КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 3 (31), 2025
ISSN 2663 - 4023
DOI 10.28925/2663-4023.2025.31.1010
УДК 004.62

Момрик Ярослава Богданівна
асистент кафедри захисту інформації
Національний університет "Львівська політехніка", Львів, Україна
ORCID: 0009-0001-4114-0347
Yaroslava.b.momryk@lpnu.ua

ПІДХОДИ ДО БЕЗПЕЧНОГО ЗБЕРЕЖЕННЯ ТА ОПРАЦЮВАННЯ ДАНИХ НА ОСНОВІ UUID ІДЕНТИФІКАТОРІВ – МЕТОД ГРАФІЧНОГО ПРЕДСТАВЛЕННЯ

Анотація. У статті проаналізовано сучасні тенденції використання у базах даних унікальних ідентифікаторів типу UUID (Universal Unique Identifier) для створення ключових записів. Висвітлено напрями застосування таких ідентифікаторів у програмних системах різних сфер діяльності та визначено актуальні виклики, пов'язані з їх використанням у контексті сучасних вимог до безпеки даних. Розглянуто переваги ULID (Universally Unique Lexicographically Sortable Identifier) — ідентифікаторів нового покоління, що відрізняються вищою швидкістю роботи та можливістю лексикографічного сортування завдяки монотонному принципу генерації. На основі аналізу сучасних потреб у захисті даних зроблено висновок, що ідентифікатори на основі UUID потребують високого рівня безпеки, зокрема від SQL (Structured Query Language), XSS (Cross-Site Scripting) , CSRF (Cross-Site Request Forgery) та Insecure Direct Object Reference (IDOR)-атак. Підкреслено, що методи захисту мають враховувати специфіку типів даних, на яких побудовано ключові ідентифікатори. Як варіант розширення інструментів безпеки запропоновано метод конвертації UUID у графічну форму, що візуально нагадує унікальний графік. Побудовано алгоритм і схему для визначення координат UUID та ULID, наведено приклад перетворення і візуального представлення. Розглянуто форми та сфери застосування запропонованого методу, а також можливість зворотної конвертації для перевірки цілісності, маскування у клієнт - серверних застосунках, звірки та відновлення даних після атак або збоїв. Описано використання як самого зображення графіка, так і масиву координат, що його формує. Запропоновано підходи до кодування та шифрування координатних даних на етапі формування та промальовування графіка. Метод дає змогу приховати такий ключ у графічній формі, що ускладнює його ідентифікацію під час несанкціонованого доступу (наприклад, у разі викрадення cookie), а також може виконувати функції графічного хешу. Запропоновано комбінований спосіб використання методу у вебзастосунках: зашифровані координати, сформовані за цим методом, зберігаються у прихованому полі клієнтської форми, а на сервері перевіряються за даними, збереженим у сесії. Такий спосіб забезпечує перевірку достовірності ідентифікатора (захист від IDOR) і водночас виконує функцію анти-CSRF токена. У сфері збереження даних запропоновано підходи до застосування методу для перевірки цілісності та відновлення даних після атак та інцидентів без необхідності негайного підняття резервних копій. Запропонований метод конвертації може бути реалізований у вигляді графічного хешу або захищеної копії. Він дає змогу працювати як із графічним зображенням, так і з масивом координат у безпечнішому, порівняно із символьним, форматі. Метод має низку переваг перед QR-кодом: на відміну від останнього, структура графічного хешу не є стандартизованою, тому може бути розшифрована лише сервером, що володіє параметрами генерації та алгоритмом декодування, забезпечує можливість візуального представлення результатів. Метод конвертації унікальних ідентифікаторів типу UUID сприятиме вирішенню низки прикладних задач і доповнить засоби забезпечення безпеки та конфіденційності даних у всіх сферах застосування таких ідентифікаторів.

Ключові слова: кодування ідентифікаторів; графічне представлення хешу; безпека даних; цілісність даних; захист ключових ідентифікаторів; UUID; ULID, безпека вебзастосунків, унікальні ідентифікатори графічно; захист від CSRF, захист від IDOR.

ВСТУП

Ідентифікатори є важливим компонентом у багатьох сферах застосування, зокрема при роботі з базами даних, оскільки забезпечують унікальність записів і дозволяють будувати ключі для ідентифікації користувачів та об'єктів. Одним із найпоширеніших форматів таких ідентифікаторів є UUID (Universal Unique Identifier) — 128-бітне число для унікальної ідентифікації об'єктів у комп'ютерних системах, яке зазвичай подається у вигляді текстового рядка у шістнадцятковому форматі.

У невеликих базах даних доцільніше використовувати числові ключі, оскільки вони потребують менше пам'яті та спрощують виконання операцій порівняння і пошуку. UUID застосовують для зв'язування даних між різними базами або в системах з розподіленим зберіганням даних, де інформація однієї бази розташована на кількох серверах. Аналогічним типом є LUID (Locally Unique Identifier), що забезпечує впорядкованість за часом генерації та оптимізацію ресурсів.

Для захисту ідентифікаторів у вебсередовищі застосовуються сесійні механізми, HMAC-підписи (Hash-based Message Authentication Code) та CSRF-токени. Однак ці методи мають обмеження: вони потребують зберігання стану сесії або секретних ключів і не завжди захищають від спроб ідентифікації структури ID під час перехоплення трафіка.

Постановка проблеми. Для побудови ключів баз даних зазвичай використовують числові ідентифікатори — це оптимальний підхід як з точки зору витрат ресурсів, так і з точки зору реалізації захисту даних. Проте існує низка задач, де використання UUID подібних ідентифікаторів є необхідним — зокрема для інтеграції розподілених систем або у випадках, коли кількість записів перевищує можливості звичайного числового діапазону.

