УДК 004.056.5

Пуздрановський І.В.
Салієва О.В.

АНАЛІЗ ІСНУЮЧИХ АЛГОРИТМІВ ГЕНЕРАЦІЇ ДИНАМІЧНО ЗМІНЮВАНИХ КЛЮЧІВ

Вінницький Національний Технічний Університет

Анотація

В даній роботі розглянуто алгоритми генерації динамічно змінюваних ключів. Наведено їх особливості, переваги та недоліки. Здійснено порівняння досліджуваних алгоритмів.

Ключові слова: Захист, інформація, генерація.

Abstract

This article discusses algorithms for generating dynamically changing keys. Their features, advantages, and disadvantages are presented. A comparison of the studied algorithms is made.

Key words: Protection, information, generation.

Вступ

У даній роботі здійснено аналіз алгоритмів генерації динамічно змінюваних ключів, які забезпечують високий рівень захищеності інформаційних систем. Особливу увагу приділено оцінці стійкості алгоритмів до атак, складності синхронізації між компонентами системи та їх ефективності в умовах реального використання. Проведений аналіз дозволяє виявити ключові переваги та недоліки розглянутих підходів, а також сформулювати рекомендації щодо вибору алгоритмів для забезпечення надійного захисту інформаційних ресурсів.

Дослідження

HOTP (Hashed-Based One-Time Password) є алгоритмом генерації одноразових паролів, який використовує криптографічну хеш-функцію HMAC (Hash-Based Message Authentication Code). Основна ідея алгоритму полягає у забезпеченні унікальності паролів для кожної сесії через використання спільного секретного ключа і лічильника, які зберігаються як на стороні клієнта, так і на сервері. Для створення одноразового пароля алгоритм використовує секретний ключ і значення лічильника, які проходять через функцію HMAC. Результат хешування обрізається до певного формату (зазвичай 6 або 8 символів) і формується одноразовий пароль. На стороні сервера цей пароль перевіряється на відповідність, після чого значення лічильника збільшується для унікальності наступної сесії [1].

HOTP демонструє високу стійкість до таких атак як повторне використання або перехоплення пароля, оскільки кожен пароль є унікальним і може бути використаний лише один раз. Проте ефективність роботи алгоритму значною мірою залежить від коректної синхронізації лічильника між клієнтом і сервером. У разі втрати синхронізації система може блокувати доступ, що є основним недоліком HOTP. Алгоритм також не має часових обмежень для пароля, що може створювати ризик його використання у випадку компрометації, якщо лічильник не оновлюється після певного періоду.

HOTP використовується в системах автентифікації, де потрібно забезпечити унікальність паролів для кожної сесії. Він ідеально підходить для сценаріїв, де час не є критичним фактором, наприклад, у системах багатофакторної автентифікації або для захисту API через одноразові токени. Перевагою HOTP є простота його реалізації через використання стандартних криптографічних функцій, таких як HMAC-SHA1. Завдяки цьому алгоритм став стандартом, визначеним у RFC 4226, що забезпечує його широке застосування та сумісність із багатьма платформами [1].

Проте, попри високу стійкість до атак типу "відтворення" або фішингу, HOTP має певні обмеження. Найбільш суттєвою проблемою є залежність від синхронізації лічильника між клієнтом і сервером. Розсинхронізація лічильника може призвести до блокування автентифікації, особливо в розподілених системах із високим навантаженням. Відсутність часових обмежень також створює ризик використання скомпрометованого пароля, якщо його не було застосовано раніше.

Для підвищення безпеки алгоритму пропонуються різні вдосконалення. Наприклад, динамічно змінюваний секретний ключ значно підвищує стійкість HOTP, оскільки після кожної сесії ключ оновлюється, що робить компрометацію системи малоймовірною. Додавання часових обмежень для пароля може зменшити ризики, пов’язані з перехопленням. Крім того, використання HOTP у поєднанні

з іншими факторами автентифікації, такими як біометрія або SMS-коди, забезпечує додатковий рівень захисту.

