УДК 004.056.5

А. Р. Магденко
О. В. Салієва

АНАЛІЗ ЗАСТОСУВАННЯ ПРОТОКОЛУ WIREGUARD ДЛЯ РЕАЛІЗАЦІЇ VPN НОВОГО ПОКОЛІННЯ

Вінницький національний технічний університет

Анотація

У даній роботі досліджується еволюція протоколів віртуальних приватних мереж (VPN) та обґрунтовується необхідність переходу до рішень нового покоління на прикладі протоколу WireGuard. Проаналізовано ключові недоліки традиційних технологій тунелювання, таких як IPsec та OpenVPN, серед яких надмірна складність архітектури, великий обсяг вихідного коду, підтримка застарілих криптографічних алгоритмів та зниження продуктивності через необхідність перемикання контексту між простором користувача (user space) та простором ядра (kernel space). Як сучасну альтернативу розглянуто архітектуру WireGuard, яка базується на принципах інженерного мінімалізму та використанні виключно сучасних, високошвидкісних криптографії (ChaCha20 для симетричного шифрування, Poly1305 для аутентифікації, Curve25519 для обміну ключами). Доведено, що інтеграція протоколу безпосередньо в ядро операційної системи забезпечує значний приріст пропускної здатності та радикальне зменшення мережевих затримок. Особливу увагу приділено механізму маршрутизації на основі криптографічних ключів (Cryptokey Routing) та властивості безшовного роумінгу (roaming), що дозволяє клієнтським пристроям динамічно змінювати IP-адреси при переході між різними фізичними мережами без розриву захищеного тунелю. Зроблено висновок, що кардинальне зменшення обсягу кодової бази (до орієнтовно 4000 рядків) не лише значно спрощує процес аудиту безпеки коду, але й критично звужує поверхню атаки, роблячи WireGuard оптимальним, швидким та найбільш захищеним вибором для побудови сучасних корпоративних та приватних мереж.

Ключові слова: віртуальні приватні мережі (VPN), WireGuard, криптографія, тунелювання трафіку, кібербезпека, IPsec, OpenVPN, криптомаршрутизація (Cryptokey Routing), ядро Linux, мережевий роумінг.

Abstracts

The abstract examines the evolution of virtual private network (VPN) protocols and justifies the need to transition to next-generation solutions using the WireGuard protocol as an example. It analyses the key shortcomings of traditional tunnelling technologies, such as IPsec and OpenVPN, including excessive architectural complexity, large source code size, support for outdated cryptographic algorithms, and reduced performance due to the need to switch context between user space and kernel space . The WireGuard architecture, based on the principles of engineering minimalism and the use of exclusively modern, high-speed cryptography (ChaCha20 for symmetric encryption, Poly1305 for authentication, Curve25519 for key exchange), is considered as a modern alternative.It has been proven that integrating the protocol directly into the operating system kernel provides a significant increase in throughput and a radical reduction in network delays. Particular attention is paid to the cryptographic key-based routing mechanism (Cryptokey Routing) and the seamless roaming feature, which allows client devices to dynamically change IP addresses when switching between different physical networks without breaking the secure tunnel. It has been concluded that a radical reduction in the size of the code base (to approximately 4,000 lines) not only significantly simplifies the code security audit process, but also critically narrows the attack surface, making WireGuard the optimal, fastest, and most secure choice for building modern corporate and private networks.

Keywords: Virtual Private Networks (VPN), WireGuard, cryptography, traffic tunneling, cybersecurity, IPsec, OpenVPN, Cryptokey Routing, Linux kernel, network roaming.

Вступ

Стрімкий розвиток розподілених інформаційних мереж, перехід до концепції віддаленої роботи та експоненційне зростання кількості мобільних пристроїв вимагають створення надійних, швидких та безпечних каналів передачі даних. Традиційно для вирішення цього завдання використовуються віртуальні приватні мережі (VPN) на базі протоколів IPsec та OpenVPN. Однак ці технології, розроблені кілька десятиліть тому, архітектурно не завжди відповідають сучасним вимогам до продуктивності, мобільності та легкості аудиту. Зокрема, архітектура IPsec характеризується надмірною складністю специфікацій та громіздкою кодовою базою (сотні тисяч рядків коду), що суттєво ускладнює незалежний аудит безпеки та підвищує ймовірність наявності прихованих вразливостей. З іншого боку,

