УДК 004.056.55:004.032.26

Ю. В. Баришев, канд. техн. наук, доц.; В. В. Казміревський

АНАЛІЗ АТАК НА ОСНОВІ МУЛЬТИКОЛІЗІЙ НА ДЕРЕВОПОДІБНІ ГЕШ-ФУНКЦІЇ

Стаття присвячена аналізу мультиколізійних атак на деревоподібні геш-функції, що застосовуються у постквантових підписах, блокчейнах та системах довготривалого зберігання. Розглянуто ключові класи атак: мультиколізії Joux (Joux-multicollision), похідні від мультиколізій Joux, колізії на основі зграйного підходу (Herding-based Tree Collision), прищеплення геш-дерева (Hash Tree Grafting) та позиційно незалежні мультиколізії (Position-Blind Multicollision) з акцентом на структурні передумови їх успішності. Показано, що позиційна нейтральність істотно знижує вартість побудови мультиколізій і ускладнює їх виявлення. Перевірка лише за кореневим гешем може не виявити компрометації без додаткових структурних маркерів і доступу до проміжних вузлів. Обговорено, що зі збільшенням глибини дерева можливе експоненційне здешевлення пошуку мультиколізій за відсутності маркування рівнів/позицій. Окреслено формальні передумови вразливостей, успадкованих від ітеративної природи дерев: ланцюговий ефект поширення локальних колізій угору та вплив повторного використання блоків (freq) у узагальнених ICE/TCE-подібних конструкціях. Підкреслено, що навіть за стійкої базової функції ущільнення відсутність позиційного кодування, доменного розмежування та типізації вузлів створює умови для комбінування локальних колізій у великі сімейства повідомлень з незмінним кореневим гешем. На цій основі сформульовано практичні рекомендації, сумісні з вимогами продуктивності: суворе позиційне кодування (ідентифікація рівня, ліво/право-позиції, ролі вузла); доменне розмежування для всіх типів входів функції ущільнення; явна типізація листків та внутрішніх вузлів; контроль/обмеження повторного використання блоків у дереві; журналювання й вибіркове зберігання проміжних гешів для структурного аудиту. Такі заходи руйнують передумови ефективних атак мультиколізійного типу і зменшують ймовірність непомітної підміни піддерев за поліноміальний час. Практична значущість висновків охоплює постквантові системи підпису, розподілені реєстри та архівні платформи, де цілісність і автентичність даних критично залежать від архітектури геш-дерева. Результати можуть слугувати методичною основою для проєктування та оцінювання деревоподібних геш-структур з підвищеною стійкістю до мультиколізійних атак, а також для подальшої формалізації меж складності з урахуванням глибини, політик логування та процедур виявлення аномальних піддерев у реальних протоколах.

Ключові слова: *атака мультиколізій, деревоподібні геш-функції, криптографічна стійкість, атака Joux, блокчейн, криптоаналіз, гешування, перевірка цілісності, захист даних.*

Вступ

Деревоподібні геш-функції широко застосовуються у постквантових підписах (наприклад, SPHINCS+), блокчейнах, системах електронного підпису та довготривалого архівування, оскільки забезпечують паралельність обчислень і локальну перевірку цілісності великих даних. Попри ці переваги, саме архітектура геш-дерева визначає уразливість до мультиколізій: відсутність позиційного маркування, доменного розмежування та структурних ідентифікаторів вузлів створює умови для експоненційного розмноження колізійних шляхів без зміни кореневого гешу.

Практика та відомі результати криптоаналізу демонструють, що мультиколізії можуть бути досягнуті значно дешевше за «ідеальний» випадок завдяки ітеративності та ланцюговому ефекту в деревах. Це проявляється у низці атак, зокрема у мультиколізіях Joux (нарощування колізій шар за шаром), у зграйному підході (діамантова структура з офлайн-підготовкою), у прищепленні геш-дерева (структурне «прищеплення» піддерев через фолт в OTS) та позиційно нейтральних підходах (експлуатація позиційної нейтральності). Стандартні перевірки на рівні кореневого гешу часто не виявляють таку компрометацію, що створює ризики для транзакційної цілісності, цифрової ідентичності та історії змін у розподілених реєстрах.

Метою роботи є вдосконалення захисту деревоподібних геш-функцій шляхом виявлення критичних вразливостей до мультиколізійних атак та обґрунтування підходів до підвищення їхньої криптографічної стійкості.

Для досягнення мети необхідно виконати такі завдання:

1. Проаналізувати відомі підходи до побудови мультиколізій на деревоподібні геш-функції.

2. Виконати порівняльний аналіз відомих атак.

Отримані результати й аналітичний огляд формують теоретичне підґрунтя для подальшого розвитку механізмів криптографічного захисту інформації та оптимізації деревоподібних схем гешування в умовах сучасних кіберзагроз.

Основи деревоподібних геш-функцій та їх структурні вразливості

Деревоподібні геш-функції є розширенням класичних ітеративних геш-функцій, що дозволяють ефективно обробляти великі набори даних через ієрархічну структуру. Вони базуються на ідеї бінарного або t-арного дерева, де обчислення гешу відбувається рекурсивно від листків до кореня, який представляє весь набір. Ця конструкція, вперше запропонована Ральфом Мерклом у 1979 році [1], забезпечує властивості, такі як ефективна перевірка цілісності з логарифмічною складністю O(log k) для k блоків та стійкість до локальних змін. Однак, структурна подібність до ітеративних функцій робить їх вразливими до атак, що експлуатують колізії [2].