Тип UUID застосовується у стандартному механізмі унікальної ідентифікації об'єктів у вебдодатках. Його широке використання обумовлене можливістю незалежної генерації без ризику колізій. Водночас UUID часто стає об'єктом атак типу IDOR, коли зловмисник підмінює ідентифікатор у запиті з метою отримання несанкціонованого доступу до даних. Тому забезпечення цілісності, достовірності та невідтворюваності UUID під час передачі між клієнтом і сервером є актуальним завданням. Особливо небезпечними є ситуації, коли UUID передаються у URL-адресах, параметрах запитів чи формах, або використовуються як ідентифікатори сесій, що зберігаються у cookie-файлах. Такі сценарії створюють потенційні вразливості для SQL-, XSS-, CSRF- та IDOR-атак.

У світлі сучасних кіберзагроз особливої актуальності набувають питання безпечного зберігання, шифрування та відновлення даних після атак. Важливо не лише забезпечити конфіденційність обробки інформації, але й розробити методи перевірки цілісності даних у випадках, коли доступ до резервних копій ускладнений або тимчасово неможливий (наприклад, після кібератаки чи фізичного пошкодження інфраструктури).

Отже, окрім перевірки прав доступу на сервері, необхідними стають методики ідентифікаторів та забезпечення їх безпечної обробки і зберігання [1-3].

У прикладних сферах, крім технічних аспектів, виникає потреба враховувати сучасні соціальні та технологічні виклики — зокрема автоматизацію процесів, доступність сервісів для осіб з інвалідністю та необхідність використання візуальних, а не лише цифрових підписів користувачів. Це зумовлює необхідність пошуку інноваційних методів безпечного використання ідентифікаторів типу UUID.

Мета статті. Метою статті є дослідження сучасних тенденції та потреби у розробці та використанні баз даних і запропонувати методологічне рішення для підвищення рівня їх безпеки та ефективності

Для досягнення поставленої мети необхідно вирішити наступні завдання:

1. Проаналізувати особливості проектування баз даних та програмного забезпечення, побудованого на основі UUID-ідентифікаторів, і виявити основні вразливості та ризики безпеці та конфіденційності даних.

2. Розробити метод конвертації UUID у графічну форму, що забезпечує додатковий рівень безпеки, а також побудувати його алгоритм, модель та продемонструвати приклад виконання.

3. Запропонувати підходи до застосування розробленого методу для захисту та контролю цілісності даних у різних предметних областях.

4. Запропонувати застосування розробленої конвертації для покращення способів захисту від підробки ідентифікаторів у вебдодатках, що підвищить стійкість до потенційних атак

Аналіз останніх досліджень та публікацій. У літературі значна увага приділяється унікальним ідентифікаторам та їх застосуванню. Зокрема, автор [4] проаналізував види таких ідентифікаторів (GUID, UUID, ULID), їх переваги та недоліки. Ще глибший аналіз проведено дослідниками [5], які оцінили дев’ять технологій і систем для призначення постійних ідентифікаторів у науках про Землю. Вони дійшли висновку, що схема UUID найбільше відповідає вимогам до унікального ідентифікатора. У роботі [6] експериментально підтверджено, що тип UUID у довгостроковій перспективі при операціях запису працює майже вдвічі швидше за GUID, що узгоджується з рекомендаціями про вибір числових полів для побудови ключів баз даних, поданими у праці [7]. У роботі [7] також проаналіровано загрози ідентифікаторам і запропоновано методи захисту числових ідентифікаторів.. Разом з тим є ряд задач, для яких такого типу ключів недостатньо. Ряд досліджень [8;9] обґрунтовують доцільність використання UUID для підвищення продуктивності у великих розподілених базах даних:“якщо у вас є кілька незалежних баз даних, які ви потім синхронізуєте, використання UUID означає, що один ідентифікатор є унікальним для всіх баз даних, а не тільки для тієї, де ви перебуваєте і де він був згенерований. При об'єднанні в єдиний кластер конфліктів не виникає” [8]. Автори [10] не лише виділяють аспекти доцільності використання таких ідентифікаторів, але і запропонували метод MPE, який дозволяє підвищити ефективність UUID-систем за допомогою багатокритеріальної оптимізації. Згідно проаналізованої літератури такі унікальні ідентифікатори використовуються в ряді прикладних задач:

- **Синхронізація даних.** У роботі Chen & Li [8] реалізовано систему синхронізації баз даних SQLite із використанням UUID як основного ключа;
- **Логістика та стандартизація.** У звіті Thurman та ін. [11] подано рекомендації щодо використання UUID у життєвому циклі продукту й стандартах цифрових потоків;
- **Ідентифікація пристроїв.** UUID широко використовується для ідентифікації пристроїв — від USB-накопичувачів до Android Device ID [12];
- **Електронне голосування.** Jadhav, Chouhan & Maskar [13] запропонували систему голосування, де UID перевіряється разом із біометричними даними виборця;
- **Документообіг.** У [14] також зазначено, що UUID доцільно використовувати для ідентифікації файлів, користувачів і категорій документів у ієрархічних сховищах;
- **Маркування проб.** Triebel та ін. [15] описують етап графічного представлення UUID у вигляді QR-кодів для ідентифікації контейнерів для екологічних зразків.

