ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ТЕЛЕКОМУНІКАЦІЙ
КАФЕДРА ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМ ТА МЕРЕЖ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «МЕТОДИ ЗАХИСТУ ПЕРСОНАЛЬНИХ ДАНИХ У ТЕЛЕКОМУНІКАЦІЙНИХ МЕРЕЖАХ»

на здобуття освітнього ступеня магістра
зі спеціальності 172 Електронні комунікації та радіотехніка
(код, найменування спеціальності)
освітньо-професійної програми Телекомунікаційні системи та мережі
(назва)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

_____ Роман КАШПУР
(підпис) *(Ім'я ПРІЗВИЩЕ)*

Виконав: здобувач вищої освіти гр. ТСДМ-62
Роман КАШПУР
Ім'я, ПРІЗВИЩЕ

Керівник: Юрій МЕЛЬНИК
д.т.н., професор *Ім'я, ПРІЗВИЩЕ*

Рецензент: _____
Ім'я, ПРІЗВИЩЕ

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
Навчально-науковий інститут Телекомунікацій

Кафедра ___Телекомунікаційних систем та мереж_____
Ступінь вищої освіти __магістр_____
Спеціальність ___172 Електронні комунікації та радіотехніка _____
Освітньо-професійна програма _____Телекомунікаційні системи та мережі_____

ЗАТВЕРДЖУЮ
Завідувач кафедри __ТСМ_____
_____ Віктор ЗАЇКА
«____» _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Кашпуру Роману Олександровичу
(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Методи захисту персональних даних у телекомунікаційних мережах
керівник кваліфікаційної роботи Юрій МЕЛЬНИК, д.т.н., професор
(Ім'я, ПРІЗВИЩЕ науковий ступінь, вчене звання)
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «30» 10. 2025 р. № 467

2. Строк подання кваліфікаційної роботи «___» _____ 2025 р.

3. Вихідні дані до кваліфікаційної роботи:
1. Персональні дані.
2. Симетричний та асиметричний захист від DDoS.
3. Методи захисту.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Безпека в базах даних.
2. Симетричний та асиметричний захист від DDoS.
3. Технологія захисту SRAM PUF.
4. Захист на рівні Backend.

5. Перелік ілюстративного матеріалу: ___презентація на ____слайдах_____

6. Дата видачі завдання «___» _____ 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Підбір та аналіз науково-технічної літератури	15.10.2025	Викон.
2	Розробка першого розділу	23.10.2025	Викон.
3	Розробка другого розділу	03.11.2025	Викон.
4	Розробка третього розділу	13.11.2025	Викон.
5	Розробка четвертого розділу	23.11.2025	Викон.
6	Висновки, вступ, реферат	03.12.2025	Викон.
7	Розробка презентації	13.12.2025	Викон.

Здобувач вищої освіти _____ Роман КАШПУР
(підпис) *(Ім'я ПРІЗВИЩЕ)*

Керівник кваліфікаційної роботи _____ Юрій МЕЛЬНИК
(підпис) *(Ім'я ПРІЗВИЩЕ)*

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
Навчально-науковий інститут телекомунікацій
ПОДАННЯ
ГОЛОВІ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ
ЩОДО ЗАХИСТУ КВАЛІФІКАЦІЙНОЇ РОБОТИ
на здобуття освітнього ступеня магістра

Направляється здобувач Кашпур Р.О. до захисту кваліфікаційної роботи
(прізвище та ініціали)
за спеціальністю 172 Електронні комунікації та радіотехніка
(код, найменування спеціальності)
освітньо-професійної програми Телекомунікаційні системи та мережі
(назва)
на тему: «Методи захисту персональних даних у телекомунікаційних мережах»

Кваліфікаційна робота і рецензія додаються.

Директор ННІТ _____ Владислав КРАВЧЕНКО
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Висновок керівника кваліфікаційної роботи

Здобувач Кашпур Роман Олександрович показав гарну теоретичну та інженерну підготовку, уміння володіти новими комп'ютерними технологіями, користуватися навчальною, довідковою і науково-технічною літературою. Працюючи над завданнями, які були поставлені керівником, проявив ініціативність, сумлінність та хист до наукової роботи.

Все це дозволяє оцінити виконану кваліфікаційну роботу здобувача Кашпура Романа Олександровича на оцінку «відмінно» та присвоїти йому кваліфікацію магістр з електронних комунікацій та радіотехніки.

Керівник кваліфікаційної роботи _____ Юрій МЕЛЬНИК
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

«____» _____ 20___ року

Висновок кафедри про кваліфікаційну роботу

Кваліфікаційна робота розглянута. Здобувач Кашпур Р.О. допускається до захисту даної роботи в Екзаменаційній комісії.

Завідувач кафедри телекомунікаційних систем та мереж _____ Віктор ЗАЇКА
(назва) *(підпис)* *(Ім'я, ПРІЗВИЩЕ)*

«____» _____ 20___ року

ВІДГУК РЕЦЕНЗЕНТА
на кваліфікаційну роботу
на здобуття освітнього ступеня магістра

здобувача вищої освіти Кашпура Романа Олександровича
(прізвище, ім'я, по батькові)
на тему «Методи захисту персональних даних у телекомунікаційних мережах»

Актуальність.
У сучасних умовах глобальної цифровізації питання забезпечення кібербезпеки та захисту персональних даних є одним із найпріоритетніших напрямів розвитку телекомунікаційних систем. Масове впровадження технологій 5G/6G, хмарних сервісів та інтернету речей (IoT) призводить до зростання обсягів оброблюваної інформації та підвищення ризику її компрометації.
Тема роботи є надзвичайно актуальною, оскільки охоплює як захист баз даних і веб-додатків від типових атак (SQL-ін'єкцій, несанкціонованого доступу, зламу паролів), так і сучасні методи протидії DDoS-атакам у симетричних та асиметричних мережах. Це безпосередньо відповідає сучасним викликам у сфері кіберзахисту телекомунікаційних інфраструктур.
Позитивні сторони.
1. У роботі розглянуто реальні приклади кібератак (Equifax, LinkedIn), подано технічні рішення для запобігання інцидентам.
2. Особливу увагу було приділено вивченню таких типових атак, як SQL-ін'єкції, XSS, переповнення буфера, CSRF, DDoS.
Недоліки.
1. У роботі не розглянуто нормативно-правовий аспект захисту персональних даних.
2. У роботі було б доцільно додати кількісну оцінку ефективності методів захисту.