Формально, деревоподібна геш-функція H визначається на основі функції ущільнення f(·), яка перетворює фіксовану довжину входу на коротший вихід. Для бінарних дерев функція ущільнення зазвичай має вигляд:

$$f : \{0,1\}^{2n} \to \{0,1\}^{n}, \tag{1}$$

де n – розмір гешу в бітах. Повідомлення m довільної довжини розбивається на k блоків фіксованої довжини: $m = m_1 \| m_2 \| \dots \| m_k$, де ‖ позначає конкатенацію. Листки дерева – це геші окремих блоків (наприклад, $h(m_i)$), а внутрішні вузли обчислюються як геш конкатенації гешів їхніх дітей. Корінь дерева є остаточним значенням H(m), що представляє весь набір даних.

Обчислення для бінарного дерева можна описати рекурсивно. Нехай v – вузол дерева з лівою дитиною left та правою дитиною right. Тоді геш вузла визначається як:

$$h(v) = f(h(left) \| h(right)), \tag{2}$$

де h(left) і h(right) – гешіпіддерев, а для листків:

$$h(leaf) = h(m_i), \tag{3}$$

де h – базова геш-функція. Ця формула візуально ілюструє злиття двох n-бітних гешів у один n-бітний через конкатенацію та функцію ущільнення f(·). Якщо дерево незбалансоване або має змінну глибину, процес повторюється рекурсивно до кореня.

$$h(v) = f\big(h(child_1) \| h(child_2) \| \dots \| h(child_t)\big). \tag{4}$$

Це дозволяє паралельну обробку, але збільшує вразливість до колізій, оскільки більше входів (довжина tn) може призводити до більшої ймовірності колізій на нижніх рівнях [3]. Наприклад, для t=4 (квадратичне дерево) вхід функції f(·) матиме довжину 4n, що вимагає адаптації f(·) для збереження стійкості.

Подальші узагальнення включають конструкції з повторним використанням блоків, такі як ICE (iterated concatenated expanded – ітеративна, конкатенована, розширена) та TCE (tree-based concatenated expanded – деревоподібна, конкатенована, розширена). У ICE блоки повідомлення можуть повторюватися в ітеративному ланцюгу з конкатенацією, а в TCE – у деревоподібній структурі. Ключовий параметр – частота використання блоку, позначена як $freq(\alpha) \leq q$, де α – послідовність індексів блоків, а q – максимальна кількість повторів одного

блоку [4]. Наприклад, якщо q=2, блок може з'являтися двічі в дереві, що полегшує атаки на мультиколізії. Формально, для TCE з freq≤q, обчислення кореня включає розширені шляхи, де один блок m_i може входити в кілька вузлівз урахуванням повторів m_i в піддеревах. Це розширення робить TCE подібними до графових геш-функцій, де вузли дерева інтерпретуються як вершини графа з ребрами, що представляють ущільнення [3].

У контексті блокчейнів деревоподібні функції подібні до конструкцій Merkle-Damgard (M-D), де геш обчислюється ітеративно як ланцюг з вектором ініціалізації h_0=IV[2]:

$$h_i = f(h_{i-1} \| m_i). \quad (5)$$

Однак, на відміну від лінійного M-D, дерева вводять паралелізм гілок, дозволяючи одночасну обробку кількох піддерев (наприклад, транзакцій у блоці Bitcoin [4]). Це підвищує ефективність, але також створює додаткові точки для атак: колізія в одній гілці може поширюватися на корінь, подібно до мультиколізій Joux. Наприклад, у деревах Merkle для блокчейну з k транзакціями, паралельні гілки дозволяють атаки на незалежні піднабори, з потенційною складністю $O(2^{n/2} \log k)$ для мультиколізій [3].

Ці конструкції підкреслюють баланс між ефективністю та безпекою: бінарні дерева прості, але вразливі до ланцюгових колізій, t-арні та TCE пропонують паралелізм, але вимагають суворого контролю freq для уникнення вразливостей [5, 6].

Структурні вразливості деревоподібних геш-функцій полягають в їх ієрархічній та ітеративній природі, яка успадкована від класичних конструкцій типу M-D. Ці вразливості полегшують атаки, такі як мультиколізії та другі прообрази, дозволяючи зловмисникам експлуатувати колізії на нижніх рівнях дерева для компрометації всього кореня. Ключовими факторами є ітеративне обчислення та повторне використання блоків повідомлення, що призводить до поширення колізій вгору по структурі [6]. Ітеративність деревоподібних функцій означає, що геш кореня залежить від послідовного (або паралельного) застосування функції ущільнення f(·) на піддеревах. Якщо в нижньому рівні виникає колізія, тобто два різні блоки m_i і m_j дають однаковий геш $h(m_i)=h(m_j)$, то вона може поширюватися вгору через конкатенацію та ущільнення, породжуючи так само колізію батьківських вузлів та кореня дерева. Повторне використання блоків, вимірюване параметром частоти freq(α) посилює цю проблему. Якщо freq(α)>1 (один блок з'являється в кількох вузлах), то колізія в одному екземплярі блоку автоматично створює множинні шляхи для атаки [3, 6]. Формально, для дерева з послідовністю $\alpha=(\alpha_1,\alpha_2,\ldots,\alpha_s)$ (де s – кількість вузлів), частота визначається як:

$$freq(\alpha) = max\{|\{i : a_i = x\}|\}, x \in |1, l|, \quad (6)$$

де l – кількість унікальних блоків, а $|\cdot|$ – кардинальність множини. Якщо freq(α)>1, то колізії в нижніх рівняхпоширюються вгору, створюючи ланцюговий ефект, подібний до мультиколізій Joux [7]. Це відрізняється від ідеальної (випадкової) геш-функції, де колізії незалежні.