Таким чином, HOTP є ефективним і відносно простим у реалізації алгоритмом генерації одноразових паролів. Він демонструє високу стійкість до більшості атак, але потребує точного налаштування для уникнення розсинхронізації та забезпечення надійності. Для критичних систем із підвищеними вимогами до безпеки доцільно використовувати вдосконалені варіанти HOTP із динамічними ключами або часовими обмеженнями. Це дозволить забезпечити баланс між безпекою, продуктивністю та зручністю використання [2].

TOTP (Time-Based One-Time Password) – це алгоритм генерації одноразових паролів, який базується на часових мітках. Алгоритм є розширенням HOTP (Hashed-Based One-Time Password), але замість лічильника використовує поточний час для формування паролів. Завдяки часовій залежності TOTP забезпечує унікальність паролів для певного інтервалу часу, що підвищує безпеку і робить його особливо ефективним у системах з обмеженим часовим доступом.

Принцип роботи TOTP передбачає, що сервер і клієнт мають спільний секретний ключ і однакову часову базу, синхронізовану через мережу або за допомогою зовнішніх джерел часу. Алгоритм генерує пароль шляхом обчислення HMAC для секретного ключа і поточної часової мітки, розподіленої на задані інтервали (наприклад, 30 секунд). Ця мітка слугує заміною лічильника, який використовується в HOTP, і змінюється динамічно. Для зручності використання результат обчислення хешу обрізається до певного формату (зазвичай, 6 або 8 символів).

TOTP забезпечує високу стійкість до таких атак як повторне використання пароля, оскільки кожен пароль діє лише протягом короткого часу. Часова залежність також мінімізує ризики компрометації, навіть якщо пароль перехоплений, оскільки його термін дії швидко закінчується. Основною перевагою алгоритму є його здатність інтегрувати часові обмеження без необхідності зберігати додаткові параметри, такі як лічильник [2].

Проте TOTP має певні недоліки. Головною проблемою є залежність від точності синхронізації часу між сервером і клієнтом. Якщо часові бази розсинхронізуються через технічні збої або затримки в мережі, автентифікація може бути заблокована. Щоб уникнути цієї проблеми, системи використовують невелике вікно толерантності, дозволяючи перевірку паролів із сусідніх інтервалів часу. Однак це також створює ризики для безпеки, оскільки збільшується кількість допустимих паролів.

Для підвищення рівня захищеності TOTP потребує вдосконалення. Наприклад, можна використовувати більш точні джерела синхронізації часу або комбінувати алгоритм із іншими методами автентифікації, такими як біометрія чи апаратні токени. Крім того, додавання додаткових параметрів, таких як IP-адреса або геолокація, може зробити пароль залежним від контексту і підвищити загальний рівень безпеки [3].

Nonce (Number Once) – це випадкове значення, яке використовується для забезпечення унікальності кожної транзакції або сесії в криптографічних і автентифікаційних системах. Головна особливість Nonce полягає в тому, що кожне згенероване значення може бути використане лише один раз, що запобігає повторному використанню даних або атакам типу "відтворення".

Принцип роботи Nonce базується на генерації випадкового числа або унікального ідентифікатора для кожної операції. Це значення може бути додатково зашифроване або хешоване перед використанням. Nonce передається між клієнтом і сервером разом із запитом або автентифікаційними даними. Сервер перевіряє, чи використовувалося це значення раніше. Якщо Nonce є унікальним, операція вважається дійсною, і сервер зберігає це значення для майбутніх перевірок[3].

Основною перевагою використання Nonce є його стійкість до атак типу "відтворення". Наприклад, якщо зловмисник перехопить запит із даними та спробує повторно його надіслати, сервер відхилить його, оскільки Nonce вже було використано. Це робить Nonce особливо корисним у протоколах автентифікації, електронних транзакціях і криптографічних системах.