Відзначені зауваження не впливають на загальну позитивну оцінку магістерської кваліфікаційної роботи

Висновок: *кваліфікаційна робота на здобуття ступеня магістра заслуговує оцінку "відмінно", а здобувач Кашпур Роман Олександрович заслуговує присвоєння кваліфікації: магістр з електронних комунікацій та радіотехніки.*

Рецензент _____ _____
науковий ступінь, вчене звання (підпис) *(Ім'я ПРІЗВИЩЕ)*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 79 стор., 25 рис., 24 джерела.

Мета роботи – дослідження сучасних методів захисту персональних даних у телекомунікаційних мережах та аналіз ефективності симетричних і асиметричних підходів до побудови систем протидії DDoS-атакам, з подальшим формуванням рекомендацій щодо підвищення рівня безпеки інформаційних систем.

Об’єкт дослідження – процеси забезпечення інформаційної безпеки у телекомунікаційних мережах, зокрема захист персональних даних і стабільності роботи мережевої інфраструктури в умовах кіберзагроз.

Предмет дослідження – методи, алгоритми та технології захисту даних і мережевих ресурсів — зокрема, засоби автентифікації, авторизації, шифрування, а також механізми симетричного та асиметричного DDoS-захисту.

Короткий зміст роботи: у кваліфікаційній роботі проаналізовано сучасний стан інформаційної безпеки та основні кіберзагрози. Розглянуто типові атаки — SQL-ін’єкції, XSS, CSRF, DDoS, а також ефективні методи протидії: параметризовані запити, фільтрація введення, SSL/TLS, OAuth, OpenID і політика найменших привілеїв. Підкреслено важливість організаційних заходів — підготовки персоналу, аудиту безпеки, оновлення ПЗ та DevSecOps-підходу. Зроблено висновок, що надійний захист персональних даних можливий лише за умови комплексного підходу, який поєднує технічні, криптографічні та організаційні засоби.

КЛЮЧОВІ СЛОВА: ЗАХИСТ ПЕРСОНАЛЬНИХ ДАНИХ, ТЕЛЕКОМУНІКАЦІЙНІ МЕРЕЖІ, ІНФОРМАЦІЙНА БЕЗПЕКА, SQL-ІН’ЄКЦІЇ, XSS, CSRF, DDOS, SSL/TLS, OAUTH, OPENID, АВТЕНТИФІКАЦІЯ, АВТОРИЗАЦІЯ, ШИФРУВАННЯ, DEVSECOPS, ZERO TRUST, ШТУЧНИЙ ІНТЕЛЕКТ

ABSTRACT

Text part of the master's qualification work: 79 pages, 25 figures, 24 sources.

The purpose of the work is to study modern methods of protecting personal data in telecommunication networks and analyze the effectiveness of symmetric and asymmetric approaches to building systems to counter DDoS attacks, with the subsequent formation of recommendations for increasing the level of security of information systems.

Object of the research is the processes of ensuring information security in telecommunication networks, in particular, the protection of personal data and the stability of the network infrastructure in the face of cyber threats.

Subject of the research is methods, algorithms and technologies for protecting data and network resources - in particular, means of authentication, authorization, encryption, as well as mechanisms for symmetric and asymmetric DDoS protection.

Summary of the work: the qualification work analyzes the current state of information security and the main cyber threats. Typical attacks are considered - SQL injections, XSS, CSRF, DDoS, as well as effective countermeasures: parameterized queries, input filtering, SSL/TLS, OAuth, OpenID and the policy of least privilege. The importance of organizational measures is emphasized - personnel training, security audits, software updates and DevSecOps approach. It is concluded that reliable protection of personal data is possible only with a comprehensive approach that combines technical, cryptographic and organizational means.

KEYWORDS: PERSONAL DATA PROTECTION, TELECOMMUNICATION NETWORKS, INFORMATION SECURITY, SQL INJECTION, XSS, CSRF, DDOS, SSL/TLS, OAUTH, OPENID, AUTHENTICATION, AUTHORIZATION, ENCRYPTION, DEVSECOPS, ZERO TRUST, ARTIFICIAL INTELLIGENCE

ЗМІСТ

ВСТУП 9
РОЗДІЛ 1 БЕЗПЕКА В БАЗАХ ДАНИХ 11
1.1 Основні загрози 11
1.2 Методи автентифікації 14
1.3 Авторизація та права доступу 21
РОЗДІЛ 2 СИМЕТРИЧНИЙ ТА АСИМЕТРИЧНИЙ ЗАХИСТ ВІД DDOS 25
2.1 Поняття симетрії в телекомунікаціях 25
2.2 Видідення потоку та захист асиметричних даних 29
2.3 Використання фільтруючого обладнання 39
РОЗДІЛ 3 ТЕХНОЛОГІЯ ЗАХИСТУ SRAM PUF 47
3.1 Технічні поняття технології 47
3.2 Робота з ключами користувачів 56
3.3 Реалізація SRAM PUF 60
РОЗДІЛ 4 ЗАХИСТ НА РІВНІ BACKEND 67
4.1 Проблематика на рівні Backend 67
4.2 Основні загрози 69
4.3 Можливі рішення 72
ВИСНОВКИ 78
ПЕРЕЛІК ПОСИЛАНЬ 80
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ 83

ВСТУП

Сучасний етап розвитку інформаційного суспільства характеризується стрімким зростанням обсягів переданих даних, глобальною цифровізацією та масовим використанням телекомунікаційних мереж. Мобільні технології, хмарні обчислення, сервіси електронної комерції, соціальні мережі, а також поява концепцій 5G/6G та інтернету речей (IoT) значно розширили спектр можливостей для бізнесу, державних структур та кінцевих користувачів. Водночас із цими перевагами виникають нові виклики, пов’язані із забезпеченням безпеки персональних даних, що дедалі частіше стають об’єктом атак і зловживань.