OpenVPN, хоч і є гнучким стандартом де-факто, функціонує переважно у просторі користувача (user space). Це призводить до значних накладних витрат ресурсів процесора на постійне перемикання контексту операційної системи під час шифрування трафіку та, як наслідок, до відчутного зниження пропускної здатності. Крім того, класичні VPN-протоколи важко переносять умови нестабільного зв'язку: при зміні IP-адреси клієнтського пристрою (наприклад, під час переходу від мережі Wi-Fi до стільникової мережі LTE/5G) відбувається розрив з'єднання, що вимагає повторної ініціалізації сесії та повторного узгодження ключів (handshake) [1].

Відповіддю на зазначені архітектурні виклики стала розробка протоколу WireGuard, який пропонує фундаментально новий підхід до побудови захищених мережевих тунелів. Архітектура WireGuard базується на принципах інженерного мінімалізму та глибокої інтеграції безпосередньо в мережевий стек ядра операційної системи (kernel space), що забезпечує безпрецедентну швидкість обробки пакетів. Важливою відмінністю є відмова від криптографічної гнучкості (agility), яка часто ставала причиною атак на пониження версії протоколу (downgrade attacks) у старих VPN. WireGuard використовує єдиний, фіксований набір найсучасніших криптографічних примітивів: потоковий шифр ChaCha20 для конфіденційності, алгоритм Poly1305 для автентифікації повідомлень та еліптичні криві Curve25519 для безпечного обміну ключами. Такий радикальний підхід дозволив не лише підвищити стійкість до криптоаналізу, але й зменшити обсяг вихідного коду протоколу до орієнтовно 4000 рядків. Настільки компактна кодова база робить програмний продукт придатним для повного та регулярного ручного аудиту безпеки. Крім того, інноваційна концепція маршрутизації на основі криптографічних ключів (Cryptokey Routing) забезпечує властивість безшовного роумінгу, дозволяючи підтримувати активний тунель навіть при повній зміні мережевих реквізитів вузла. Таким чином, дослідження архітектурних переваг та особливостей впровадження VPN нового покоління на базі WireGuard є надзвичайно актуальним науково-практичним завданням, що має на меті модернізацію підходів до захисту сучасних інформаційно-телекомунікаційних інфраструктур [2].

Результати дослідження

Основною ідеєю, що відрізняє WireGuard від традиційних технологій віртуальних приватних мереж (VPN), є фундаментальна зміна архітектурного підходу до тунелювання та маршрутизації мережевого трафіку.

Поглиблений аналіз технічних характеристик та архітектурних особливостей протоколу WireGuard демонструє не просто еволюційне покращення, а фундаментальний зсув у парадигмі побудови віртуальних приватних мереж, що вирішує хронічні проблеми безпеки, продуктивності та мобільності, притаманні традиційним рішенням. Першою і, мабуть, найбільш критичною перевагою з точки зору аудиту кібербезпеки є радикальний інженерний мінімалізм та безпрецедентне зменшення обсягу вихідного коду (codebase). Для порівняння, екосистема OpenVPN у поєднанні з обов'язковими криптографічними бібліотеками, такими як OpenSSL, або підсистема IPsec (зокрема її імплементація StrongSwan) налічують від чотирьохсот тисяч до понад мільйона рядків коду, що робить їхній повноцінний незалежний аудит об'єктивно неможливим завданням навіть для великих команд дослідників [1]. Натомість базова логіка WireGuard реалізована обсягом менш ніж у чотири тисячі рядків коду на мові C. Таке звуження поверхні атаки (attack surface) не лише мінімізує щільність потенційних програмних помилок (bug density), але й вперше в історії VPN-протоколів дозволило застосувати методи формальної математичної верифікації криптографічного протоколу за допомогою спеціалізованих інструментів, таких як Tamarin prover, що математично гарантує відсутність цілих класів логічних вразливостей, пов'язаних із підміною стану сесії або витоком ключів [2].

