УДК 004.056.55

А. С. Камінська
В. В. Карпінець

ДОСЛІДЖЕННЯ СТРУКТУРНОЇ ОРГАНІЗАЦІЇ ФОРМАТІВ STL ТА OBJ ДЛЯ СТЕГАНОГРАФІЧНОГО ЗАХИСТУ 3D-МОДЕЛЕЙ

Вінницький національний технічний університет

Анотація

У роботі здійснено аналіз структурної організації базових форматів 3D-графіки STL та OBJ у контексті їх використання як стегоконтейнерів для приховування даних і впровадження цифрових водяних знаків (ЦВЗ). Обґрунтовано необхідність використання гібридних моделей захисту 3D-моделей, що поєднують симетричне/асиметричне шифрування та стеганографію. Проаналізовано специфічні приховані канали: для STL (80-байтовий заголовок, 2-байтове поле атрибутів, LSB координат) та для OBJ (топологічні перестановки вершин і граней за допомогою автомата Edgebreaker, трансформні методи DWT). Оцінено стеганографічну ємність, стійкість до топологічних атак та вплив на візуальну якість (PSNR, SSIM). Запропоновано концепцію багаторівневої кібербезпеки цифрових 3D-активів.

Ключові слова: 3D-стеганографія, цифрові водяні знаки, формат STL, формат OBJ, алгоритм Edgebreaker, найменш значущий біт (LSB), дискретне вейвлет-перетворення (DWT), кібербезпека..

Abstract

The paper presents an analysis of the structural organization of the fundamental 3D graphics formats STL and OBJ in the context of their use as steganographic containers for data hiding and digital watermark embedding. The necessity of applying hybrid protection models for 3D assets, combining symmetric/asymmetric encryption with steganographic techniques, is substantiated. Specific covert channels are analyzed: for STL (80-byte header, 2-byte attribute field, and least significant bits of vertex coordinates) and for OBJ (topological permutations of vertices and faces using the Edgebreaker scheme, as well as transform-domain methods such as discrete wavelet transform, DWT). The steganographic capacity, robustness against topological attacks, and impact on visual quality (PSNR, SSIM) are evaluated. A concept of a multi-layered cybersecurity framework for digital 3D assets is proposed.

Keywords: 3D steganography, digital watermarks, STL format, OBJ format, Edgebreaker algorithm, LSB, DWT, cybersecurity.

Вступ

Швидкий розвиток програм для 3D-моделювання та технологій 3D-друку призвів до того, що тривимірні цифрові файли стали використовуватися повсюди – від створення медичних протезів до машинобудування та кіноіндустрії. Оскільки розробка таких високоточних моделей вимагає багато часу, інтелектуальних зусиль та фінансів, питання надійного захисту авторських прав на них стоїть надзвичайно гостро. Звичайне комп'ютерне шифрування чудово захищає файл під час його передачі через інтернет. Проте його головний недолік полягає в тому, що як тільки легальний покупець або користувач розшифровує файл і відкриває його у своїй програмі, модель стає повністю беззахисною. Її можна миттєво скопіювати, змінити або нелегально викласти в мережу.

Вирішити цю проблему допомагає 3D-стеганографія та використання цифрових водяних знаків. Ці технології дозволяють непомітно «вплести» секретну інформацію (наприклад, ім'я автора, номер ліцензії чи дані про покупця) прямо у внутрішній код самої 3D-моделі. На відміну від звичайних плоских фотографій, де пікселі розташовані рівними рядами, 3D-моделі мають набагато складнішу просторову будову. До того ж, 3D-моделі постійно крутять, збільшують, зменшують або трохи згладжують перед друком, що може легко знищити простий прихований код. Метою цієї роботи є глибоке порівняння внутрішньої будови двох найпопулярніших форматів (STL та OBJ), щоб зрозуміти, як саме в них можна сховати величезні масиви інформації так, щоб цього ніхто не помітив і не зміг видалити.

Результати дослідження

Будь-яка полігональна 3D-модель математично описується як сукупність вершин (Vertices), ребер (Edges) та граней (Faces), що утворюють просторову сітку (Mesh). Проте підходи до формалізації та зберігання цих даних суттєво відрізняються залежно від обраного формату файлу. Саме ця структурна різниця визначає, де і як можна знайти вільне місце для таємних даних [1].

Формат STL є галузевим стандартом для 3D-друку [2]. Його головна особливість полягає в тому, що він описує лише геометрію поверхні тривимірного об'єкта без будь-якої інформації про колір, текстури чи інші атрибути. Структурно STL-файл є невпорядкованим списком трикутних граней. Кожна грань описується вектором нормалі до площини трикутника та координатами трьох його вершин у тривимірному декартовому просторі: V=(x,y,z). У форматі STL відсутня топологічна інформація – спільні вершини сусідніх трикутників записуються дублюванням координат. Формат має дві репрезентації: текстову (ASCII) та бінарну. У найпопулярнішому бінарному STL після 80-байтового заголовка кожний трикутник займає рівно 50 байт (по 12 байт на нормаль та три вершини у форматі 32-bit floating-point, плюс 2 байти на атрибути) [2].