Персональні дані є однією з найцінніших категорій інформації, оскільки вони можуть бути використані як для легітимної діяльності, так і для злочинних дій. Несанкціонований доступ до таких даних, їх модифікація чи знищення створюють реальні загрози для користувачів та організацій, включаючи фінансові втрати, підрив репутації та втрату довіри. Саме тому питання захисту персональних даних у телекомунікаційних мережах має стратегічне значення як на рівні національної, так і корпоративної безпеки.

У сучасних умовах основними загрозами для телекомунікаційних систем залишаються атаки на веб-додатки, мережеву інфраструктуру та сервери зберігання даних. Серед найбільш поширених — SQL-ін’єкції, XSS, CSRF, переповнення буфера, DDoS-атаки та несанкціоновані підключення до баз даних. Ці методи дозволяють зловмисникам викрадати або змінювати дані, отримувати контроль над обліковими записами та навіть повний доступ до систем. Тому розробка ефективних механізмів захисту є першочерговим завданням для фахівців у сфері кібербезпеки та телекомунікацій.

Захист персональних даних у телекомунікаційних мережах потребує багаторівневого підходу, що включає технічні, організаційні та правові заходи. Серед технічних методів найбільш актуальними є використання параметризованих запитів, фільтрації й валідації вхідних даних, криптографічних протоколів SSL/TLS, механізмів багатофакторної аутентифікації, управління доступом на

основі принципу найменших привілеїв, систем моніторингу та виявлення вторгнень (IDS/IPS), а також регулярне оновлення програмного забезпечення. Важливу роль відіграють також сучасні підходи на кшталт DevSecOps та архітектури Zero Trust, які інтегрують безпеку в усі процеси розробки та експлуатації систем.

Мета цієї роботи полягає у дослідженні та аналізі сучасних методів захисту персональних даних у телекомунікаційних мережах, вивченні найбільш поширених загроз та розробці рекомендацій для підвищення рівня безпеки інформаційних систем. Для досягнення цієї мети необхідно вирішити низку завдань:

- проаналізувати існуючі типи атак і методи їх реалізації;
- дослідити наявні засоби захисту від найпоширеніших загроз;
- оцінити ефективність різних механізмів безпеки;
- сформувати рекомендації та найкращі практики для розробників і адміністраторів веб-додатків і телекомунікаційних систем.

Таким чином, актуальність теми зумовлена зростанням масштабів кібератак, цінністю персональних даних у сучасному світі та необхідністю формування комплексного підходу до їх захисту в телекомунікаційних мережах.

1 БЕЗПЕКА В БАЗАХ ДАНИХ

1.1 Основні загрози

SQL-ін’єкції, злам паролів та несанкціонований доступ до даних сьогодні становлять один із найсерйозніших викликів для кібербезпеки, адже вони безпосередньо впливають на захищеність баз даних — критичного елементу будь-якої інформаційної системи. Сучасні телекомунікаційні та корпоративні мережі, системи електронної комерції, державні реєстри чи навіть освітні онлайн-платформи — усі вони ґрунтуються на обробці, збереженні та передачі даних, що робить бази даних однією з головних цілей атак.

SQL-ін’єкції як універсальний інструмент атаки

SQL-ін’єкція є технікою, за допомогою якої зловмисник отримує змогу безпосередньо втручатися у логіку запитів до бази даних. Класичний сценарій виглядає так: користувач вводить у вебформу дані — логін, пароль, пошуковий запит чи коментар. Програма повинна перевірити ці дані та сформувати коректний SQL-запит. Проте якщо розробники не передбачили захисту (наприклад, не використовували підготовлені вирази чи не здійснили фільтрацію введення), зловмисник може ввести фрагмент SQL-коду, який змінить структуру запиту. У результаті він отримає доступ до значно ширшого кола даних, ніж передбачено, або навіть до самої бази даних у повному обсязі.