Другою фундаментальною перевагою є відмова від концепції криптографічної гнучкості (cryptographic agility) на користь використання жорстко зафіксованого набору найсучасніших криптографічних примітивів (cipher suite). Традиційні протоколи, такі як IKEv2 в IPsec або TLS в OpenVPN, на етапі встановлення з'єднання витрачають значні ресурси на узгодження (handshake) алгоритмів шифрування між клієнтом та сервером, що історично ставало вектором для атак на пониження версії (downgrade attacks), коли зловмисник, втручаючись у трафік, змушував сторони перейти на вразливі або застарілі шифри. WireGuard повністю усуває цей ризик, використовуючи безальтернативний набір на основі фреймворку Noise Protocol: потоковий шифр ChaCha20 для забезпечення конфіденційності даних, алгоритм Poly1305 для автентифікації повідомлень (MAC), еліптичні криві Curve25519 для надзвичайно швидкого та безпечного обміну ключами за протоколом Елліптичного Діффі-Геллмана (ECDH) та алгоритм BLAKE2s для криптографічного хешування. Окремої уваги заслуговує вибір шифру ChaCha20, який був спроектований спеціально для забезпечення високої швидкості програмної обробки. На відміну від стандарту AES, який вимагає

наявності спеціальних апаратних інструкцій процесора (AES-NI) для досягнення прийнятної швидкодії, ChaCha20 демонструє видатну продуктивність на мобільних архітектурах ARM, що робить WireGuard ідеальним рішенням для смартфонів та пристроїв Інтернету речей (IoT), забезпечуючи високу пропускну здатність без надмірного розряджання акумуляторних батарей [3].

Третім критичним аспектом, який забезпечує революційну продуктивність WireGuard, є його глибока інтеграція безпосередньо в мережевий стек простору ядра (kernel space) операційної системи Linux. Традиційні рішення, що працюють через віртуальні інтерфейси TUN/TAP у просторі користувача (user space), страждають від проблеми постійного контекстного перемикання (context switching). Коли мережевий пакет надходить до OpenVPN, операційна система змушена копіювати дані з простору ядра у простір користувача для шифрування, а потім повертати зашифрований пакет назад у ядро для відправки через фізичний мережевий інтерфейс. Цей процес створює колосальні накладні витрати ресурсів центрального процесора (CPU overhead) та суттєво збільшує мережеві затримки (latency). WireGuard, навпаки, здійснює всі криптографічні операції та маршрутизацію безпосередньо в ядрі, уникаючи непотрібного копіювання пам'яті. Більше того, внутрішня архітектура протоколу оптимізована для сучасних багатоядерних систем, дозволяючи ефективно розпаралелювати процес шифрування та дешифрування черг пакетів між різними ядрами процесора, що дає змогу повністю утилізувати пропускну здатність гігабітних та мультигігабітних каналів зв'язку [4].

Не менш важливою архітектурною перевагою є механізм безшовного мережевого роумінгу (roam-ability) та стійкість до мережевого сканування. Завдяки реалізації парадигми Cryptokey Routing, вузол WireGuard не покладається на збереження TCP-сесії або статичні IP-адреси. Протокол працює поверх UDP і використовує концепцію безстановісної (stateless) обробки даних, де зв'язок між пристроями визначається виключно їхніми відкритими ключами, асоційованими з переліком дозволених внутрішніх адрес (AllowedIPs). Якщо мобільний клієнт перемикається між мережами різного типу, змінюючи свою зовнішню IP-адресу, сервер автоматично оновлює інформацію про кінцеву точку одразу після отримання першого ж коректно зашифрованого та автентифікованого пакета з нової адреси. Це відбувається абсолютно прозоро для користувача та прикладного програмного забезпечення, виключаючи дратівливі розриви з'єднання та необхідність повторної ініціалізації тунелю. На додаток до цього, протокол реалізує концепцію максимальної прихованості (stealth). Перш ніж операційна система виділить хоча б байт пам'яті для обробки вхідного пакета або відповість на нього, пакет має пройти сувору криптографічну перевірку (MAC check). Будь-який неавторизований трафік, пакети з неправильними підписами або спроби сканування портів миттєво відкидаються системою (silent drop) без генерації жодних повідомлень про помилку (ICMP errors). Це робить вузли WireGuard фактично невидимими для зловмисників у глобальній мережі та забезпечує безпрецедентно високий рівень захисту від атак на виснаження ресурсів (Resource Exhaustion) та розподілених атак на відмову в обслуговуванні (DDoS), оскільки нападник не може змусити сервер витрачати обчислювальні потужності на криптографічну обробку нелегітимного сміттєвого трафіку [5].