Така жорстка структура дозволяє безпечно ховати дані кількома способами. По-перше, у 80-байтовий заголовок можна вписати до 640 біт інформації (наприклад, цифровий підпис) [3]. Це взагалі ніяк не змінить форму об'єкта, проте є дуже ненадійним способом, бо при перезбереженні заголовок часто стирається. По-друге, нестандартні 2 байти атрибутів (які історично призначалися для збереження кольору) дозволяють записати туди таємні дані: форма моделі залишиться ідеальною, а місця вистачить для великого тексту [2]. Третій спосіб передбачає впровадження цифрових водяних знаків у найменш значущі біти (LSB) мантиси чисел з плаваючою комою, що відповідають за координати вершин, з мінімальним впливом на геометрію. Якщо акуратно змінити останні 8 бітів, точка зміститься непомітно для людського ока. Однак через відсутність топологічної інформації та постійне дублювання вершин, алгоритм має спочатку знайти всі копії однієї точки і змінити їх абсолютно однаково, щоб не розірвати поверхню.

Формат OBJ, розроблений компанією Wavefront Technologies, є значно гнучкішим і складнішим [4]. Він підтримує не лише трикутники, а й багатокутники (N-gons), а також зберігає інформацію про матеріали та текстури [5]. Дані в OBJ записуються у текстовому вигляді (ASCII). Основна структурна відмінність від STL полягає у наявності індексації [4]. Файл містить окремі списки координат вершин (ідентифікатор v), текстурних координат (vt) та нормалей (vn). Опис самих полігонів (ідентифікатор f) здійснюється шляхом посилання на індекси раніше оголошених вершин, наприклад: f 1/1/1 2/2/1 3/3/1 [4].

Завдяки розділенню геометрії (координат) та топології (індексів граней), формат OBJ надає більше можливостей для стеганографічного захисту. Цифрові водяні знаки можна вбудовувати не лише шляхом просторової модифікації координат, але й за рахунок зміни порядку запису вершин або нормалей у файлі. Якщо ми перемішаємо рядки з точками у коді, на екрані 3D-модель залишиться абсолютно ідентичною, але сама послідовність працюватиме як шифр (ємність такого методу наближається до O(n\log{2}n) біт) [6]. Щоб одержувач міг правильно прочитати таке повідомлення, використовується спеціальний алгоритм обходу поверхні Edgebreaker. Цей алгоритм крок за кроком рухається по гранях, детерміновано генеруючи однаковий маршрут для кодера та декодера. Цей спосіб є стійкішим до певних типів атак на геометрію моделі (наприклад, обертання чи масштабування), хоча і ламається при штучному спрощенні сітки моделі [7].

Щоб зробити захист ще надійнішим і стійким до спроб хакерів його видалити, науковці використовують більш складні математичні перетворення [7]. Наприклад, можна працювати не з самими координатами, а з "частотами" моделі (використовуючи хвильові перетворення DWT) або проектувати точки на уявні площини. Такі методи дозволяють заховати цілі зображення всередині 3D-моделі. Дослідження алгоритмів на базі проекцій показали, що під час тестування на надзвичайно складній моделі дракона метод забезпечив найвищий показник якості та непомітності (PSNR понад 47,87 дБ) і зберіг дані навіть після того, як модель грубо спростили на 30% або додали до неї сильний цифровий "шум" [7]. Також існують багатошарові алгоритми, які дозволяють ховати дані у 7–13 шарів одночасно, перетворюючи звичайну 3D-модель на справжній архів даних [7].

Але для повноцінної кібербезпеки одного лише приховування даних недостатньо. Якщо просто

заховати звичайний текст у координати моделі, спеціальні програми-шпигуни легко знайдуть ці зміни, бо вони виглядають як неприродні аномалії [8]. Тому сучасний підхід вимагає подвійного захисту. Спочатку повідомлення обов'язково шифрується надійними алгоритмами (наприклад, AES-128 або Blowfish) [9]. Використання стандарту AES-128 дозволяє перетворити текст на абсолютно випадковий набір символів, зберігаючи при цьому найвищу якість моделі. Після цього зашифрований текст ховається не підряд, а «стрибаючи» по точках моделі дуже складним, заплутаним маршрутом (для цього використовують математичні фокуси, такі як перестановка Йосифа або коди Грея) [10].

Крім того, на допомогу інженерам з кібербезпеки сьогодні приходить штучний інтелект та машинне навчання [11]. Наприклад, алгоритм Isolation Forest здатен самостійно аналізувати форму моделі та знаходити найбільш складні, сильно вигнуті та рельєфні ділянки (глибокі складки, гострі кути чи шорсткості). Якщо ховати цифрові водяні знаки саме в таких місцях, а не на рівних, гладких площинах, мікроскопічні зміни зливаються з природним рельєфом об'єкта [12]. Завдяки цьому людське око та прості програми перевірки ніколи їх не помітять.