Деякі розробники вважають, що UUID можна в URL-рядку відкрито показувати. Такий підхід можна прийняти для таких задач, як приховування контактних даних від надавачів реклами, коли ідентифікатор створюють і передають, щоб не висвічувати, наприклад, електронну адресу [12], але для ряду задач, де йдеться ,приміром, про авторизацію чи видачу бюлетенів на вибори, чи банківські транзакції, такі ключові дані потребують першочергового захисту від розкриття. Серед векторів атак є особливо небезпечні для такого типу задач , що використовують ідентифікатори типу UUID — зокрема, у [13] йдеться про використання SQL-атак для розкриття ідентифікаторів конфіденційних даних та XSS-інфікування. Дослідження Pratama і Rhusuli [16] показало, що IDOR залишається однією з найпоширеніших проблем у сучасних вебсистемах, оскільки навіть при наявності автентифікації зловмисник може отримати доступ до конфіденційних даних шляхом прямої підстановки ID-значень. Автори [17] пояснюють небезпеку вразливості від атак CSRF (міжсайтова підробка запиту), які особливо небезпечні, коли хакери вираховують працівника з великими привілеями доступу. Зазвичай для захисту UUID застосовують формат JSON, серіалізацію або графічні QR-коди. Однак розглянуті вище вразливості обґрунтовують необхідність інтеграції додаткових механізмів контролю доступу, кодування та валідації ідентифікаторів на рівні серверної логіки, а також вироблення підходів, спрямованих на підвищення рівня захисту й надійності обробки даних у сучасних інформаційних системах. Пошук методу спрямований також на вирішення ряду прикладних задач і доповнення засобів безпеки та конфіденційності даних у всіх сферах застосування унікальних ідентифікаторів.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Актуальність графічного перетворення та побудова алгоритму методу.

Розглянемо *прикладні області* та приклади задач*, для* яких у сучасних умовах доцільне застосування запропонованого методу конвертації UUID/ULID у графічне представлення (графік):

Приклад: візуальні підписи для клієнтів банків.

Маркування проб. Triebel та ін. [15] описують етап графічного представлення UUID у вигляді QR-кодів для ідентифікації контейнерів для екологічних зразків. Розглянуто задачу зберігання візуального зразка підпису клієнта як зображення в базі даних (як це практикують деякі банки при відкритті рахунку). Такий підхід іноді ускладнює осіб з інвалідністю, якщо стандартне ПЗ не передбачає альтернативних способів підтвердження. Запропонований метод дозволяє автоматично генерувати хеш-графіки на основі UUID як технічні замінники візуального зразка — при збереженні гарантується унікальність і можливість верифікації. Клієнт може підтвердити дію, наприклад, через SMS-код, а сам графік слугуватиме візуальним маркером, сумісним із наявними інтерфейсами, без суттєвих доопрацювань програмного забезпечення. Це особливо корисно для автоматизації сервісів, що обслуговують користувачів з фізичними вадами. Таким чином, вибір методу конвертації у графік, співставного із зразком підпису, дає змогу вирішити вищезгадане питання.

Захист у вебдодатках (клієнт — сервер). Використання UUID як ключових полів у при виконанні операцій з записами баз даних ставить питання безпечної передачі й збереження даних. Пошкодження або підміна ідентифікатора в процесі запису транзакції ускладнює відновлення даних навіть за наявності бекапу. У роботі окремо розглянуто загрози SQL-ін’єкцій, CSRF (міжсайтова підробка запиту) та інші вектори, пов’язані з використанням ідентифікаторів у cookie та параметрах запитів. Для таких сценаріїв

важливо не лише запобігати ін'єкціям та спотворенню, а й запровадити механізми перевірки автентичності клієнта та достовірності форм. Виходимо з того для таких ідентифікаторів важливо розглянути не лише захист від інфекцій та спотворення але і також засоби перевірки достовірності клієнта- захист від підробки форм при застосуванні у вебдодатках.

Потреба в удосконаленні методів збереження та перевірки цілісності.

Сучасні вектори атак часто спрямовані не лише на те, щоб видалити всі дані, а й на пошкодження їх ідентифікації – знищення цілісності. Крім того, внутрішні збої можуть призводити до псування інформації. Тому актуальним є пошук додаткових підходів, що дозволяють швидко перевіряти цілісність ідентифікаторів без негайного підняття резервних копій (які можуть зберігатися в інших локаціях або бути тимчасово недоступними).

Доцільність вибору методу обумовлена також тим, що бінарне зображення важко піддається SQL-інфекції, а чисельні масиви легше захистити від таких атак.

Генерація графічного зображення на зразок унікального графіка.

Виходимо з того, що на етапі проектування бази даних побудовано ключовий ідентифікатор uuid. Для зменшення ризиків можна запропонувати кодування за наступним покроковим алгоритмом конвертації. Припускаємо, що початкове значення сформоване стандартними функціями генерації. Для прикладу візьмемо значення: UUID = d9e7a184-5d5b-4586-a62a-3499710062d0

Алгоритм перетворення UUID.

Етап 1. Розбити на логічні частини Розбити на частини поділені " –".

Етап *2*. Кожну з частин перетворити у масив десяткових чисел.

Складові числа, які дає наш приклад, подані у таблиці 1.

Таблиця 1

Приклад перетворення UUID з 16-кової у десяткову систему числення

Номер	Початкове значення у 16 системі числення (hex)	Значення після перетворення у 10 системі числення ((dec))
1	D9E7A184	3655836036
2	5d5b	23899
3	4586	4586
4	a62a	42538
5	3499710062d0	57833630491344

По суті вже на даному етапі маємо масив чисел A={3655836036;23899;4586;42538;57833630491344}, де задача захисту при передачі може бути зведена до захисту числових ідентифікаторів - можна застосувати методи запропоновані у [4]. Але оскільки йде мова про масив чисел, а не про одне число, обчислення будуть довшими, тому розглянемо можливість інших підходів.

Етап 3. Побудувати координати графіка.