Проте Nonce має свої обмеження. Однією з ключових проблем є необхідність синхронізації або збереження використаних значень на сервері для перевірки унікальності, що створює додаткове навантаження на систему і потребує ефективного управління пам’яттю. Крім того, для забезпечення високого рівня безпеки необхідно гарантувати, що згенеровані Nonce є дійсно випадковими та мають достатню довжину, щоб уникнути повторення.

Nonce широко використовується у багатьох сучасних системах. Наприклад, у криптографічних протоколах, таких як TLS, Nonce використовується для запобігання атакам на повторне використання сесійних даних. У блокчейнах Nonce застосовується для створення нових блоків і забезпечення унікальності транзакцій. У системах автентифікації Nonce дозволяє створювати одноразові токени для підтвердження дій користувачів [4].

Для підвищення безпеки Nonce може бути вдосконалений. Наприклад, використання криптографічно стійких генераторів випадкових чисел забезпечує унікальність значень навіть за високих навантажень.

Удосконалений HOTP (Hashed-Based One-Time Password) представляє собою розширення базового алгоритму HOTP, що забезпечує підвищений рівень безпеки завдяки динамічно змінюваним секретним ключам. Це вдосконалення спрямоване на вирішення основних обмежень базового HOTP, таких як залежність від синхронізації лічильника та вразливість до компрометації ключів. Удосконалений HOTP поєднує сильні сторони базового алгоритму та додає додатковий рівень захисту, що робить його ідеальним вибором для систем із високими вимогами до безпеки.

Основна ідея вдосконаленого HOTP полягає в динамічній зміні секретного ключа після кожної успішної сесії автентифікації. Це досягається шляхом використання спеціального алгоритму, який генерує новий секретний ключ на основі попереднього ключа, значення лічильника або додаткових параметрів, таких як часові мітки або випадкові значення. Новий ключ синхронізується між сервером і клієнтом, після чого використовується для наступної сесії. Таким чином, навіть у разі компрометації поточного ключа, зловмисник не зможе використовувати його для майбутніх автентифікацій.

Алгоритм удосконаленого HOTP працює за наступною схемою. Сервер і клієнт спільно зберігають початковий секретний ключ і лічильник. Для кожної сесії клієнт використовує поточний секретний ключ і лічильник для генерації одноразового пароля за допомогою HMAC. Після успішної автентифікації сервер і клієнт одночасно обчислюють новий секретний ключ за допомогою функції оновлення, яка може базуватися на криптографічному хешуванні. Новий ключ записується замість старого, і процес повторюється для наступної сесії [4].

Переваги вдосконаленого HOTP є значними. Завдяки динамічній зміні ключа після кожної сесії алгоритм забезпечує високу стійкість до атак. Навіть якщо зловмисник перехопить одноразовий пароль або поточний секретний ключ, він не зможе скористатися ними у майбутніх сесіях, оскільки ці дані більше не будуть дійсними.

Однак впровадження вдосконаленого HOTP супроводжується певними складнощами. Головною з них є необхідність синхронізації нового ключа між сервером і клієнтом. Якщо синхронізація буде порушена через технічні збої або втрату з'єднання, це може призвести до блокування автентифікації.

Таким чином, удосконалений HOTP є потужним інструментом для забезпечення інформаційної безпеки. Його динамічна зміна ключів значно підвищує стійкість до атак і робить систему більш захищеною, хоча й вимагає додаткових ресурсів і ретельної організації синхронізації. Це рішення є оптимальним вибором для критичних систем, де безпека даних має першочергове значення.

У таблиці 1 здійснено порівняння проаналізованих алгоритмів.

