пользовать механизм синтаксических деревьев с целью обеспечения более высокоуровневой работы с исходным кодом при его анализе на соответствие условиям алгоритма выявления определённой уязвимости. В материале рассмотрен статический анализ исходного кода программ для мобильных платформ как одно из средств обеспечения безопасности мобильных систем.

ЛИТЕРАТУРА

1. Drake, J. J. Android Hacker's Handbook. – Indianapolis: Wiley, 2020. – 576 p.

2. Bergman, N. Hacking Exposed: Mobile Security Secrets & Solutions. – NY: Mc Graw Hill, 2013. – 289 p.

3. Chess, B. Secure Programming with Static Analysis. – Boston: Addison-Wesley Professional, 2007. – 624 p.

УДК 003.26

Маг. А.Н. Николайчук; ст. преп. Е.А. Блинова
(БГТУ, г. Минск)

КОМБИНИРОВАННОЕ ПРИМЕНЕНИЕ ДВУХ СТЕГАНОГРАФИЧЕСКИХ МЕТОДОВ ДЛЯ РАЗМЕЩЕНИЯ ЦИФРОВОГО ВОДЯНОГО ЗНАКА В ФАЙЛАХ ФОРМАТА SVG

При хранении и передаче цифровых электронных документов важную роль играет обеспечение их безопасности. Одним из способов защиты информации является применение технологий цифрового водяного знака на основе стеганографического метода, суть которого заключается в скрытом размещении сообщения [1].

Ввиду уникальности свойств форматов цифровых документов необходимо разрабатывать специфические методы защиты для каждого из них. Однако для некоторых видов контейнеров могут рассматриваться методы, представляющие собой комбинацию различных подходов. Например, электронный текстовый документ формата SVG можно рассматривать как текст, как векторное изображение, как набор полей, содержащих метаинформацию и как контейнер, имеющий определенную структуру, что позволяет комбинировать классические подходы как текстовой, так и графической стеганографии [2−4].

Формат векторной графики SVG (Scalable Vector Graphics) позволяет легко манипулировать объектами, из которых состоит, благодаря своей структуре, описание файла осуществляется с помощью языка разметки XML. С его помощью можно создавать как примитив-

ные геометрические объекты (окружности, линии), так и составные объекты. Для создания сложных графических объектов обычно используется общий элемент <*path*>, который определяется атрибутом *d*, который содержит последовательность из команд и значений координат ключевых точек, используемых этими командами [5].

Существует три команды, которые используются для создания плавных кривых линий, две из этих кривых – кривые Безье: квадратичная, Q, и кубическая, C.

Параметрические уравнения кривой Безье можно представить в следующем виде:

$$Q = (1-t)^2P_1+2(1-t)tP_2+t^2P_3. \quad (1)$$

$$C = (1-t)^3P_1+3(1-t)^2tP_2+3(1-t)t^2P_3+t^3P_4,\ t \in [0,1]. \quad (2)$$

Для однозначного определения кривой второго порядка (1) необходимы три точки: P_1, P_2 и P_3. Начальная точка P_1, конечная точка P_3 – опорные точки кривой, а точка P_2 – контрольная.

Для однозначного определения кривой третьего порядка (2) потребуются четыре точки: P_1, P_2, P_3 и P_4. Начальная точка P_1, конечная точка P_4 – опорные, а точки P_2 и P_3 – контрольные.

Элемент <path> отображающий кривые второго и третьего порядка может быть представлен следующим образом:

<*path d = "M* x_1, y_1 *Q* x_2, y_2 ... *T* x_i, y_i ... *C* x_j, y_j ... *S* x_k, y_k ..." >.

В данном докладе рассматриваются два различных метода внедрения информации в файл формата SVG. Файлы этого формата редко состоят из простых фигур, поэтому в качестве контейнера для внедрения сообщения чаще всего рассматриваются фигуры, описанные с помощью кривых Безье.

Первый из методов взаимодействует с кривой Безье как с графическим объектом. Согласно методу разбиения кривых Безье, предложенному де Кастельжо, эту кривую можно разделить на две части в некотором отношении. Основная сущность этого стеганографического метода состоит в том, что скрытая информация размещается в точке разделения кривой на сегменты, а именно в отношении разделения скрывается только часть секретного сообщения, а для остальной части используются появляющиеся контрольные и опорные точки. При получении новых контрольных и опорных точек в них изменяются младшие знаки таким образом, чтобы, с одной стороны, это изменение было визуально незаметно, но в них осаждалось секретное сообщение, а с другой стороны, чтобы при извлечении контролировалось, правильно ли было извлечено отношение разделения [6].