Висновки

Детальний аналіз форматів STL та OBJ доводить, що вони чудово підходять для створення надійних систем захисту авторських прав на тривимірні об'єкти. Формат STL дозволяє ховати величезні обсяги інформації у своїх текстових полях або за рахунок мікроскопічного зсуву координат. Однак він вимагає дуже обережного ставлення, щоб випадково не пошкодити поверхню моделі мікротріщинами. Формат OBJ, завдяки своїй структурі зі списками, є ідеальним майданчиком для того, щоб ховати інформацію, просто змінюючи порядок запису рядків – це взагалі не змінює форму об'єкта і є абсолютно непомітним.

Справжній і найвищий рівень безпеки досягається лише тоді, коли технологія приховування працює в нерозривній парі з класичним потужним шифруванням та алгоритмами штучного інтелекту. В умовах сучасного високотехнологічного виробництва зловмисники намагаються не просто вкрасти файли, але й непомітно змінити машинний код для 3D-принтера, щоб зробити надруковану деталь крихкою чи бракованою. Тому зараз активно розвивається напрямок кіберфізичної перевірки: коли захований у комп'ютерному файлі водяний знак переноситься на реальну фізичну деталь під час друку, і згодом авторство чи цілісність можна довести, просто відсканувавши цей об'єкт. Розвиток таких технологій допомагає надійно захистити інтелектуальну власність та відповідає ключовим цілям сучасної стратегії кібербезпеки.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. 3D Compression Made Simple: Edgebreaker on a Corner-Table URL: https://www.cs.cmu.edu/~alla/edgebrea ker_simple.pdf (дата звернення: 15.03.2026)
2. STL (file format) URL: https://en.wikipedia.org/wiki/STL_(file_format) (дата звернення: 15.03.2026)
3. A Fragile Watermarking Scheme for Authenticity Verification of 3D Models in GLB Format URL: https://www.mdpi.com/2076-3417/15/13/7246 (дата звернення: 15.03.2026)
4. Wavefront .obj file URL: https://en.wikipedia.org/wiki/Wavefront_.obj_file (дата звернення: 15.03.2026)
5. The Difference Between STL and OBJ URL: https://chiggofactory.com/stl-vs-obj/ (дата звернення: 15.03.2026)
6. Multi-carrier information hiding based on projection-driven vertex embedding in 3D models URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC12469212/ (дата звернення: 15.03.2026)
7. A High Capacity 3D Steganography Algorithm URL: https://www.researchgate.net/publication/320722403_A_HighCapacity_3D_Steganography_Algorithm_With_Adjustable_Distortion (дата звернення: 15.03.2026)
8. Three-Dimensional Mesh Steganography and Steganalysis: A Review URL: http://staff.ustc.edu.cn/~zhangwm/Paper/2021_25.pdf (дата звернення: 15.03.2026)
9. A robust high capacity Gray code-based double layer security scheme for secure data embedding in 3D objects URL: https://www.itu.int/pub/S-JNL-VOL3.ISSUE2-2022-A26/fr (дата звернення: 15.03.2026)
10. Image-to-Image Steganography with Josephus Permutation and Least Significant Bit (LSB) 3-3-2 Embedding URL: https://www.mdpi.com/2076-3417/14/16/7119 (дата звернення: 15.03.2026)

11. On the Usability of Isolation Forest for 3D Mesh Analysis and Watermarking URL: https://www.mdpi.com/2076-3417/15/21/11364 (дата звернення: 15.03.2026)
12. Yaremchuk Yu., Saliieva O., Karpinets V., Nikolaienko A., Kunanets N. Enhancing the steganographic resistance of hidden information to active attacks. Proceedings of the International Workshop on Cybersecurity Providing in Information and Telecommunication Systems II (CPITS-II 2024), Kyiv, October 26, 2024. 2024. Vol. 3826. Pp. 350-355. URL: https://ceur-ws.org/Vol-3826/short27.pdf (дата звернення: 15.03.2026)

Камінська Анна Сергіївна **–** студентка групи 1КІТС-22б, Факультет менеджменту та інформаційної безпеки, Вінницький національний технічний університет, м. Вінниця, e-mail: akaanyaa@gmail.com
Карпінець Василь Васильович **–** завідувач кафедри МБІС, кандидат технічних наук, доцент, Вінницький національний технічний університет, м. Вінниця, e-mail: karpinets@vntu.edu.ua

Kaminska Anna S. – student of group 1KITS-22b, Faculty of Management and Information Security, Vinnytsia National Technical University, Vinnytsia, e-mail: akaanyaa@gmail.com
Karpinets Vasyl V. – Head of the Department of Management and Security of Information Systems, Candidate of Technical Sciences, Associate Professor, Vinnytsia National Technical University, Vinnytsia, e-mail: karpinets@vntu.edu.ua