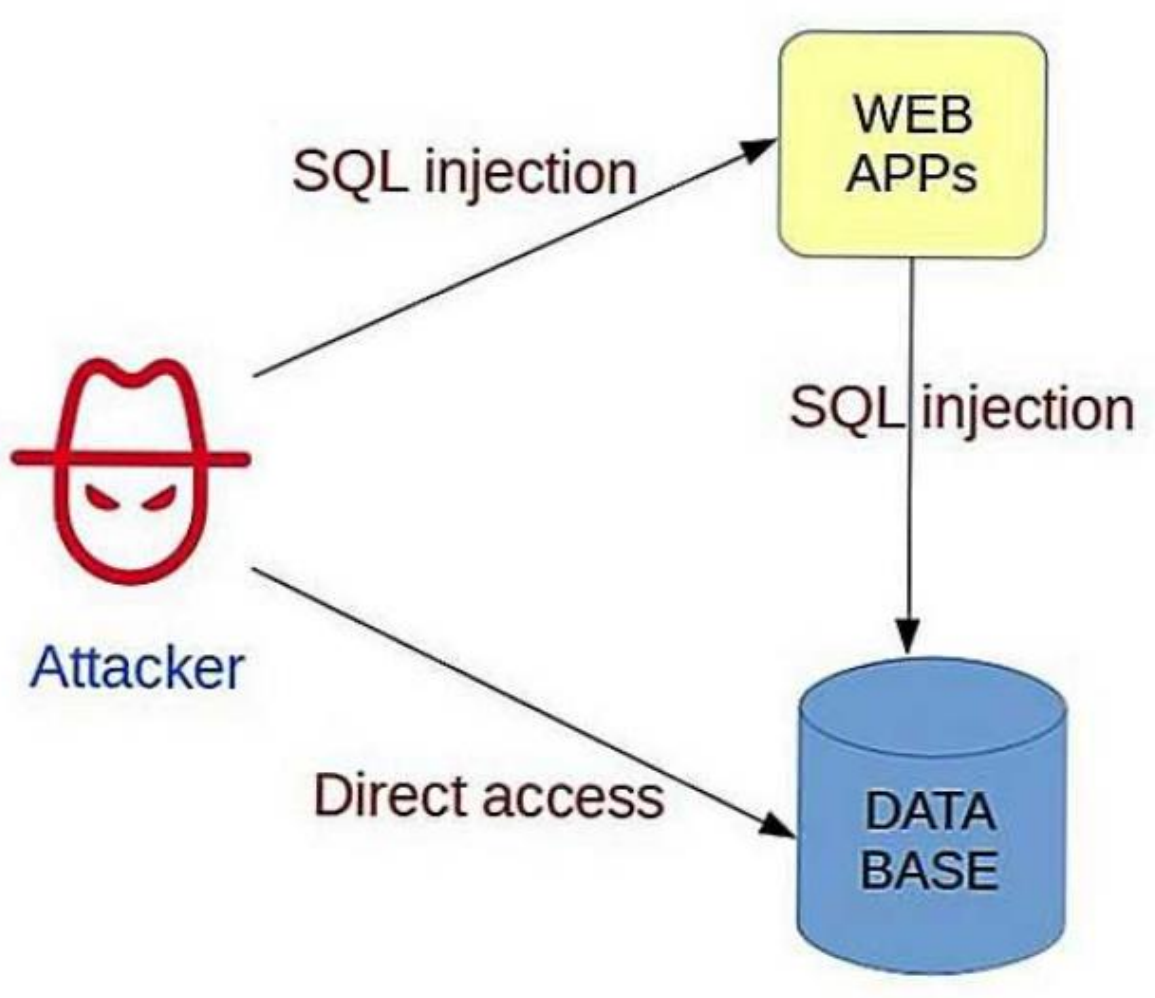


Рис. 1.1 Ілюстрація роботи SQL-ін'єкції

Зловмисник може, наприклад, сформувати запит виду:

' OR '1'='1

Цей простий рядок змінює логіку автентифікації таким чином, що система сприймає умову як істинну для будь-якого введення, і користувач отримує доступ без справжнього пароля. Більш витончені варіанти SQL-ін’єкцій дають змогу отримати повний список таблиць, копіювати вміст бази або навіть змінювати її структуру. У корпоративних системах наслідки можуть бути катастрофічними: втрата клієнтських даних, фінансових транзакцій, комерційних таємниць, а іноді й компрометація систем управління підприємством.

Автоматизація зламу паролів

Другим вразливим напрямом є паролі. Злам паролів має довгу історію і продовжує залишатися ефективним через людський фактор: користувачі часто обирають надто прості комбінації, повторно використовують ті самі паролі на

різних сервісах, не змінюють їх роками. Зловмисники активно користуються цими слабкостями, застосовуючи як простий перебір варіантів, так і цілі бази викрадених паролів, отриманих у результаті попередніх атак.

Автоматизовані програми брутфорсу здатні виконувати мільйони спроб підбору на секунду, особливо якщо парольний захист не доповнено додатковими бар'єрами. У багатьох випадках зловмисники застосовують словникові атаки — використання готових списків найпоширеніших паролів. За статистикою, десятки відсотків користувачів обирають саме ті комбінації, які входять до таких списків, що значно спрощує злам. Якщо ж база даних містить критично важливі відомості (наприклад, банківські транзакції чи персональні дані), успішний підбір пароля фактично означає повну компрометацію системи.

Несанкціонований доступ і його наслідки

Несанкціонований доступ до даних відрізняється від попередніх видів атак тим, що він часто ґрунтується на використанні вразливостей у самій архітектурі безпеки системи. Це може бути відсутність багатофакторної автентифікації, використання застарілих протоколів передачі даних, неправильне налаштування доступу або відсутність сегментації мережі. У таких випадках зловмисник може отримати доступ до бази даних не лише ззовні, але й із внутрішньої мережі організації, що робить атаку ще небезпечнішою, адже внутрішнім користувачам часто довіряють більше прав.

Прикладом може бути ситуація, коли адміністратор надає широкі повноваження усім користувачам бази даних, не розділяючи їх за рівнями доступу. У результаті компрометація навіть одного облікового запису дозволяє зловмиснику дістатися до конфіденційної інформації. Несанкціонований доступ небезпечний тим, що він може довго залишатися непоміченим: зловмисники часто «сидять» у системі тижнями чи навіть місяцями, непомітно копіюючи інформацію, що призводить до витоку даних глобального масштабу.

Масштабні інциденти як уроки для світу

В історії телекомунікацій відомо чимало прикладів таких атак. Найгучнішим став інцидент із компанією **Equifax у 2017 році**, коли було викрадено персональні

дані понад 143 мільйонів осіб. Зловмисники використали вразливість у програмному забезпеченні та слабкі механізми захисту, щоб отримати доступ до чутливих відомостей: номери соціального страхування, дати народження, адреси, фінансові дані. Це стало справжнім потрясінням для ринку, адже Equifax була однією з провідних компаній у сфері кредитних звітів. Наслідки інциденту виявилися колосальними: компанія втратила репутацію, зазнала величезних фінансових збитків і була змушена виплачувати мільярдні компенсації.

Подібні інциденти демонструють, що проблема безпеки баз даних не є локальною чи технічною — вона має глобальний соціально-економічний вимір. Витоки інформації підривають довіру користувачів до цифрових сервісів, шкодять репутації компаній та державних інституцій, створюють загрозу для приватності мільйонів людей.

Таким чином, SQL-ін’єкції, атаки на паролі та несанкціонований доступ є не лише технічними викликами, а й факторами, що визначають стійкість сучасного інформаційного суспільства. Вони показують, наскільки важливою є багаторівнева стратегія безпеки, що включає превентивні заходи, навчання користувачів, постійний моніторинг та своєчасне оновлення систем. Без цього будь-яка інформаційна інфраструктура залишається вразливою перед обличчям організованих кіберзлочинців.

1.2 Методи автентифікації

Хешування паролів є одним із ключових механізмів у сфері інформаційної безпеки, що використовується практично у всіх сучасних системах автентифікації. Основна його мета полягає в тому, щоб захистити конфіденційні дані користувачів від прямого доступу навіть у випадках, коли база даних потрапляє до рук зловмисників. Якщо пароль зберігати у відкритому вигляді (plain text), то будь-який несанкціонований доступ одразу розкриває всю інформацію. Натомість хешування перетворює пароль за допомогою спеціальних математичних алгоритмів у унікальне числове значення — хеш. Це значення є незворотним: воно не дозволяє

відновити оригінальний пароль безпосередньо, а лише слугує цифровим відбитком для подальшого порівняння.