Для цього для кожного елемента масиву A[s] створюється набір координат (xi,yi)(x_i, y_i)(xi,yi), який визначає положення відповідної точки або лінії на графіку. Кожне число розміщується у своїй чверті системи координат, що дозволяє візуально розділити складові UUID. Оскільки складових 5 - кожне з них відобразити у певній чверті системи координат, число посередині (3) на осі 0 по x, див. таблицю 2 (з послідовністю розташування векторів у чвертях координат можна експериментувати відповідно до потреб задачі та якщо важливо графічне сприйняття зображення).

ISSN 2663 - 4023

Таблиця 2

Розподіл числових складових UUID за координатними чвертями

№ числа	**Координатна чверть**	**Знак x**	**Знак y**
1	I	+	+
2	II	–	+
3	(на осіY x=0)	0	+
4	IV	+	–
5	III	–	–

Маємо масив векторів A, представлений як масив масивів точок, з врахуванням символьної довжини чисел A[5]={{A[1,1],…,A[1,n1]},{A[2,1],…,A[2,n2]}, {A[3,1],…,A[3,n3]},{A[4,1],…,A[4,n4]},{A[5,1],…,A[5,n5Ї]}}, де n_s – довжина відповідного A[s] числа – вектора масиву.

Тоді X Y – масиви координат для кожного A[s], де s=1,5 за формулами 1.

$$X[s,i] = \begin{cases} x=i, & s=1 \\ x=-1*i, & s=2 \\ x=0, & s=3 \\ x=i, & s=-4 \\ x=-1*I, & s=5 \end{cases} \qquad Y[s,i] = \begin{cases} y=A[s,i], & s=1 \\ y=-1*A[s,i], & s=2 \\ y=A[s,i], & s=3 \\ y=A[s,i], & s=4 \\ y=-1*A[s,i], & s=5 \end{cases} \quad (1)$$

i=1, n_s з кроком 1

По суті це табуляція функції з кроком 1 (крок можна задавати інший), але врахуванням номера вектора у масиві, $f(y) = y * \varphi(s)$ де $\varphi(s)$ – функція для визначення знака числа, яка залежить від номера s числа в масиві A; відрізок табуляції визначається відповідно до номера s або іншими словами чверті (для числа 3 на осі), у якій хочемо розмістити число на графіку. Знаючи номер чверті числа, вираховуємо ще координати початку та кінця самого відрізка табуляції, для наочності вузли табуляції у формулі 5 вираховуються через позначення X [s,i].

Для підвищення надійності можна додати “сіль” — додаткове контрольне число, що розміщується симетрично відносно осі у для A[3] (x=0 а у модуль кожної цифри з від’ємним значенням) - фактично добавляємо ще одне число в масив - A[6]. Ще одним маячком достовірності може бути наявність у солі підмножини, яка складає контрольне число- код, це може бути також контрольна сума цілочисельних координат.

Примітка. *Можна проекспериментувати з самим алгоритмом формування графіка – наприклад при формуванні координат першу половину останнього числа A[5] поміняти на “сіль”, правильні дані вивести на осі у з від’ємним значенням –частково на місці A[6].*

Етап 4. Побудова графіка.

Отримані координати використовуються для побудови лінійного графіка або точкового зображення. Для цього можливо застосувати:

- збереження координат у форматі JSON для подальшої обробки;
- створення зображення у форматі BMP, який зберігає кожну точку з координатною точністю;
- побудову у Excel (.xls**)** – як на рис 7 або за допомогою бібліотек графічного виводу у C# (`System.Drawing`, `ScottPlot`, `OxyPlot` тощо).

Такий графік є унікальним для кожного UUID. У разі побудови зображення у форматі BMP або SVG, можливе також зворотне перетворення — відновлення числових значень із координат графіка та повторна перевірка автентичності UUID.

ISSN 2663 - 4023

Алгоритм для ULID. Для ULID можна теж говорити про представлення у вигляді графіка – для цього для кожного символа отримати ASCII код і створити графік.(наприклад побудувати графік - по осі х – індекс символа, починаючи з 0.1 з кроком 0.1 а по у розташувати саме значення - абсолютну величину цифри). Якщо задачу звести до попередньої - розбити отриманий масив цифр на 5 – можна будувати координати графіка за формулами які наведено для UUID.

Загалом алгоритм конвертації у графічне представлення зводиться до зображеного на блок-схемі (рис.1.), де m- кількість чисел для побудови координат

Рис.1. Блок схема методу графічної конвертації ідентифікаторів типу UUID

ISSN 2663 - 4023

X	Y	Номер у масиві А
0	4	3
0	5	
0	8	
0	6	
1	3	1
2	6	
3	5	
4	5	
5	8	
6	3	
7	6	
8	0	
9	3	
10	6	
-1	2	2
-2	3	
-3	8	
-4	9	
-5	9	
1	-4	4
2	-2	
3	-5	
4	-3	
5	-8	
-1	-5	5
-2	-7	
-3	-8	
-4	3	
-5	-3	
-6	-6	
-7	-3	
-8	0	
-9	-4	
-10	-9	
-11	-1	
-12	-3	
-13	-4	
-14	-4	