Отметим, что содержимое SVG-файла описывается на беско-

нечном холсте и может быть любого размера, но при описании фигур учитываются области отображения. Основываясь на особенностях свойств атрибутов определения размеров изображения, был предложен второй метод. Размеры изображения, отображаемого на экране можно задавать с помощью двух разных областей: системной (viewport) и пользовательской. Область viewport задается с помощью атрибутов корневого тега <svg>: height – высота и width – ширина. Пользовательская область просмотра устанавливается с помощью атрибута viewBox, значение которого принимает четыре параметра. С помощью последних задаются размеры: min-x – начало оси координат X; min-y – начало оси координат Y; width – ширина; height – высота.

SVG предоставляет возможность управлять поведением содержимого относительно области viewport, что позволит обрезать изображение. Таким образом, можно описывать фигуры, чтобы кривая создавалась с координатами, превышающими значения ширины и высоты изображения.

В качестве модифицируемых параметров контейнера используются координаты, описывающие ключевые точки кривых Безье, так как графический объект такого типа может иметь наибольшее число значений, которыми можно описать фигуру.

Вне зависимости от типа контейнера с помощью стеганографических методов решаются два основных класса задач: скрытая передача данных и защита авторских прав. Первое подразумевает незаметную передачу информации по открытым каналам, а также скрытое хранение информации. Второе реализуется с помощью цифровых водяных знаков. Цифровые отпечатки и водяные знаки могут использоваться для защиты авторского права на каждую копию контента, а также для подтверждения достоверности и целостности переданной информации. И если для задач первого класса можно использовать оба метода по отдельности, для второго класса целесообразнее было бы совместить два метода.

Комбинированное применение двух стеганографических методов для размещения цифрового водяного знака можно реализовать следующим алгоритмом.

1. Получить сформированное сообщение.

2. С помощью метода, основанного на использовании особенностей отображения элементов разместить сообщение в файле SVG.

3. Вычислить хеш сообщения с помощью алгоритма SHA-256.

4. С помощью метода, с разделением в кривых Безье на сегменты разместить хеш в файле SVG.

Таким образом, для того чтобы подтвердить целостность данных, сообщение извлекается из файла и вычисляется его хеш. Извлеченный хеш и вычисленный сравниваются и, если данные совпали, значит передаваемое сообщение не подвергалось изменению.

С помощью метода, основанного на использовании особенностей отображения элементов можно разместить сообщение любой длины. Однако для метода, основанного на встраивании скрытых сообщений в кривые Безье существует ограничение на минимальное количество кривых Безье. В виду условий этого ограничения для внедряемого хеша вычисленного по алгоритму SHA-256, длина которого будет составлять 256 бит, необходимо наличие 66 кривых Безье в исходном файле-контейнере.

ЛИТЕРАТУРА

1. Урбанович, П.П. Защита информации методами криптографии, стеганографии и обфускации: учеб.-метод. пособие / П.П. Урбанович. – Минск: БГТУ, 2016. – 220 с.

2. Almutairi, A. A Comparative Study on Steganography Digital Images: A Case Study of Scalable Vector Graphics (SVG) and Portable Network Graphics (PNG) Images Formats / A. Almutairi// (IJACSA) International Journal of Advanced Computer Science and Applications. – 2018. – V. 9. – № 1. – P. 170–175.

3. Сейеди, С. А. Сравнение методов стеганографии в изображениях/ С. А. Сейеди, Р. Х. Садыхов // Информатика. – 2013. – № 1(37). – С. 66–75.

4. Text Steganography utilizing XML, HTML And XHTML Markup Languages / S. Imran [et al.] // International Journal of Computational Geometry & Applications. – 2017. – № 3. – P. 99–116.

5. Блинова, Е. А. Стеганографический метод на основе встраивания дополнительных значений координат в изображения формата SVG/ Е. А. Блинова, П. П. Урбанович // Труды БГТУ. Серия 3. – 2018. – № 1. – С. 104–109.

6. Blinova, E. A. Steganographic method based on hidden messages embedding into Bezier curves of SVG images / E. A. Blinova, P. P. Urbanovich // Journal of the Belarusian State University. Mathematics and Informatics. – 2021. – № 3. – P. 68–83.