Механізм роботи хешування

Принцип роботи досить простий, але надзвичайно ефективний. Коли користувач створює обліковий запис і задає пароль, система одразу обчислює його хеш і зберігає лише його, а не сам пароль. Під час входу користувач вводить пароль ще раз, система обчислює хеш цього введення і порівнює з тим, що вже збережений у базі. Якщо обидва значення співпадають, доступ дозволяється. Якщо ж ні — автентифікація вважається невдалою.

Завдяки такому підходу навіть у випадку витоку бази даних зловмисники отримують не паролі у відкритому вигляді, а лише їхні хеші. І хоча існують спеціальні методи атак (наприклад, атаки словниками чи з використанням rainbow tables), складність сучасних алгоритмів хешування значно ускладнює процес відновлення оригінальних паролів.

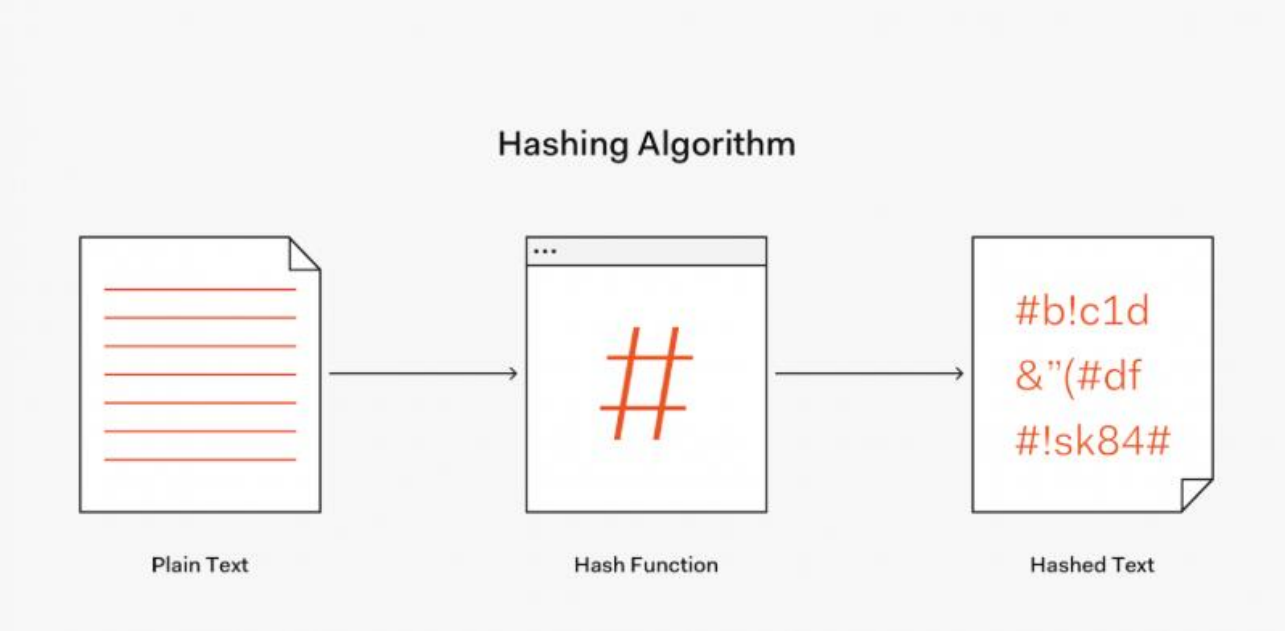


Рис. 1.2 Процес хешування

Безпечні й небезпечні алгоритми

Не всі алгоритми однаково ефективні. Наприклад, такі алгоритми, як MD5 або SHA-1, у минулому широко використовувались, але сьогодні вважаються небезпечними через наявність відомих колізій і високу швидкість обчислення, яка грає на руку зловмисникам при brute force-атаках. Для підвищення рівня безпеки у

сучасних системах застосовуються спеціалізовані алгоритми, зокрема bcrypt, scrypt та Argon2. Вони відрізняються тим, що включають у процес хешування елементи уповільнення (штучне збільшення кількості обчислень), що робить масовий перебір паролів вкрай дорогим за часом і ресурсами.

Ще одним важливим компонентом є використання солевих значень (salt) — випадкових рядків, які додаються до пароля перед обчисленням хешу. Це робить неможливим застосування попередньо підготовлених таблиць (rainbow tables), адже однакові паролі у різних користувачів матимуть різні хеші.

Роль SSL/TLS-шифрування

Проте навіть найсильніші алгоритми хешування не гарантують повної безпеки, якщо пароль передається мережею у відкритому вигляді. У таких випадках зловмисники можуть перехопити його за допомогою методів аналізу мережевого трафіку (наприклад, через атаки типу Man-in-the-Middle). Саме тому використання SSL/TLS-шифрування є ще одним базовим елементом сучасної системи захисту даних.

SSL/TLS забезпечує захищений канал зв’язку між клієнтом (браузером користувача) та сервером. Його робота ґрунтується на криптографічних методах із використанням асиметричних і симетричних ключів.

Як це працює на прикладі відвідування вебсайту:

1. Користувач уводить адресу сайту в браузері.
2. Браузер надсилає запит на з’єднання, а сервер у відповідь відправляє свій цифровий сертифікат із публічним ключем.
3. Браузер перевіряє цей сертифікат, порівнюючи його із записами центрів сертифікації, щоб упевнитися у справжності сайту.
4. Якщо сертифікат дійсний, браузер створює випадковий сесійний ключ шифрування й передає його серверу у захищеній формі.
5. Весь подальший обмін даними відбувається вже з використанням цього сесійного ключа. Це означає, що навіть у разі перехоплення трафіку сторонньою особою інформація залишиться зашифрованою й недоступною для розшифрування.

6. Коли сеанс завершено, сесійний ключ видаляється і не може бути використаний повторно.

Приклади наслідків відсутності хешування та шифрування

Історія кібербезпеки знає безліч інцидентів, коли саме відсутність сучасних методів хешування чи шифрування призводила до масштабних витоків даних. Наприклад, якщо компанія зберігає паролі у відкритому вигляді, витік навіть невеликої частини бази користувачів одразу призводить до масових зламів, адже більшість людей використовують одні й ті ж паролі для кількох сервісів. Аналогічно, якщо сайт не використовує HTTPS (який базується на TLS), будь-який перехоплювач трафіку може зчитувати паролі, логіни й особисту інформацію користувача.