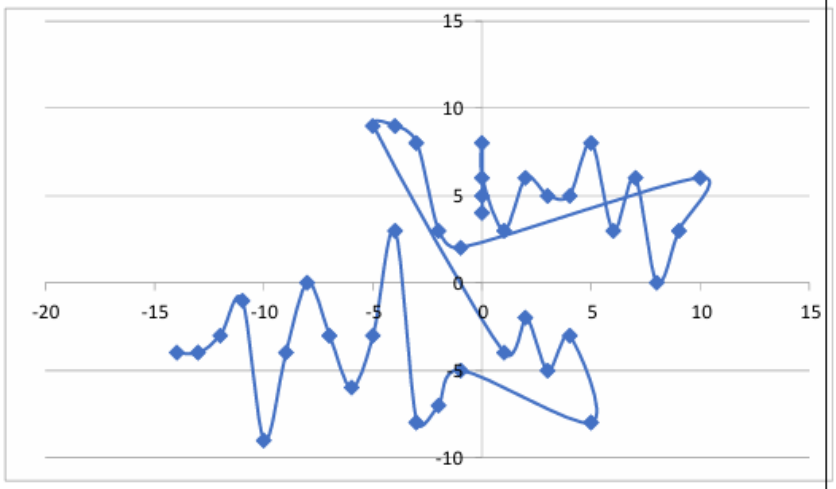

Рис2 Координати та зображення UUID, згенероване у ексел

Якщо на графічному зображенні, на зразок побудованого для нашого прикладу (рис.2), будуть відсутні координатні позначення, за якими можливо визначити масштаб та систему відліку, початок координат - то здійснити зворотне перетворення піксельних значень у числові координати практично неможливо. Параметри, що формують унікальну конфігурацію графічного простору:

- координати початку (0,0);
- крок між точками;
- масштаб (пікселі на одиницю);
- колірна схема або фрактальний порядок.

Таким графічним способом можна задати UUID вхід на важливу інформацію (або й саму інформацію – бо метод можна доробити для відображення символів що мають числовий ascii код), На відміну від QR-кодів, структура графічного хешу не є стандартизованою, а тому може бути розшифрована лише сервером, що володіє параметрами генерації та алгоритмом декодування.

Захист даних та цілісності і шляхи до відновлення ідентифікаторів.
Підходи до методики захисту вебдодатків на основі розробленого методу.
Застосування методу у вебдодатках

При передачі ідентифікаторів між клієнтом і сервером масив координат або бінарне зображення хеш-графіка можна зберігати в куках або в прихованому полі форми зберігати координати (достатньо лише координати у, бо серверу відомі координати х, обрахунок яких закладено в алгоритмі конвертації). До методики бажано поєднувати стандартні захисні практики (HttpOnly, Secure, SameSite) і поєднувати з кодуванням UUID за допомогою розробленого методу. При корегуванні запису пропонуються передавати або масив координат (графічний метод з координатами) або зображення графіка (графічне зображення або хеш)**,** що передається у куках. У разі використання координат, передача відбувається в прихованих полях форми або в куках, а самі координати можуть бути додатково зашифровані або модифіковані за певним ключем (наприклад, зсувом за "шифром Цезаря" з використанням персоналізованого параметра — індивідуальних особливостей користувача чи запису про нього або хешу його сесії чи з застосуванням складніших алгоритмів шифрування). Такий підхід підвищує рівень захисту, адже без знання алгоритму шифрування відновити справжнє значення UUID практично неможливо. Тут слід зазначити, що оскільки по осі х буде завжди та сама послідовність координат, тому передавати, чи зберігати достатньо лише координати у. При такому форматі даних хакеру важче буде зорієнтуватися , що в куках чи в прихованому полі знаходиться унікальний ідентифікатор. Порівняння запропонованого методу з існуючими способами захисту від IDOR наведено у таблиці 3.

Таблиця 3

Порівняння способів захисту від підробки ідентифікаторів IDOR .

Підхід	Переваги	Недоліки
Сесійне збереження ID	Простота реалізації; не потребує криптографії	Необхідність зберігання стану; ускладнення масштабування
HMAC-підпис	Підтверджує цілісність без збереження стану	Потребує секретного ключа; складність адміністрування
CSRF-токен	Захист від міжсайтових запитів; інтеграція у форми	Не захищає сам ідентифікатор UUID від розкриття
Графічний хеш (зображення в куках)	Не потребує збереження стану; візуальна перевірка автентичності	Витрати ресурсів сервера на побудову/розшифрування; можливість реконструкції
Графічний метод з координатами (приховане поле)	Додатковий рівень шифрування; суміщення з анти-CSRF маркером	Підвищена обчислювальна складність; залежність від алгоритму шифрування

Комбінований метод застосування

Щоб мінімізувати ризики, наведені у таблиці вище пропонуємо використовувати гібридну модель, у якій зашифровані за даним методом координати зберігаються у прихованому полі форми клієнта, а на сервері перевіряються за хешем, збереженим у сесії**.** Це поєднує переваги обох методів — контроль цілісності та безпеку без розкриття самого UUID. Такий механізм діє аналогічно до анти-CSRF токена, але має розширену функцію — верифікацію ідентифікатора та його приховування**.**

Візуальна верифікація

З точки зору користувача, відображення на сторінці власного "графічного підпису" або маркера UUID створює додатковий візуальний сигнал безпеки — ознаку, що він перебуває на автентичному сайті, який володіє справжнім алгоритмом генерації графіка.

Користувачу можна показувати на його сторінці наприклад графік на даних ідентифікатора та числового відображення дати і часу. Отже такий підхід забезпечує:

- для сервера — підтвердження, що запит надіслано авторизованим користувачем;
- для користувача — гарантію, що він взаємодіє з автентичним сервером або вебресурсом;

Збереження та відновлення даних після атак

Якщо реалізувати зворотну конвертацію графіка в UUID, це дозволить відновлювати значення ідентифікатора після пошкодження. Навіть без повної інверсії, порівняння збереженого масиву координат або хешу графіка дозволяє оперативно виявити спотворення ключів. Серіалізація координат і збереження додаткових копій у відокремлених полях дають змогу виконати швидку валідацію цілісності без підняття архівних копій.