Для ідеальної n-бітної геш-функції ймовірність колізії між двома випадковими входами становить приблизно 2^n, а знаходження колізії за допомогою birthday-атаки (парадокс дня народження) вимагає очікуваної складності:

$$O(\sqrt{2^n}) = O(2^{n/2}). \quad (7)$$

Однак, у деревоподібних функціях мультиколізії (набір з k>2 повідомлень з однаковим гешем) значно дешевші через ланцюговий ефект, колізія на одному рівні генерує множинні варіанти на вищих рівнях без додаткових обчислень. Складність для k-мультиколізії в ітеративних (і, за узагальненням, деревоподібних) функціях знижується до:

$$O(2^{n/2} \log_2 k) \quad (8)$$

замість $O(2^{n(k-1)/k})$ для ідеальної функції [7].

Ці вразливості особливо виражені для слабких геш-функцій, таких як MD5 або SHA-1, де вже знайдено практичні колізії. У деревах на базі MD5 (наприклад, в старих P2P-системах

або Tiger Tree Hash) [8], повторне використання блоків (freq> 1) дозволяє зловмиснику генерувати мультиколізії в піддеревах, фальсифікуючи корінь без зміни всього повідомлення. Аналогічно, у блокчейнах з SHA-подібними гешами, довгі ланцюги (з freq> 1 через повторні транзакції) роблять ланцюговий ефект реальною загрозою. Наприклад, якщо freq=2, колізія в одному блоці створює два ідентичні піддерева, поширюючи колізію на корінь з мінімальнимидодатковими обчисленнями.

Ітеративність та freq> 1 перетворюють локальні колізії на глобальні вразливості, роблячи деревоподібні функції менш стійкими, ніж ідеальні. Це слугує основою для атак, детально розглянутих далі, і підкреслює необхідність обмеження freq та використання стійких примітивів.

У деревоподібних структурах атаки фокусуються на ефективному знаходженні таких наборів, експлуатуючи ієрархічну структуру. Колізії в піддеревах поширюються вгору до кореня через рекурсивне обчислення, створюючи ланцюговий ефект. Це робить мультиколізії важливою загрозою, і є особливо небезпечними в системах з великими даними, як блокчейни.

Класична атака Joux та її розширення

Класична атака Joux, запропонована в 2004 році спочатку розроблена для ітеративних геш-функцій типу M-D, але легко узагальнюється на деревоподібні структури завдяки їх ітеративній природі [1, 3, 7]. Механізм атаки базується на послідовному знаходженні колізій і розширенні їх ланцюжком: починаючи з ініціального вектора h_0, зловмисник знаходить колізію на першому блоці, а потім повторює процес для наступних, генеруючи експоненціально зростаючу кількість колізійних повідомлень. Для ітеративної функції з n-бітним гешем, де обчислення відбувається у формулі 4, атака починається з пошуку пари блоків m_1 таm_1', таких, що $f(h0, m1)=f(h0, m1')=h_1$. Потім процес повторюється для h_1, подвоюючи кількість колізійних шляхів на кожному кроці (рис. 1).

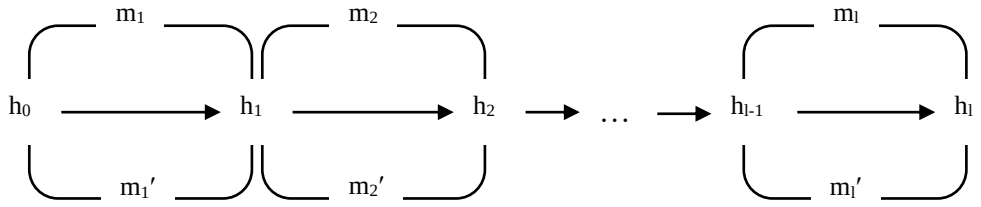

Рис. 1. Схематичне зображення мультиколізії Joux

Складність атаки для знаходження k-мультиколізії (набір з k повідомлень з однаковим гешем) становить $O(2^{n/2}\cdot\log_2 k)$ (формула 7). Це значно ефективніше, ніж для ідеальної геш-функції. Ця перевага виникає через ланцюговий ефект, колізія на ранньому етапі генерує множинні варіанти без повного перебору [6].

Для деревоподібних функцій атака узагальнюється, колізія в піддереві (наприклад, двох листках з однаковим гешем) поширюється на корінь через рекурсивне обчислення (формула 1). Якщо дерево має послідовність $(\alpha_1,\alpha_2,\ldots)$ з частотою $\text{freq}(\alpha)\leq 2$ (максимум два повторення блоку), то атака стає ефективною. Зокрема, Теорема 5Нанді та Стінсона стверджує, що для такої функції існує 2^r-шляхова мультиколізія (набір з 2^r повідомлень з однаковим кореневим гешем) зі складністю[5]:

$$O(r2^n 2^{n/2}), \tag{9}$$

де r – параметр мультиколізії (логарифм від кількості шляхів), 2^n– фактор від розміру гешу, а $2^{n/2}$ – від birthday-пошуку. Це узагальнення базується на пропозиціях 2-4 Нанді та Стінсона, де незалежні елементи в послідовності α дозволяють будувати колізії в незалежних піддеревах, поширюючи їх на корінь[5].

