Least Significant Bit steganography in SVG XML architecture

UDC 004.056.5:004.92

Nataliya Zagorodna[1], Oleh Yarema[2]

Ternopil Ivan Puluj National Technical University,
[1]zagorodna_n@tntu.edu.ua, [2]yarema.oleh.m@gmail.com

Traditional graphical steganography relies on raster images to embed covert data by modifying pixel color values. However, as web systems shift toward scalable, responsive media, raster-based methods face security scrutiny and structural limitations. Steganography is, first of all, the science of hiding a secret message within a non-secret file to avoid triggering suspicion. The Least Significant Bit (LSB) method is the foundational algorithm of this domain. It operates on the principle that digital media files possess data bits that contribute minimally to human perception [1].

In a standard uncompressed image, each pixel is typically represented by 24 bits of data – 8 bits each for the red, green and blue color channels (RGB). The leftmost bits (most significant bits) describe the primary color information, while the rightmost bits (least significant bits) represent very minimal variations in shade [1]. LSB steganography exploits this by replacing these LSBs with the binary stream of an encrypted secret file.

Mainstream implementations of LSB are localized to raster graphics, audio files, and video containers. These mainstream applications suffer from bottlenecks that limit their utility in modern security bases. Because raster pixels follow natural statistical distributions, sequential LSB injection alters the global pixel histograms. Modern steganalysis tools can easily detect these changes [2]. Raster LSB is highly fragile. Everyday web processes, such as converting a PNG to a lossy JPEG, scaling an image down, or applying compression, can destroy the LSB array, corrupting the hidden data. Modern firewalls and automated Deep Packet Inspection (DPI) systems target traditional media attachments on the first priority basis and subject them to algorithmic checks for embedded hidden data.

To bypass some of the limitations of raster media, steganography can use Scalable Vector Graphics (SVG). SVGs are text files written in structured XML code. They do not contain a grid of pixels, instead they contain mathematical instructions how to render an image by the web browser [3].

We propose a framework where LSB principles are mapped directly onto the XML DOM structure. An SVG defines colors textually via hexadecimal strings or standard CSS RGB strings [3]. By targeting the lowest bits of these color palettes, inline fills, and stroke attributes, a secret payload can be distributed across the structural elements of a webpage.

Future scientific investigations should focus on the following unexplored dimensions. Research is needed to develop parsers that dynamically map the XML tree of an SVG, isolate color attributes, and handle the LSB flipping within string data types rather than raw binary matrices. SVGs also rely on precise floating-point coordinate points. A massive avenue for research lies in coordinate LSB manipulation, where the thousandths decimal place of a geometric shape is altered to hold data. Because vector points are highly precise, shifting an object by 0.0001 millimeters is visually non-existent but offers massive data capacity.

Future studies must benchmark SVG LSB against standard defensive tools. Because SVGs are processed as code by firewalls rather than images, research can empirically prove whether vector steganography can bypass mainstream pixel-based steganalysis engines.

LSB steganography must evolve alongside modern web standards. By embedding data into the XML architecture of SVG files, we can open up a lightweight and novel vector for secure data transmission.

1. Aditya, S., Ved, M., Shashikant, K., Samadhan, K., & Shilpa, M. A. (2024). Image steganography using least significant bit. International Journal of Research Publication and Reviews, 5(4), 2505–2508. https://doi.org/10.55248/gengpi.5.0424.0966
2. Madoš, B., Hurtuk, J., Čopjak, M., Hamaš, P., & Ennert, M. (2014). Steganographic algorithm for information hiding using scalable vector graphics images. Acta Electrotechnica et Informatica, 14(4), 42–45. https://doi.org/10.15546/aeei-2014-0040
3. Xu Z. Xu D. Li Z. Zheng X. & Zhang C. (2026). GVIS: Generative vector image steganography. In proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 9384–9393). https://openaccess.thecvf.com/content/CVPR2026/papers/Xu_GVIS_Generative_Vector_Image_Steganography_CVPR_2026_paper.pdf

Метод виявлення ботнет-активності в корпоративній мережі на основі багатокритеріальної оптимізації XGBoost

УДК 004.056:004.85

Владислав Самойленко[1], Сергій Гахов[2]

Державний університет інформаційно-комунікаційних технологій,
[1]v.samoylenko@duikt.edu.ua, [2]gakhovsa@gmail.com

Ботнет-активність залишається однією з найнебезпечніших загроз для корпоративних мереж, оскільки використовується для DDoS-атак, розсилання спаму, крадіжки даних та прихованого віддаленого керування зараженими вузлами. У дослідженні авторів [1] на наборі даних CSE-CIC-IDS2018 було порівняно Random Forest, XGBoost та SVM для виявлення ботнетів. Найкращим базовим рішенням стала XGBoost, яка досягла середнього значення F1-міри 0,99 за крос-валідацією при часу навчання близько 16 с. Це обґрунтувало перехід від простого вибору моделі до її оптимізації з урахуванням умов практичного мережевого моніторингу.

Метою роботи є розроблення методу виявлення ботнет-активності, який одночасно забезпечує високу якість класифікації, контроль рівня хибних спрацювань та достатню пропускну здатність для потокового моніторингу. Для уникнення витоку інформації дані Friday-02-03-2018 було поділено на раннє вікно S1 для навчання й оптимізації та пізніше вікно S2 для фінального тестування, а benign-підмножину Thursday-15-02-2018 використано як незалежний негативний день для калібрування порога рішення за обмеженням FPR ≤ 1 %. Основою експерименту слугував набір даних CSE-CIC-IDS2018 [2].