Одним із відомих випадків була атака на компанію LinkedIn у 2012 році, коли було викрадено мільйони паролів користувачів через те, що вони зберігались із використанням слабкого алгоритму SHA-1 без додаткового соління. У результаті величезна кількість облікових записів була скомпрометована, а сама компанія зазнала серйозних репутаційних втрат.

Сучасні підходи до захисту

Сьогодні захист автентифікаційних даних будується на багаторівневій основі:

- Використання стійких алгоритмів хешування (bcrypt, scrypt, Argon2).
- Обов’язкове додавання salt до кожного пароля.
- Використання SSL/TLS для шифрування всього мережевого трафіку.
- Запровадження багатофакторної автентифікації (MFA), яка додає додатковий рівень перевірки (наприклад, SMS-код або мобільний додаток).
- Регулярне оновлення ключів і сертифікатів, застосування технології Forward Secrecy, що забезпечує нові ключі для кожної сесії.

Таким чином, поєднання надійного хешування паролів та шифрування трафіку за допомогою SSL/TLS є фундаментом сучасних систем захисту даних. Без цих елементів будь-яка інформаційна система залишається вкрай вразливою для зловмисників.

SSL/TLS-шифрування є одним із найважливіших інструментів захисту даних у глобальній мережі. Воно створює захищений канал комунікації між клієнтом (користувачем) та сервером, унеможливлюючи перехоплення або підміну інформації сторонніми особами. У сучасних умовах, коли кількість кібератак зростає, а їхні методи стають дедалі витонченішими, роль SSL/TLS виходить далеко за межі формального стандарту — цей механізм є основою довіри до цифрових сервісів.

Кожна передача даних через Інтернет без шифрування наражається на серйозну небезпеку. Зловмисники можуть перехоплювати незахищений трафік, зчитувати логіни та паролі, отримувати доступ до фінансових транзакцій чи персональної інформації користувачів. Використання SSL/TLS дозволяє мінімізувати ці ризики: дані кодуються спеціальними алгоритмами та стають недоступними для розшифрування без відповідних ключів. Навіть якщо пакети інформації будуть перехоплені, їхній вміст залишиться нерозгаданим.

Особливо важливим є впровадження SSL/TLS-шифрування у сферах, де відбувається обробка конфіденційних даних: в електронній комерції, онлайн-банкінгу, медичних системах, державних реєстрах чи корпоративних базах даних. Саме тому провідні міжнародні організації та стандартизуючі органи наголошують на обов’язковості використання захищених протоколів у будь-яких вебсервісах.

Розробники програмного забезпечення й адміністратори баз даних мають відповідати за впровадження SSL/TLS у всіх процесах взаємодії між клієнтом і сервером. Це означає, що будь-яка операція — від простої авторизації користувача до передачі великих масивів інформації — повинна відбуватися виключно в межах зашифрованого каналу. Такий підхід не лише гарантує максимальний рівень безпеки, а й формує довіру користувачів до сервісу, що є критично важливим у цифрову епоху.

Зрештою, SSL/TLS-шифрування слід розглядати не як додаткову опцію чи рекомендацію, а як невід’ємний стандарт роботи будь-якої сучасної інформаційної системи. Воно забезпечує конфіденційність, цілісність і автентичність даних, що є базовими принципами інформаційної безпеки.