Розширення атаки на більш загальні структури, такі як TCE, запропоновано в [4]. Тут

атака застосовується до дерев G з частотою freq(G)≤q (максимум q повторів блоку), де TCE включає конкатенацію та розширення повідомлень. Складність для 2^k-мультиколізії становить:

$$O(poly(n,k) \cdot 2^{n/2}), \tag{10}$$

де poly(n,k) – поліном від розміру гешу n та k (наприклад, k^3n^3). Алгоритм включає наступні кроки:

1. Знайти незалежні послідовності листків (незалежні підмножини індексів, де немає перетинів піддерев, як визначено в Лемі 6 [5]).

2. Генерувати колізії в кожному піддереві за допомогою birthday-атаки ($O(2^{n/2})$ на піддерево).

3. Поширити колізії вгору, використовуючи незалежність для створення множинних шляхів до кореня.

Приклад: для бінарного дерева з 2^k листками (наприклад, дерева Merkle в блокчейні з 2^k транзакціями) атака генерує 2^k колізійних повідомлень, дозволяючи фальсифікацію історії без зміни кореня [2]. У Bitcoin це могло б означати підміну транзакцій у блоці з $O(2^{128})$ операцій для SHA-256 (n=256), що теоретично можливо для довгих ланцюгів [5]. Це розширення підкреслює, як атака Joux адаптується до TCE, роблячи мультиколізії практичними для q > 1.

Колізії на основі зграйного підходу

Колізія на основі зграйного підходу (Herding-атака), запропонована Kelseyта Kohno в 2006 році, є потужним методом, де зловмисник спочатку обирає цільовий геш h і будує структуру колізій, що дозволяє «зганяти» довільні префікси до цього h через відповідні суфікси. Атака складається з двох фаз: офлайн (підготовка структури) та онлайн (пошук лінки – повідомлення m_{link}, яке, додане до префікса, виведе на один із листків дерева). Ключовим елементом є «діамантова структура» (diamond structure) – повне бінарне дерево колізій, де множина шляхів з листків сходяться до одного кореня. Складність офлайн-фази для створення структури з 2^k шляхів становить$O(2^{(n+k)/2})$[7].

Це ефективніше, ніжпошук, бо структура будується шар за шаром, використовуючи мультиколізії для зменшення обчислень. Для кращого розуміння, наведено схематичну ілюстрацію діамантової структури для 2^3 шляхів (рис. 2), де джерельний вузол з'єднаний з листками, а рівні зменшують кількість вузлів через колізії.

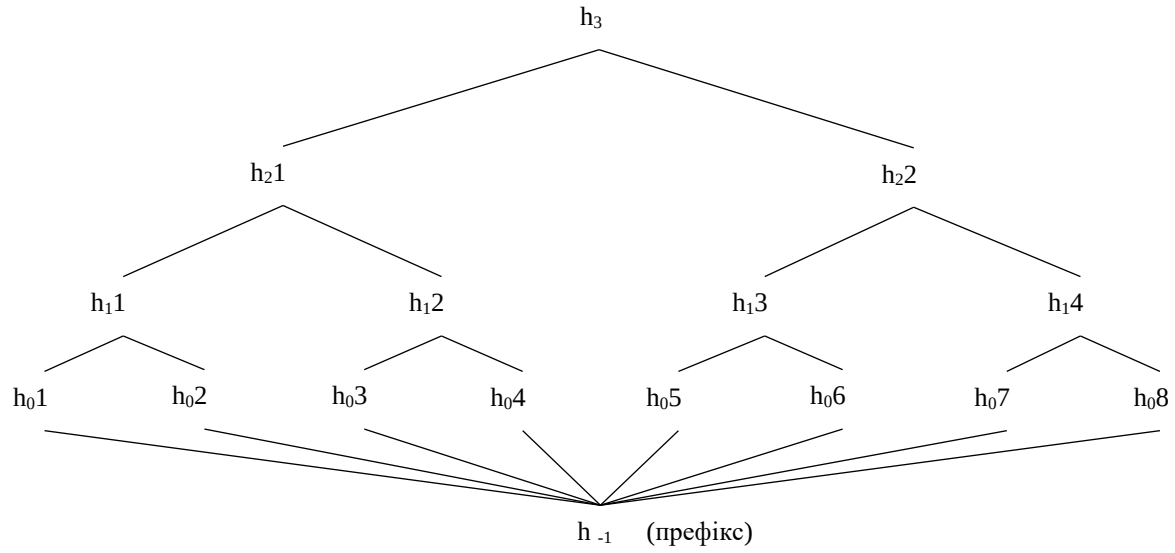

Рис. 2. 2^3 діамантова структура

На схемі джерельний вузол з'єднаний з листками, а кожен рівень генерує колізії,

зменшуючи кількість вузлів удвічі. Це дозволяє «зганяти» будь-який префікс (від листка) до h через відповідний шлях (суфікс)[8]. Алгоритм побудови включає:

1. Генерацію 2^k листків через birthday-атаки.
2. Парування вузлів на кожному рівні для створення колізій.
3. Повторення до кореня.