Таблиця 1 – Порівняння алгоритмів генерації динамічно змінюваних ключів

Алгоритм	Основний параметр	Стійкість до атак	Складність синхронізації	Використання в DRM-системах
HOTP	Лічильник	Висока, але залежить від синхронізації лічильника	Вимагає синхронізації лічильника між сервером і пристроєм	Підходить для одноразових сесій
TOTP	Час	Висока, але залежить від точності синхронізації часу	Висока вимога до точності синхронізації часу	Підходить для коротких сесій із обмеженим доступом
Nonce (випадкові значення)	Випадкове значення (nonce)	Висока, за умови унікальності значення	Вимагає синхронізації nonce на сервері та пристрої	Ефективний для короткочасних сесій
Удосконалений HOTP з динамічним ключем	Динамічно змінюваний секретний ключ	Дуже висока, завдяки зміні ключа після кожної сесії	Висока складність синхронізації нового ключа між сервером і пристроєм	Ідеально підходить для захисту з високими вимогами до безпеки в DRM-системах
TOTP	Час	Висока, але залежить від точності синхронізації часу	Висока вимога до точності синхронізації часу	Підходить для коротких сесій із обмеженим доступом
Nonce (випадкові значення)	Випадкове значення (nonce)	Висока, за умови унікальності значення	Вимагає синхронізації nonce на сервері та пристрої	Ефективний для короткочасних сесій

Усі розглянуті алгоритми мають свою специфічну сферу застосування та забезпечують захист за допомогою різних методів, тобто кожен з них адаптований для вирішення певної задачі.

Висновок

Аналіз алгоритмів генерації динамічно змінюваних ключів показав, що кожен із них має свої переваги та недоліки залежно від сфери застосування. HOTP забезпечує простоту реалізації та ефективність для одноразових сесій, але залежність від синхронізації лічильника може створювати ризики. TOTP, заснований на часових мітках, забезпечує високу безпеку для коротких сесій, але потребує точної синхронізації часу. Nonce гарантує унікальність і швидкість генерації ключів, однак вимагає організації синхронізації між клієнтом і сервером. Удосконалений HOTP із динамічно змінюваними ключами демонструє найвищий рівень безпеки завдяки оновленню ключа після кожної сесії, що робить його оптимальним для критичних систем із високими вимогами до захищеності. Результати дослідження підкреслюють важливість адаптації алгоритму до специфічних потреб системи для досягнення оптимального балансу між безпекою, продуктивністю та складністю реалізації.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. KIRKLAND S. Understanding One-Time Passwords (OTPs) [Електронний ресурс] / SAMANTHA KIRKLAND. – 2023. – Режим доступу до ресурсу: https://www.cybersecurityguide.org/otp-basics/.
2. Time-Based One-Time Password Algorithm [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: https://tools.ietf.org/html/rfc6238.
3. SECURITY TODAY. What is Nonce in Cryptography? [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: https://www.securitytoday.com/articles/what-is-cryptographic-nonce.
4. Dynamic Key Generation in Modern Authentication Systems [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: https://www.infosecurity-magazine.com/dynamic-key-generation-guide/.

Пуздрановський Ілля Володимирович – студент групи КІТС-23М, факультет менеджменту та інформаційної безпеки, Вінницький національний технічний університет, Вінниця, e-mail: ilia.puzdranovskuy@gmail.com

Науковий керівник: ***Салієва Ольга Володимирівна*** – доктор філософії (PhD) за спеціальністю 125 «Кібербезпека», доцент кафедри менеджменту та безпеки інформаційних систем, Вінницький національний технічний університет, Вінниця, e-mail: salieva8257@gmail.com

Puzdranovskyi Illia V. – student of KITS-23M group, Faculty of Management and Information Security, Vinnytsia National Technical University, Vinnytsia, e-mail ilia.puzdranovskuy@gmail.com

Supervisor: ***Saliieva Olha V.*** – Doctor of Philosophy (PhD) in 125 "Cybersecurity", Associate Professor of the Department of Management and Security of Information Systems, Vinnytsia National Technical University, Vinnytsia, e-mail: salieva8257@gmail.com