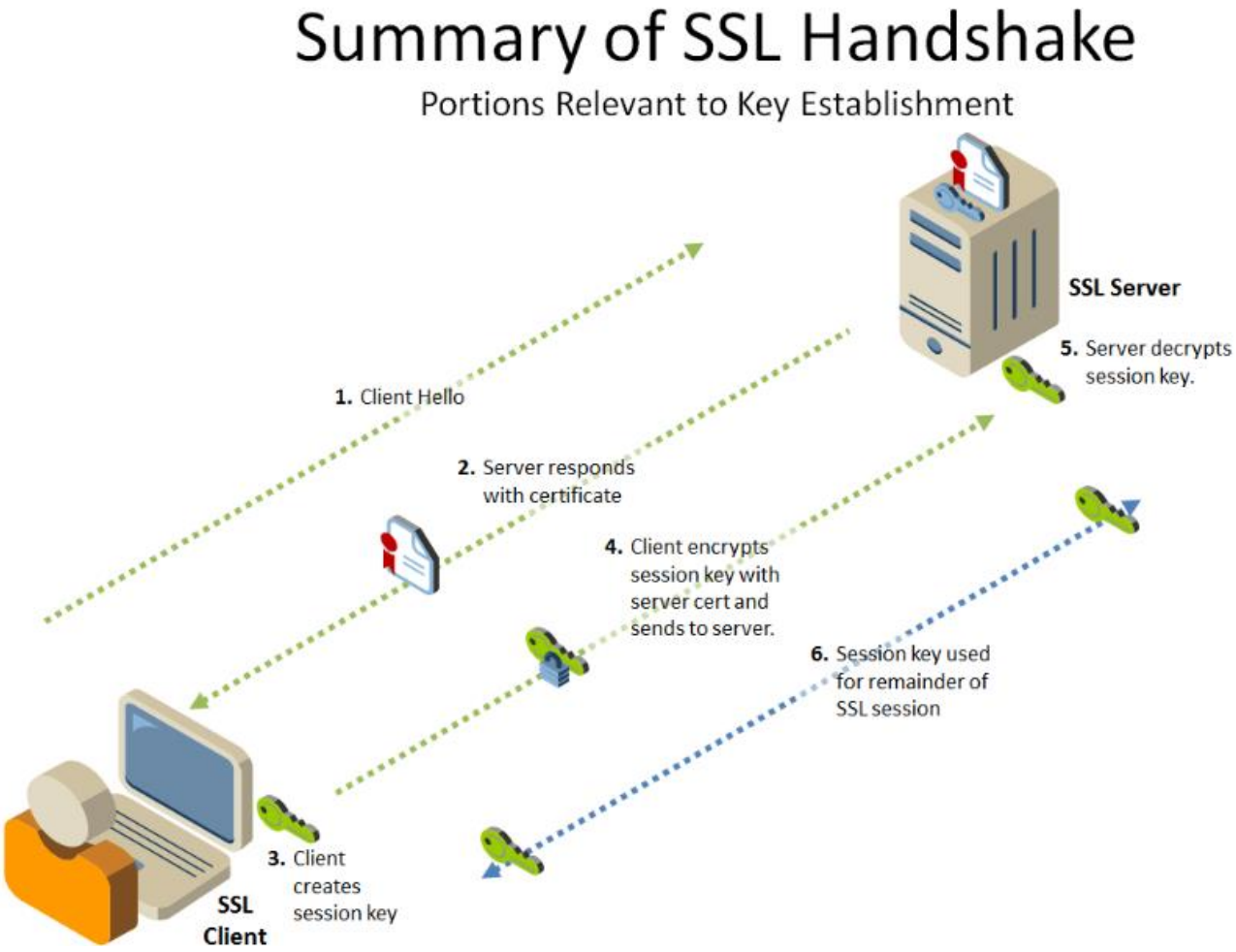


Рис. 1.3 Процес SSL/TLS-шифрування

Протоколи OAuth та OpenID сьогодні є фундаментом сучасної системи автентифікації та авторизації в Інтернеті, і саме завдяки ним користувачі можуть швидко та безпечно входити до різноманітних онлайн-сервісів. Вони дозволяють уникнути необхідності запам’ятовувати десятки різних логінів і паролів, а також значно зменшують ризики, пов’язані з крадіжкою автентифікаційних даних.

Загальний принцип роботи

Традиційна модель автентифікації передбачала, що для доступу до кожного нового ресурсу користувач створює окремий обліковий запис із власним паролем. Це не лише незручно, а й небезпечно: користувачі часто повторно використовують ті самі паролі, що дає зловмисникам додаткові можливості для атак. Використання OAuth та OpenID усуває ці ризики, адже користувач більше не передає свій пароль кожному новому сайту. Замість цього застосовуються токени доступу, які підтверджують його особу та регулюють рівень дозволів.

Уявімо, що ви хочете зареєструватися на новому сайті. Замість створення нового акаунта ви обираєте варіант «Увійти за допомогою Google» чи «Авторизуватися через Facebook». У цьому випадку сайт не отримує ваш пароль від Google чи Facebook, а лише токен, який засвідчує, що ви дійсно є власником облікового запису. Цей токен має обмежений строк дії та може бути відкликаний у будь-який момент.

OAuth — протокол керування доступом

OAuth (Open Authorization) є стандартом, розробленим для того, щоб сторонні сервіси могли отримувати обмежений доступ до облікових записів користувачів без необхідності дізнаватися їхні паролі. Наприклад, мобільний додаток може запитати дозвіл на читання вашого профілю у Twitter чи Facebook, на доступ до списку контактів у Google або на публікацію повідомлень від вашого імені. Ви підтверджуєте цей запит, і додаток отримує відповідний токен із конкретними правами. Таким чином, контроль доступу залишається у ваших руках, а ризики розголошення пароля повністю усуваються.

Особливо важливим є те, що **OAuth дозволяє задавати різні рівні доступу**. Наприклад, один сервіс може отримати лише право читання профілю, інший — право надсилати повідомлення, а третій — доступ до електронної пошти. Це надає гнучкість і безпеку, адже користувач чітко знає, яку інформацію він надає і кому.

OpenID — єдиний цифровий ідентифікатор

OpenID розвиває цю ідею ще далі. Якщо OAuth фокусується на делегуванні прав доступу, то OpenID зосереджений на автентифікації — підтвердженні особи користувача. Головна ідея полягає у створенні єдиного цифрового ідентифікатора, який можна використати для входу на різні сайти та сервіси. Тобто замість десятків пар логінів і паролів користувач отримує один універсальний акаунт, наприклад, у Google чи іншому провайдері OpenID. Це значно полегшує процес і зменшує ризики, адже пароль вводиться лише на одній довіреній платформі.

Таким чином, OpenID можна розглядати як механізм «єдиного входу» (Single Sign-On), що спрощує взаємодію користувачів із величезною кількістю онлайн-платформ.

Переваги використання

Застосування OAuth та OpenID має очевидні переваги як для користувачів, так і для розробників:

- **Вищий рівень безпеки**: паролі зберігаються лише у провайдера, а сторонні сервіси працюють із токенами. Це суттєво зменшує ймовірність крадіжки автентифікаційних даних.
- **Простота використання**: користувачеві достатньо одного облікового запису, щоб отримати доступ до великої кількості ресурсів.
- **Гнучкість**: можна чітко регулювати, які саме дозволи надаються стороннім програмам чи сайтам.
- **Масштабованість**: для великих платформ використання єдиної схеми автентифікації знижує витрати на підтримку та адміністрування користувачів.

Значення для сучасного цифрового середовища

У світі, де кількість онлайн-сервісів зростає щодня, OAuth і OpenID стали стандартом де-факто для автентифікації. Вони інтегровані в соціальні мережі, банківські застосунки, освітні платформи, корпоративні рішення. Завдяки їм користувачі отримують єдину точку довіри, а розробники — універсальний інструмент управління доступом.

Таким чином, ці протоколи не лише підвищують безпеку, але й роблять цифрову взаємодію комфортнішою та зрозумілішою для мільйонів людей у всьому світі.

1.3 Авторизація та права доступу

Методи авторизації в базах даних

У сучасному світі інформаційних технологій бази даних є серцем більшості бізнес-процесів, адже саме вони містять критично важливу інформацію про клієнтів, фінансові операції, внутрішні ресурси та інші дані. Проте навіть за наявності розвинених систем інформаційної безпеки зберігається ризик того, що

користувач може отримати несанкціонований доступ до бази даних. Це створює реальну загрозу для конфіденційності даних, цілісності інформації та стабільності роботи організації. Несанкціонований доступ може призвести до витоку особистих даних клієнтів, фінансових втрат, порушення роботи компанії чи навіть повної втрати довіри з боку партнерів і користувачів.