У Анрєєвої атака розширена на дерева з троянськими повідомленнями: зловмисник вставляє колізійний суфікс для з'єднання довільного префіксу з цільовим гешем [9]. Варіант для дерева з L рівнями має складність:

$$O(2^{n/2-L/2}), \tag{11}$$

де L – глибина дерева. Це розширення включає троянські повідомлення – фіксовані блоки, що дозволяють контролювати шляхи, з офлайн-підготовкою ($O(2^{(n+(L-1)+1)/2})$ для першого шару) та онлайн-лінкуванням.

Атака прищеплення геш-дерев

Атака прищеплення геш-дерева (Hash Tree Grafting Attack) є гібридною атакою, що безпосередньо пов'язана з мультиколізіями в деревоподібних структурах, оскільки створює множинні шляхи (колізійні варіанти) до одного кореня через ін'єкцію помилок. Атака застосовується до SPHINCS – пост-квантової схеми на базі багаторівневих дерев Merkle. Механізм атаки базується на введенні помилки в одноразовому підписі (OTS – one-time signature) на верхньому рівні, що дозволяє генерувати двавалідні підписи з одного OTS-ключа, створюючи мультиколізію: один OTS аутентифікує два різні піддерева (справжнє та фальшиве), сходячись до одного кореня[10]. Формально, якщо помилка генерує фальшиву сигнатуру для фальшивого кореня, то мультиколізія виникає як набір підписів $\{\delta, \delta^*\}$ для повідомлень з однаковим публічним ключем PK, порушуючи властивість одноразовості і дозволяючи компроментацію:

$$\delta_1 = OTS.Sign(sk, M_1), \delta_2 = OTS.Sign(sk, M_2), \tag{12}$$

де sk – приватний ключ OTS, а δ_1 і δ_2 – валідні підписи для різних $M_1 \neq M_2$, що дозволяє прищепити фальшиве дерево з колізійними шляхами [10]. Атака незалежна від базової геш-функції, фокусуючись виключно на структурі багаторівневого дерева, і вимагає лише однієї успішної помилки для компрометації. Складність атаки являє собою суму складності офлайн-фази, яка включає генерацію фальшивого дерева, та онлайн-фази – підбору шляху в прищепленому дереві:

$$O(2^{\lambda}) + O(1), \tag{13}$$

де λ – рівень безпеки, наприклад λ=128 для SPHINCS-128[10]. Більше помилок знижують обчислювальну вартість, генеруючи більше колізійних шляхів.

Далі наведено ілюстрацію багаторівневих дерев SPHINCS з помилкою в OTS на верхньому рівні та «прищепленням» фальшивого піддерева (рис. 3).

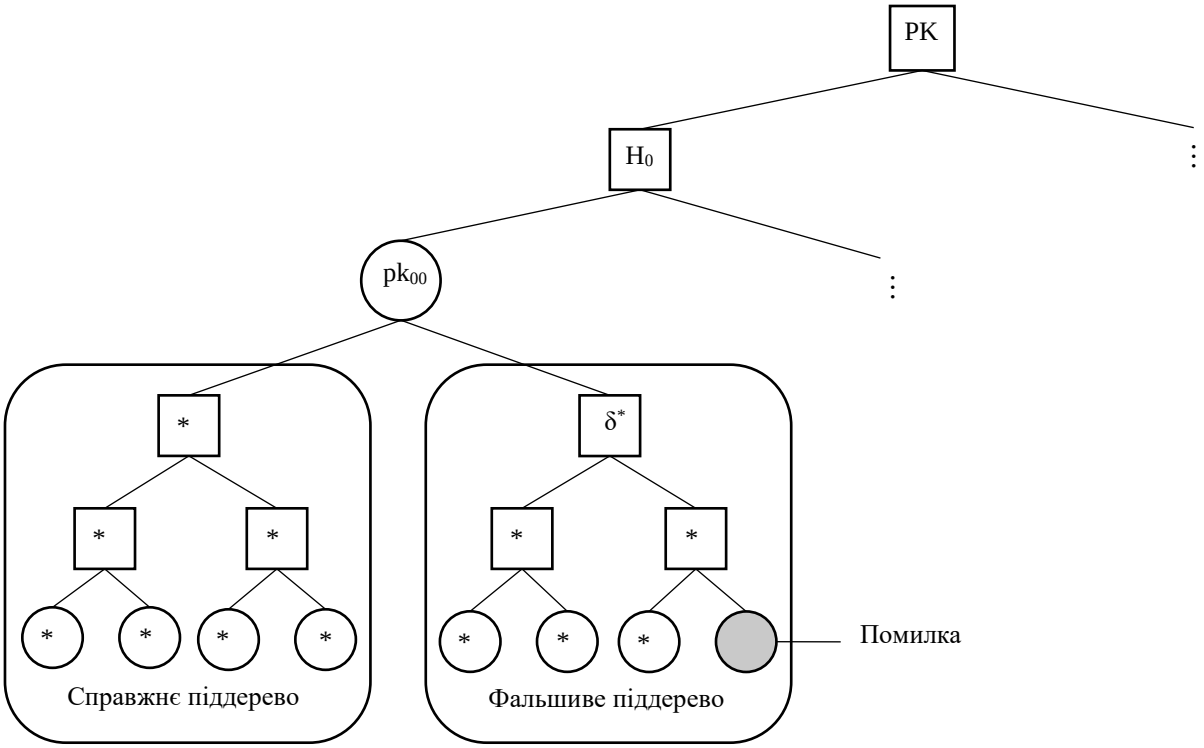

Рис. 3. Принцип Grafting-атаки

Алгоритм атаки включає наступні кроки:

1. Ввести помилку в OTS під час підпису, генеруючи два підписи з одного ключа.