Незважаючи на беззаперечні архітектурні переваги, визначну продуктивність та високий рівень криптографічної стійкості, об’єктивний науковий аналіз протоколу WireGuard вимагає детального розгляду його структурних недоліків та експлуатаційних обмежень, які не дозволяють вважати цю технологію універсальним рішенням для абсолютно всіх сценаріїв використання. Першим та найбільш дискусійним недоліком є фундаментальний конфлікт внутрішньої логіки маршрутизації протоколу з концепцією абсолютної приватності та політикою відмови від логування (zero-log policy), яка є критично важливою для комерційних VPN-провайдерів та правозахисних організацій. Для забезпечення механізму безшовного мережевого роумінгу (Cryptokey Routing) ядро WireGuard повинно постійно зберігати у своїй оперативній пам'яті (у спеціальних хеш-таблицях) актуальну зовнішню IP-адресу кожного підключеного клієнта, прив’язану до його відкритого ключа, причому ці дані, згідно зі специфікацією, не видаляються автоматично навіть після тривалої відсутності активності з боку вузла. Це означає, що у разі компрометації сервера зловмисники або правоохоронні органи можуть отримати доступ до реальних IP-адрес користувачів шляхом банального дампу оперативної пам’яті ядра, що змушує адміністраторів розробляти додаткові милиці у вигляді зовнішніх скриптів для примусового очищення таблиць маршрутизації після розірвання сесії, порушуючи тим самим філософію інженерного мінімалізму [1].

Другою суттєвою проблемою є повна відсутність вбудованих механізмів обфускації (маскування) трафіку. Оскільки WireGuard працює виключно поверх протоколу UDP, має строго визначену послідовність криптографічного узгодження (handshake) та фіксовані розміри ініціалізаційних пакетів, сучасні системи глибокого інспектування пакетів (Deep Packet Inspection), які використовуються в країнах із жорсткою інтернет-цензурою або в суворих корпоративних мережах, здатні миттєво

ідентифікувати специфічні патерни (сигнатури) цього протоколу та заблокувати з'єднання. На відміну від OpenVPN, який підтримує маршрутизацію через TCP-порт 443 і може імітувати звичайний HTTPS-трафік за допомогою додаткових плагінів (наприклад, Stunnel або obfs4), WireGuard не має механізму резервного перемикання на протокол TCP (TCP fallback). Це робить його абсолютно неробочим у рестриктивних мережах, наприклад, у публічних Wi-Fi зонах готелів чи аеропортів, де адміністратори з міркувань безпеки часто блокують будь-який вихідний UDP-трафік, за винятком запитів до DNS-серверів [6].

Третім вагомим обмеженням, яке гальмує впровадження протоколу у великому корпоративному секторі (Enterprise), є відсутність вбудованих інструментів для динамічного виділення IP-адрес та централізованого управління криптографічними ключами. Архітектура WireGuard передбачає виключно статичну конфігурацію: мережевий адміністратор змушений вручну або за допомогою сторонніх систем автоматизації генерувати пари ключів для кожного нового пристрою, жорстко прив’язувати їх до унікальних внутрішніх IP-адрес (AllowedIPs) та безпечно поширювати ці конфігураційні файли користувачам через зовнішні (out-of-band) канали зв'язку. Якщо для малого бізнесу або приватної мережі на кілька десятків пристроїв це не становить проблеми, то масштабування такої інфраструктури до тисяч співробітників перетворюється на серйозний адміністративний виклик, що вимагає розгортання додаткових надбудов, таких як Tailscale або Netmaker, які беруть на себе функції оркестрації ключів та динамічної маршрутизації, але водночас нівелюють первинну ідею використання чистого, легковагового та незалежного VPN-тунелю [7].