Застосування у критичних сферах і можливість централізованого аудиту

Для критично важливих обробок персональних даних і транзакцій доцільно передбачати централізований журнал/реєстр (аналог блокчейну) для фіксації фактів генерації та використання одноразових хеш-графіків. Такий підхід забезпечить простежуваність операцій, контроль використання маркерів та зниження ризику зловживань.

З іншого боку графік має візуальну підлаштованість під підпис людини, може бути розміщеним поруч із ним і виконувати роль маркера достовірності документа.

Інші сфери застосування

Метод можна інтегрувати в IoT, робототехніку та системи аутентифікації пристроїв: кожен пристрій може генерувати власний хеш-графік як додатковий маркер ідентичності. Графічне представлення доповнює існуючі ідентифікаційні методи (RFID, QR, штрих-коди) і може бути використане для візуальної верифікації, одноразових маркерів доступу, або як допоміжний механізм для забезпечення доступності сервісів (наприклад, для користувачів з інвалідністю).

Порівняльний аналіз продуктивності та безпеки запропонованого методу

Для оцінки ефективності запропонованого графічного підходу до захисту UUID проведено порівняльний аналіз за двома ключовими критеріями — продуктивність та рівень безпеки**.** У дослідженні враховано типові сценарії використання ідентифікаторів у вебдодатках: створення, передача, перевірка достовірності, валідація під час редагування записів.

Продуктивність

У базовій реалізації UUID (текстовий формат типу 550e8400-e29b-41d4-a716-446655440000) обробка ідентифікатора займає незначний час. У запропонованому методі UUID перетворюється у масив координат та формується графічне зображення.

Основними факторами, що впливають на продуктивність, є:

- обчислення координат (≈ O(n), де *n* — кількість символів UUID),
- шифрування та можливий зсув значень осі *y*,
- генерація графіка (витрати ресурсів процесора або GPU на візуалізацію).

Під час експериментального тестування (на прикладі середовища ASP.NET Core) середній час генерації графічного коду для одного UUID становив 3–7 мс**,** що є прийнятним для інтерактивних вебзапитів. У випадку зберігання лише масиву координат без побудови зображення — час скорочується до 1–2 мс**.**

Таким чином, графічний метод може застосовуватись у реальному часі без суттєвого навантаження на сервер, особливо якщо використовувати кешування або асинхронну обробку.

Рівень безпеки

З точки зору безпеки, перевагою методу є те, що UUID не передається у відкритому вигляді. Зловмисник, який перехопить куки або приховане поле форми, отримає лише набір координат або зображення, не здатне без знання алгоритму конвертації відтворити початковий ідентифікатор.

Для додаткового захисту реалізовано можливість випадкового зсуву (offset) або модифікації даних по осі Y згідно з персоналізованим ключем користувача. Це забезпечує ефект псевдовипадкового кодування, що значно підвищує стійкість до атак типу *replay або pattern matching*.

Рівень стійкості до типових атак можна узагальнити в таблиці 4 :

Таблиця 4

Рівень стійкості до типових атак

Тип атаки	Класичний UUID	UUID із HMAC	Запропонований графічний метод
IDOR	високий ризик підміни	низький ризик (контроль підпису)	дуже низький ризик (шифрування + візуальне маскування)
CSRF	залежить від наявності токена	частково захищений	захищений: координати можуть виступати анти-CSRF маркером
XSS	можливість витоку через DOM	дані у куках з HttpOnly	куки або приховані поля зашифровані, доступ обмежений
SQL Injection	UUID передається як текст	підпис захищає лише від зміни	графічний або числовий масив не дозволяє ін’єкції
Replay Attack	можлива при фіксованому ID	контроль часу HMAC	координати містять часову складову або динамічний зсув

Таким чином, запропонований метод забезпечує комбінований рівень захисту**,** коли графічна форма UUID одночасно виконує функції шифрування, маскування та маркування достовірності.

Порівняльний аналіз показав, що запропонований метод графічного хешування UUID має такі ключові переваги:

- поєднання функцій ідентифікації та криптографічного захисту в одному елементі;
- можливість адаптації під анти-CSRF механізми;
- підвищена стійкість до підміни ідентифікаторів (IDOR) та ін’єкцій;
- додатковий рівень візуальної автентифікації для користувача.

Основним недоліком є додаткові обчислення при конвертації UUID у графічну форму. Проте для сучасних серверів це не становить критичного навантаження, особливо якщо шифрування реалізовано асинхронно або з попереднім кешуванням.

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

Графічне подання UUID є перспективним напрямом для розвитку механізмів безпеки вебдодатків. Для низки завдань у вебсередовищах згенерований графік, створений на основі UUID або LUID, може одночасно виконувати дві функції — приховування унікальних ідентифікаторів та маркування достовірності запитів і форм. Запропонований комбінований метод може доповнювати або частково замінювати традиційні токени безпеки (HMAC, CSRF) у сценаріях, де необхідно забезпечити одночасно захист від підробки, контроль достовірності та візуальне підтвердження справжності. Такий підхід дозволяє не лише відтворювати графічне зображення, а й

працювати з масивом вхідних координат, що створює можливість шифрування даних та захисту ідентифікаторів, зокрема під час взаємодії між сервером і клієнтом.

У сфері збереження даних метод дає змогу контролювати цілісність інформації, виявляти зміни під час атак і прискорювати процес відновлення після них. Окрім цього, використання запропонованого підходу за аналогією з хешуванням дозволяє досягати основних цілей, притаманних традиційним хеш-функціям. Водночас існує ймовірність зворотної конвертації графіка у UUID, тому для безпечного застосування методу як функції графічного хешування доцільно поєднувати його із шифруванням координат, що підвищує рівень захисту та робить його придатним для надійного використання.