2. Створити фальшиве піддерево, незалежне від секретного ключа, з множинними колізійними листками.

3. Прищепити фальшиве дерево до справжньої структури через колізійний OTS, дозволяючи підробку для будь-якого повідомлення.

Позиційно-незалежна атака

Позиційно-незалежна мультиколізійна атака (Position-Blind Multicollision Attack) є варіацією мультиколізійних атак, де колізії генеруються без залежності від позиції блоків у повідомленні або структурі дерева (blindtoposition), дозволяючи гнучке розміщення колізійних пар у будь-яких вузлах. Механізм атаки базується на пошуку незалежних послідовностей в структурі дерева, де колізії не фіксовані до конкретних позицій, дозволяючи переміщення колізійних блоків без перерахунку всього шляху. Це створює мультиколізію, де колізії можуть бути в будь-якій позиції дерева, незалежно від порядку [11].

Складність атаки для k-мультиколізії в t-арному дереві з позиційною незалежністю становить:

$$O\left(2^{n/2}\right)\log k + poly(\log k). \tag{14}$$

Це ефективніше за класичну атаку Joux для структур з високим ступенем свободи (наприклад, графи з незалежними вершинами), більша незалежність (більше позицій) знижує вартість на фактор log k для t-арних дерев [11]. Алгоритм атаки включає наступні кроки:

1. Знайти незалежні послідовності в дереві (незалежні підмножини індексів, де позиції не перетинаються [12]).

2. Генерувати колізії в кожній незалежній позиції за допомогою birthday-атаки ($O(2^{n/2})$ на інтервал).

3. Поширити колізії до кореня, використовуючи позиційну незалежність для створення множинних шляхів без фіксації позицій [11].

Для кращого розуміння, нижче наведено ілюстрацію графу з незалежними вершинами, де

позиційно-незалежні колізії створюють множинні шляхи до кореня без фіксації позицій (рис. 4).

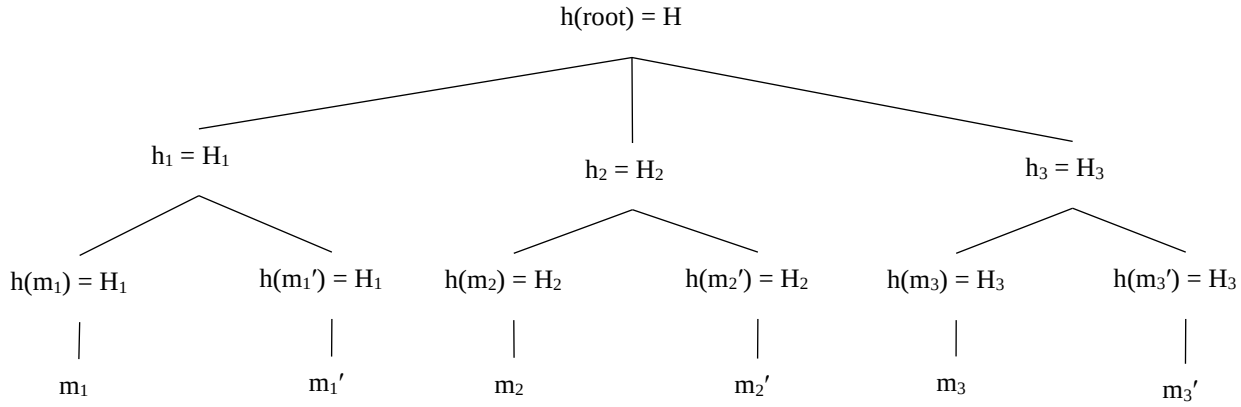

Рис. 4. Принцип позиційно незалежної атаки

На рис. 4 показано, як три незалежні підграфи ($h_1 = H_1$, $h_2 = H_2$,$h_3 = H_3$) містять колізійні пари повідомлень: m_1 та m_1', дають однаковий геш H_1, m_2 та m_2', які дають однаковий геш H_2 і m_3 та m_3', які дають однаковий геш H_3. Ці колізійні пари можуть вільно комбінуватися, створюючи $2^3 = 8$ різних повідомлень з однаковим кореневим гешем H.

Узагальнення результатів аналізу

Окрему увагу приділено структурним передумовам атаки: аналізується, які саме властивості побудови геш-дерева (наявність/відсутність позиційного маркування, жорстка топологія, доменне розмежування) обумовлюють можливість або ефективність реалізації тієї чи іншої мультиколізійної атаки. Це дозволяє ідентифікувати критичні місця архітектури, покликані бути під контролем у разі впровадження таких структур у захищених системах

Ще одним критично важливим критерієм виступає чутливість до модифікацій структури. Оцінюється, наскільки незначна трансформація архітектури (наприклад, додавання позиційних або доменних маркерів) може мінімізувати чи повністю усунути вразливість до певних атак.

