунву. — СПб. : Питер, 2004. — 560 с.

УДК 004.056.5 (045)

Ковтун В. Ю., Кинзерявый А. Н. Экспериментальное исследование метода побитового сокрытия информации в векторные изображения

Аннотация. *В работе авторами предлагается экспериментальное исследование метода побитового сокрытия информации в векторные изображения. Сокрытие данных, согласно рассмотренному методу, происходит при постепенном разделении кривых Безье на визуально одинаковые цепочки кривых, которые, за счет своих свойств, обеспечивают устойчивость к атакам на основе аффинных преобразований. Выделено основные этапы реализации рассматриваемой стеганосистемы, согласно которых было проведено исследование программной реализации данного метода сокрытия информации в кривые Безье векторного изображения формата SVG. В ходе исследований были установлены необходимые условия по подбору стеганоконтейнера, ключевые структурные элементы SVG изображения, основные характеристики и критические параметры данной стеганосистемы. Полученные результаты эксперимента демонстрируют эффективность и пропускную способность предложенного метода.*

Ключевые слова: *защита информации, стеганография, векторные изображения, метод побитового сокрытия данных, SVG изображения, критические параметры стеганосистемы, кривые Безье, алгоритм де Кастельжо.*

Kovtun V., Kinzeryavyy O. Experimental research of bitwise method for information hiding in vector images

Abstract. *In this work the authors propose an experimental research of bitwise method for information hiding in vector images. Hiding data, according to the considered method, occurs with the gradual separation of Bezier curves to visually identical chains of curves, which, due to their properties, provide resistance to attacks, based on affine transformations. The main stages of realization of examined steganosystem are highlighted, under which, there was conducted the software implementation of this method of hiding information in Bezier curves the vector image of SVG format. The necessary conditions for the selection of steganocontainer, the key structural elements of SVG images, the basic characteristics and the critical parameters of the steganosystem are established and described during the research. The obtained results of experiment demonstrate the efficiency and throughput of the proposed method.*

Key words: *information security, steganography, vector images, bitwise method of data hiding, SVG images, critical parameters of the steganosystem, Bezier curves, algorithm de Casteljau.*

Отримано *18 лютого 2014 року*, затверджено редколегією *13 березня 2014 року*