Запропонований метод може бути адаптований до звичайних числових ідентифікаторів або символів ASCII-коду, його можна базувати на застосуванні секрету сервера під час побудови графіка, шляхом варіювання параметрів, які формують унікальну конфігурацію графічного простору.

Таким чином, метод відкриває широкі перспективи для подальших досліджень у галузі захисту даних, автентифікації запитів і візуального контролю достовірності інформації, що визначає доцільність його подальшого наукового та прикладного розвитку.

Дослідити доцільність застосування методу в галузі Інтернету речей та інших прикладних областях, де використовуються QR-коди. Оскільки метод базується на графічному поданні, яке може відображатися не лише в лінійній формі, доцільно розглянути можливість його доопрацювання та впровадження у фізико-механічній сфері, зокрема для створення механізмів на зразок унікальних замків.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Kostiuk, Yu. V., Skladannyi, P. M., Bebeshko, B. T., Khorolska, K. V., Rzaieva, S. L., & Vorokhob, M. V. (2025). Information and communication systems security. [Textbook] Kyiv: Borys Grinchenko Kyiv Metropolitan University.
2. Kostiuk, Yu. V., Skladannyi, P. M., Hulak, H. M., Bebeshko, B. T., Khorolska, K. V., & Rzaieva, S. L. (2025). Information security systems. [Textbook] Kyiv: Borys Grinchenko Kyiv Metropolitan University.
3. Hulak, H. M., Zhyltsov, O. B., Kyrychok, R. V., Korshun, N. V., & Skladannyi, P. M. (2023). Enterprise information and cyber security. [Textbook] Kyiv: Borys Grinchenko Kyiv Metropolitan University.
4. Ronaldo, O. (n.d.). *GUID vs UUID vs ULID: Understanding unique identifiers*. Medium. https://medium.com/@ronaldo.oliver7/guid-vs-uuid-vs-ulid-understanding-unique-identifiers-565c88cdca13
5. Kaur, G., Mehta, S., & Singh, P. (2022). Comparative analysis of GUID, UUID and ULID for scalable data architectures. *International Journal of Computer Science and Network Security, 22*(5), 112–120.
6. Penar, M. (2020). Performance analysis of write operations in identity and UUID ordered tables. *Scientific Journal of Rzeszów University of Technology*, 81–96. https://doi.org/10.7862/re.2020.6
7. Momryk, Y., & Sabodashko, D. (2023). Numeric fields in database development: From optimal design to secure coding—SQL attacks protection method. *CEUR Workshop Proceedings, 3456*, 201–212.
8. Agarwal, V., Singh, R., & Patel, M. (2023). Performance optimization in UUID-based large-scale database systems using multi-criteria decision methods (MPE). *International Journal of Computer Applications, 184*(2), 45–52.
9. Chen, L., & Li, X. (2021). Synchronization mechanisms for distributed SQLite databases using UUIDs. *Open Automation and Control Systems Journal, 9*(4), 2201–2208.
10. Thurman, T. R., et al. (2024). *Research results and recommendations for universally unique identifiers in product data standards*. U.S. Department of Commerce, National Institute of Standards and Technology.
11. Google AdSense. (2025). *How to prevent misuse of user identifiers*. https://support.google.com/adsense/answer/6156630?hl=uk
12. Jadhav, V., Chouhan, K. I., & Maskar, V. B. (2019). Smart voting through UID verification by using face recognition. *International Journal of Engineering Trends and Technology, 6*(1), 45–51.

13. Liangong, S. (2015). Research on the construction and realization of synchronization system for wireless spatial database based on UUID. *Open Automation and Control Systems Journal*, *7*, 2201–2206.
14. Triebel, D., Reichert, W., Bosert, S., Feulner, M., Okach, D. O., Slimani, A., & Rambold, G. (2018). A generic workflow for effective sampling of environmental vouchers with UUID assignment and image processing. *Database*. Article ID bax096. https://doi.org/10.1093/database/bax096
15. Ullah, F., Edwards, M., Ramdhany, R., Chitchyan, R., Babar, M. A., & Rashid, A. (2018). Data exfiltration: A review of external attack vectors and countermeasures. *Journal of Network and Computer Applications, 101*, 18–54. https://doi.org/10.1016/j.jnca.2017.10.016
16. Pratama, H., & Rhusuli, M. (2022). IDOR vulnerability and its mitigation techniques in modern web systems. *International Journal of Information Security and Privacy, 10*(2), 77–85.
17. Khurana, P., & Bindal, P. (2014). CSRF vulnerabilities and defensive techniques. *International Journal of Computer Trends and Technology, 13*(3), 135–138. https://doi.org/10.14445/22312803/IJCTT-V13P135

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 3 (31), 2025
ISSN 2663 - 4023

Yaroslava Momryk
Assistant of Department of Information Security Lviv Politecnik, Lviv, Ukraine
ORCID: 0009-0001-4114-0347
Yaroslava.b.momryk@lpnu.ua