Важливим є вплив атаки на криптографічні протоколи, де оцінюється не лише теоретична, а й прикладна значимість вразливості. Аналізується, які саме властивості безпеки (цілісність, автентичність, невідворотність) можуть бути порушені у практичних застосуваннях, таких як блокчейни, системи електронного підпису, схеми гарантованої перевірки цілісності.

Окремо виділено критерій трудності детектування атаки. Він дає змогу оцінити, чи дозволяє поточна архітектура системи своєчасно виявити або запобігти атакуючим діям у реальному середовищі, чи атаку можна реалізувати так, що вона залишиться повністю непоміченою за стандартної експлуатації. Врахування цього аспекту повністю узгоджується із сучасними рекомендаціями щодо моделювання загроз і оцінки ризику для криптографічних примітивів.

Загалом комплекс критеріїв вибраний так, аби дозволити не лише формалізовано порівняти мультиколізійні атаки між собою, а й чітко визначити їхній практичний ризик та найбільш уразливі компоненти сучасних схем деревоподібного гешування.

Відповідно до наведених критеріїв у таблиці 1 представлено порівняльний аналіз основних класів мультиколізійних атак на деревоподібні геш-функції.

Таблиця 1

Порівняльна характеристика атак на основі мультиколізій на деревоподібні геш-функції

Атака / Параметр	Атака Joux	Атака на основі зграйного підхід	Атака прищеплення	Позиційно-незалежна атака
Обчислювальна складність	$O(r2^n 2^{n/2})$	$O(2^{n/2-L/2})$	$O(2^{\lambda}) + O(1)$	$O(2^{n/2}) \log k + poly(\log k)$
Структурні передумови атаки	Ітеративність без позиційного/доменного маркування	Відсутність жорсткого порядку об'єднання, гомогенність рівнів	Топологічний ізоморфізм, відсутність ідентифікаторів/індексації	«Сліпота» функції, відсутність унікальних атрибутів/маркерів
Чутливість до маркування	Висока	Висока	Висока	Висока
Вплив на протоколи	Масова підміна, блокчейн, архіви	Комітмент, підпис, відкладене вибір	Заміна внутрішніх даних, підпис	Теоретична уразливість дерев Merkle
Труднощі детектування	Високі	Високі	Середні	Середні

Аналіз таблиці 1 показує, що ключовим чинником є структура геш-дерева, схеми без позиційного/доменного маркування, із позиційною нейтральністю чи ізоморфізмом рівнів – найуразливіші. У таких побудовах локальні колізії легко комбінуються в експоненційно великі сімейства повідомлень з однаковим кореневим гешем (типово для атак Joux, позиційно-незалежних атак і сценаріїв прищеплення геш-дерева). Навіть мінімальне позиційне чи доменне маркування різко знижує їхню ефективність. Детектування часто утруднене, стандартний контроль кореневого гешу зазвичай не виявляє компрометації, за наявності структурного аудиту можливе виявлення аномальних піддерев (для атак на основі зграйного підходу та позиційно-незалежних атак – «середній» рівень). Наслідки включають пошкодження цілісності, автентичності та можливу підміну історії змін без зміни кореневого гешу. Отже, безпека визначається не лише криптостійкістю примітива, а насамперед архітектурою: маркуванням рівнів, доменним розмежуванням і позиційним кодуванням вузлів.

Висновки

Стійкість деревоподібних геш-структур до мультиколізій визначається насамперед їхньою архітектурою. Відсутність позиційного маркування, доменного розмежування та явної типізації вузлів знижує бар'єр атаки, локальні колізії можна компонувати у великі сімейства повідомлень із незмінним кореневим гешем, а перевірка лише кореневого гешу часто не виявляє компрометації без доступу до проміжних вузлів та структурного аудиту. Узагальнені техніки, атаки Joux і атаки на основі зграйного підходу, стають практично дієвішими саме за позиційної нейтральності та гомогенності рівнів, тоді як у випадках атак прищепленняпозиційно-незалежних атак вирішальними виступають помилки в OTS та наявність незалежних підграфів відповідно. У сукупності це підтверджує, що головний ризик полягає у структурі дерева, а не в ізольованих властивостях функції ущільнення.

Архітектурні недоліки не перетворюють проблему на тривіальну, але суттєво зменшують ефективний запас міцності, підвищують гнучкість атаки й ускладнюють її детектування стандартними засобами. Саме тому доцільно зміцнювати конструкцію дерев задля підвищення стійкості без помітної втрати ефективності з дотриманням та впровадженням саме тих умов, які робитимуть мультиколізійні атаки не здійсненими за поліноміальний час.

Проведений аналіз мультиколізійних атак на деревоподібні геш-функції засвідчив: вирішальним чинником їх успішності є архітектура дерева. Найуразливішими є схеми без позиційного маркування, доменного розмежування та структурної ідентифікації вузлів. За таких умов навіть стійкі функції ущільнення не перешкоджають експоненційному здешевленню пошуку мультиколізій.

Практичні наслідки найбільш відчутні для постквантових підписів на базі багаторівневих дерев Merkle, блокчейнів і систем довготривалого зберігання, порушення цілісності та

автентичності може залишатися непоміченим без структурного контролю проміжних вузлів. Подальші роботи варто спрямувати на формалізацію меж складності для конкретних параметрів глибини, на емпіричну оцінку детектованості за різних політик логування проміжних вузлів, на покращення методів автоматизованих засобів моніторингу аномальних піддерев у реальних системах і на покращення стійкості деревоподібних геш-функцій.