Висновки

Підсумовуючи результати проведеного дослідження, можна з упевненістю стверджувати, що протокол WireGuard є не просто черговим еволюційним кроком у розвитку технологій віртуальних приватних мереж, а справжнім парадигмальним зсувом, який фундаментально переосмислює архітектуру захищеного мережевого тунелювання. Завдяки відмові від застарілих та громіздких стандартів на користь інженерного мінімалізму, глибокій інтеграції безпосередньо в мережевий стек простору ядра (kernel space) операційної системи Linux та використанню виключно сучасних, математично доведених криптографічних примітивів, таких як потоковий шифр ChaCha20 та алгоритм обміну ключами Curve25519, розробникам вдалося вирішити багаторічний компроміс між високим рівнем безпеки та пропускною здатністю. Зменшення кодової бази до кількох тисяч рядків не лише мінімізувало поверхню атаки, але й вперше дозволило повноцінно застосувати методи формальної математичної верифікації для підтвердження надійності протоколу. Хоча об’єктивний аналіз виявив низку експлуатаційних обмежень, зокрема вразливість до систем глибокого інспектування пакетів (DPI) через відсутність вбудованої обфускації UDP-трафіку та складнощі з централізованим масштабуванням у великих корпоративних мережах без залучення сторонніх оркестраторів, ці недоліки не нівелюють загальної цінності технології, а лише чітко окреслюють її оптимальну нішу. Зокрема, унікальні архітектурні особливості WireGuard, такі як парадигма криптографічної маршрутизації (Cryptokey Routing), властивість безшовного мережевого роумінгу при зміні точок доступу та механізм максимальної прихованості (stealth), що передбачає ігнорування будь-яких неавторизованих мережевих запитів, роблять його ідеальним інструментом для побудови високонадійних, стійких до сканування ізольованих інфраструктур. На практиці розгортання таких захищених середовищ є критично необхідною умовою для безпечної роботи фахівців із кібербезпеки під час проведення комплексних заходів розвідки на основі відкритих джерел (OSINT), безпечного збору та парсингу цифрових ідентифікаторів усередині закритих екосистем месенджерів, таких як Telegram, або під час здійснення глибокого кореляційного аналізу тіньових фінансових потоків та інфраструктури DarkWeb-маркетплейсів. У подібних сценаріях високого ризику, де будь-який розрив з'єднання або витік реальної IP-адреси дослідника може призвести до деанонімізації та повної компрометації аналітичного процесу, стабільність та скритність WireGuard відіграють вирішальну роль. Таким чином, впровадження протоколу WireGuard суттєво підвищує мобільність, енергоефективність та загальну криптографічну стійкість захищених каналів зв'язку, що дає всі підстави вважати його новим стандартом де-факто для побудови віртуальних приватних мереж як у сегменті індивідуального використання, так і для забезпечення захисту спеціалізованих аналітичних процесів у сучасній кібербезпеці.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Аналіз VPN технологій. URL: https://www.researchgate.net/publication/393410400_A_Performance_Analysis_of_VPN_Technologies_Used_in_an_IoT_Environment (дата звернення: 01.03.2026).

2. Протокол WireGuard. URL: https://www.ndss-symposium.org/wp-content/uploads/2017/09/ndss2017_04A-3_Donenfeld_paper.pdf (дата звернення: 01.03.2026).

3. ChaCha20. URL: https://xilinx.github.io/Vitis_Libraries/security/2019.2/guide_L1/internals/chacha20.html (дата звернення: 02.03.2026).

4. WireGuard in the Linux. URL: https://eprint.iacr.org/2019/482.pdf (дата звернення: 03.03.2026).

5. Безпека протоколу. URL: https://www.researchgate.net/publication/335351986_A_Mechanised_Cryptographic_Proof_of_the_WireGuard_Virtual_Private_Network_Protocol (дата звернення: 03.03.2026).

6. Аналіз трафіку. URL: https://www.semanticscholar.org/paper/Towards-a-Comprehensive-Picture-of-the-Great-DNS/7ad11ecb66a12f9084a08e15d5b8a0d36cd78b40 (дата звернення: 04.03.2026).

7. Enterprise WireGuard Management with Netmaker. Netmaker Official Documentation. URL: https://docs.netmaker.io/docs/about (дата звернення: 04.03.2026).

Магденко Анастасія Романівна **–** студентка групи 1КІТС-22б, Факультет менеджменту та інформаційної безпеки, Вінницький національний технічний університет, м. Вінниця, e-mail: anastasiiamahdenko@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@vntu.edu.ua

Mahdenko Anastasiia R. – student of group 1KITS-22b, Faculty of Management and Information Security, Vinnytsia National Technical University, Vinnytsia, e-mail: anastasiiamahdenko@gmail.com

Supervisor: ***Saliieva Olha V.*** – Doctor of Philosophy (PhD) in specialty 125 "Cybersecurity", Associate Professor of the Department of Management and Security of Information Systems, Vinnytsia National Technical University, Vinnytsia, email: salieva8257@ vntu.edu.ua