APPROACHES TO SECURE DATA STORAGE AND PROCESSING BASED ON UUID IDENTIFIERS: THE GRAPHICAL CONVERSION METHOD

Abstract. The article analyzes current trends in the use of unique identifiers of the UUID (Universal Unique Identifier) type in databases for creating key records. It highlights areas of application of such identifiers in software systems across various domains and identifies relevant challenges related to their use in the context of modern data security requirements. The advantages of ULID (Universally Unique Lexicographically Sortable Identifier)—next-generation identifiers distinguished by higher performance and the ability to support lexicographical sorting through a monotonic generation principle—are examined. Based on an analysis of modern data protection needs, it is concluded that UUID-based identifiers require a high level of security, particularly against SQL (Structured Query Language), XSS (Cross-Site Scripting), CSRF (Cross-Site Request Forgery), and Insecure Direct Object Reference (IDOR) attacks. It is emphasized that protection methods must consider the specifics of the data types on which key identifiers are built. As an extension of security tools, a method for converting UUIDs into a graphical form visually resembling a unique chart is proposed. An algorithm and a diagram for determining the coordinates of UUID and ULID are developed, with an example of transformation and visual representation provided. The forms and areas of application of the proposed method are discussed, including the possibility of reverse conversion for integrity verification, masking in client–server interactions, data reconciliation, and recovery after attacks or failures. The use of both the graphical image itself and the coordinate array that forms it is described. Approaches to encoding and encrypting coordinate data at the stage of chart generation and rendering are proposed. The method enables concealing an identifier in graphical form, which complicates its identification during unauthorized access (for example, in the case of cookie theft) and can also serve as a graphical hash. A combined method of application in web systems is proposed: encrypted coordinates generated by this method are stored in a hidden field of the client form and verified on the server based on session data. This approach ensures identifier authenticity verification (protection against IDOR) while simultaneously functioning as an anti-CSRF token. In the field of data storage, approaches are proposed for applying the method to integrity verification and data recovery after security incidents without the immediate need to restore backup copies. The proposed conversion method can be implemented either as a graphical hash or as a protected copy. It allows operations with both the graphical image and the coordinate array in a format that is more secure compared to symbolic representation. The method has several advantages over QR codes: unlike the latter, the structure of the graphical hash is not standardized and therefore can only be decoded by a server possessing the generation parameters and decoding algorithm, while also providing the possibility of visual output. The proposed UUID-to-graph conversion method can help address several applied problems and enhance tools for ensuring data security and confidentiality across all areas where such identifiers are used.

Keywords: identifier encoding; graphical hash representation; data security; data integrity; protection of key identifiers; UUID; ULID; web application security; unique identifiers as a graph; CSRF protection; IDOR protection.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Kostiuk, Yu. V., Skladannyi, P. M., Bebeshko, B. T., Khorolska, K. V., Rzaieva, S. L., & Vorokhob, M. V. (2025). Information and communication systems security. [Textbook] Kyiv: Borys Grinchenko Kyiv Metropolitan University.
2. Kostiuk, Yu. V., Skladannyi, P. M., Hulak, H. M., Bebeshko, B. T., Khorolska, K. V., & Rzaieva, S. L. (2025). Information security systems. [Textbook] Kyiv: Borys Grinchenko Kyiv Metropolitan University.
3. Hulak, H. M., Zhyltsov, O. B., Kyrychok, R. V., Korshun, N. V., & Skladannyi, P. M. (2023). Enterprise information and cyber security. [Textbook] Kyiv: Borys Grinchenko Kyiv Metropolitan University.

4. Ronaldo, O. (n.d.). *GUID vs UUID vs ULID: Understanding unique identifiers*. Medium. https://medium.com/@ronaldo.oliver7/guid-vs-uuid-vs-ulid-understanding-unique-identifiers-565c88cdca13
5. Kaur, G., Mehta, S., & Singh, P. (2022). Comparative analysis of GUID, UUID and ULID for scalable data architectures. *International Journal of Computer Science and Network Security, 22*(5), 112–120.
6. Penar, M. (2020). Performance analysis of write operations in identity and UUID ordered tables. *Scientific Journal of Rzeszów University of Technology*, 81–96. https://doi.org/10.7862/re.2020.6
7. Momryk, Y., & Sabodashko, D. (2023). Numeric fields in database development: From optimal design to secure coding—SQL attacks protection method. *CEUR Workshop Proceedings, 3456*, 201–212.
8. Agarwal, V., Singh, R., & Patel, M. (2023). Performance optimization in UUID-based large-scale database systems using multi-criteria decision methods (MPE). *International Journal of Computer Applications, 184*(2), 45–52.
9. Chen, L., & Li, X. (2021). Synchronization mechanisms for distributed SQLite databases using UUIDs. *Open Automation and Control Systems Journal*, *9*(4), 2201–2208.
10. Thurman, T. R., et al. (2024). *Research results and recommendations for universally unique identifiers in product data standards*. U.S. Department of Commerce, National Institute of Standards and Technology.
11. Google AdSense. (2025). *How to prevent misuse of user identifiers*. https://support.google.com/adsense/answer/6156630?hl=uk
12. Jadhav, V., Chouhan, K. I., & Maskar, V. B. (2019). Smart voting through UID verification by using face recognition. *International Journal of Engineering Trends and Technology, 6*(1), 45–51.
13. Liangong, S. (2015). Research on the construction and realization of synchronization system for wireless spatial database based on UUID. *Open Automation and Control Systems Journal, 7*, 2201–2206.
14. Triebel, D., Reichert, W., Bosert, S., Feulner, M., Okach, D. O., Slimani, A., & Rambold, G. (2018). A generic workflow for effective sampling of environmental vouchers with UUID assignment and image processing. *Database*. Article ID bax096. https://doi.org/10.1093/database/bax096
15. Ullah, F., Edwards, M., Ramdhany, R., Chitchyan, R., Babar, M. A., & Rashid, A. (2018). Data exfiltration: A review of external attack vectors and countermeasures. *Journal of Network and Computer Applications, 101*, 18–54. https://doi.org/10.1016/j.jnca.2017.10.016
16. Pratama, H., & Rhusuli, M. (2022). IDOR vulnerability and its mitigation techniques in modern web systems. *International Journal of Information Security and Privacy, 10*(2), 77–85.
17. Khurana, P., & Bindal, P. (2014). CSRF vulnerabilities and defensive techniques. *International Journal of Computer Trends and Technology, 13*(3), 135–138. https://doi.org/10.14445/22312803/IJCTT-V13P135