Отримані результати поглиблюють розуміння природи мультиколізійних вразливостей і можуть слугувати основою для вдосконалення наявних та розробки нових схем криптографічного захисту даних у цифрових платформах.

СПИСОК ЛІТЕРАТУРИ

1. Jonathan J. Hoch, Adi Shamir. Breakingthe ICE – Finding Multicollisionsin Iterated Concatenated and Expanded (ICE) Hash Functions. *Sci Space Repository*. 2006. URL: https://scispace.com/pdf/breaking-the-ice-finding-multicollisions-in-iterated-3zzx6moso8.pdf.

2. Nandi M., Stinson D. Multicollision attacks on a class of hash functions. *IEEE Transactions on Information Theory*. 2007. URL: https://www.researchgate.net/publication/3086233_Multicollision_Attacks_on_Some_Generalized_Sequential_Hash_Functions (Last accessed: 25.09.2025).

3. On the similarities between blockchains and Merkle-Damgard hash functions / K. Halunen et al. *2018 IEEE International Conference on Internet of Things, Embedded Systems and Communications (IINTEC)*. 2018. URL: https://www.researchgate.net/publication/327000228_On_the_Similarities_between_Blockchains_and_Merkle-Damgard_Hash_Functions (Last accessed: 25.09.2025).

4. Halunen K. Multicollisions and graph-based hash functions. *Trusted Systems (INTRUST 2011). Lecture Notes in Computer Science*, 2012. Vol. 7222. URL: https://link.springer.com/chapter/10.1007/978-3-642-32298-3_11 (Last accessed: 25.09.2025).

5. Nakamoto S. Bitcoin: A peer to peer electronic cash system. *Bitcoin Whitepaper*. 2022. URL: https://www.zouantcha.com/blog/bitcoin-whitepaper (Last accessed: 25.09.2025).

6. Hoch J. J., Shamir A. Breaking the ICE – finding multicollisions in iterated concatenated and expanded (ICE) hash functions. *Fast Software Encryption (FSE 2006)*. *Lecture Notes in Computer Science*. 2006. Vol. 4047. URL: https://link.springer.com/chapter/10.1007/11799313_12 (Last accessed: 25.09.2025).

7. Nandi M., Stinson D. R. Multicollision attacks on generalized hash functions. *IACR ePrint Archive. Report 2004/330*. 2004. URL: https://iacr.steepath.eu/2004/330-MulticollisionAttacksonGeneralizedHashFunctions.pdf (Last accessed: 25.09.2025).

8. Joux A. Multicollisions in iterated hash functions. Application to cascaded constructions. *Advances in Cryptology – EUROCRYPT 2004*. 2004. URL: https://link.springer.com/chapter/10.1007/978-3-540-28628-8_19 (Last accessed: 25.09.2025).

9. Kelsey J., Schneier B. Second preimages on n bit hash functions for much less than 2^n work. *Advances in Cryptology – EUROCRYPT 2005*. 2005. URL: https://eprint.iacr.org/2021/340.pdf (Last accessed: 25.09.2025).

10. Kelsey J., Kohno T. Herding hash functions and the Nostradamus attack. *Advances in Cryptology – EUROCRYPT 2006*. *Lecture Notes in Computer Science*. 2006. Vol. 4004. URL: https://link.springer.com/chapter/10.1007/11761679_12 (Last accessed: 25.09.2025).

11. Blackburn S. R., Stinson D. R., Upadhyay J. On the complexity of the herding attack and some variants. *Designs, Codes and Cryptography*. 2011. Vol. 59, № 1–3. URL: https://eprint.iacr.org/2010/030.pdf (Last accessed: 25.09.2025).

12. Herding, second preimage and Trojan message attacks beyond Merkle–Damgård / E. Andreeva et al. *Selected Areas in Cryptography (SAC 2009). Lecture Notes in Computer Science*. 2009. Vol. 5867. URL: https://link.springer.com/chapter/10.1007/978-3-642-05445-7_25 (Last accessed: 25.09.2025).

13. Genelle L., Leurent G., Naehrig M. Grafting trees: a fault attack against the SPHINCS framework. *IACR ePrint Archive. Report 2018/102*. 2018. URL: https://eprint.iacr.org/2018/102.pdf (Last accessed: 25.09.2025).

14. Sakura: a flexible coding for tree hashing / G. M. Bertoni et al. *International Conference on Applied Cryptography and Network Security*. 2014. URL: https://eprint.iacr.org/2013/231.pdf (Last accessed: 25.09.2025).

15. Hoch J. J., Shamir A. Breaking the ICE – finding multicollisions in iterated concatenated and expanded (ICE) hash functions. *Sci Space Repository*. 2006. URL: https://scispace.com/pdf/breaking-the-ice-finding-multicollisions-in-iterated-3zzx6moso8.pdf (Last accessed: 25.09.2025).

Стаття надійшла до редакції 01.09.2025.

Стаття пройшла рецензування 25.09.2025.

Баришев Юрій Володимирович – канд. техн. наук, доцент кафедри захисту інформації, e-mail: yuriy.baryshev@vntu.edu.ua.

Казміревський Віталій Віталійович – аспірант кафедри захисту інформації, e-mail: kazmirevskiy1999@gmail.com.

Вінницький національний технічний університет.