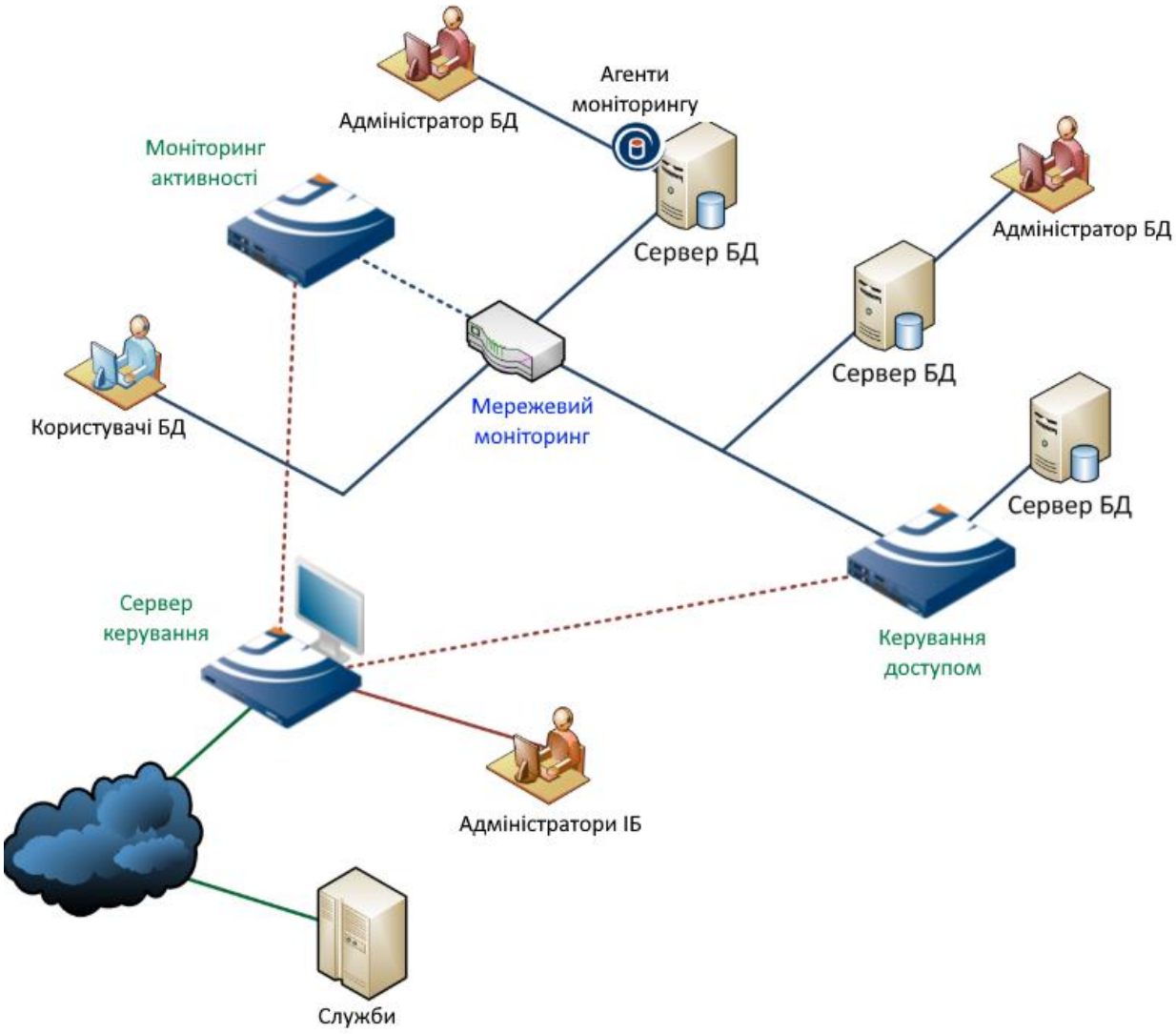


Рис. 1.4 Умовна структура компанії

Саме тому у сфері захисту інформації особлива увага приділяється методам авторизації, які визначають, хто і до якої частини бази даних має право доступу. Авторизація дозволяє не лише запобігати зловживанням, а й організовувати ефективну роботу з інформаційними ресурсами, адже кожен співробітник працює лише з тими даними, які безпосередньо потрібні йому для виконання професійних обов’язків.

Розмежування прав доступу на рівні бази даних

Одним із ключових методів є система прав доступу на рівні бази даних. Вона дозволяє адміністраторам задавати правила, які обмежують або дозволяють доступ до певних таблиць, рядків чи навіть окремих атрибутів даних. Завдяки цьому формується багаторівнева структура безпеки, де кожен користувач отримує саме ті повноваження, які йому необхідні, і не більше.

Наприклад, у компанії, що працює з великими масивами клієнтської інформації, у базі можуть зберігатися такі дані, як:

- **персональні відомості** (ім’я, адреса, дата народження, номер телефону),
- **історія покупок** (придбані товари, дати транзакцій, суми оплат),
- **контактні дані представників клієнтів**,
- **фінансові показники та звітність**.

Використовуючи права доступу, компанія може побудувати чітку ієрархію доступу:

- **аналітики** матимуть право бачити лише узагальнену інформацію про покупки, щоб проводити дослідження попиту, прогнозувати продажі та формувати звіти. Їм не потрібні контактні дані чи особиста інформація клієнтів;
- **відділ продажів** матиме доступ лише до контактів клієнтів для організації дзвінків, зустрічей та проведення маркетингових кампаній. Дані про фінансові транзакції чи деталі покупок для них будуть закритими;
- **бухгалтерія** отримає повноваження працювати з фінансовими звітами та платіжними операціями, але без доступу до персональних контактів чи внутрішньої бізнес-аналітики;
- **керівництво компанії** може мати розширений доступ до всіх даних, але з можливістю делегувати окремі повноваження іншим працівникам.

Переваги такого підходу

1. **Захист конфіденційності** – критично важливі дані не стають доступними тим, хто не має на це прав.
2. **Мінімізація ризиків** – навіть у випадку компрометації одного облікового запису зловмисник не зможе отримати доступ до всієї бази даних.

3. **Принцип найменших привілеїв** – кожен співробітник має лише ті права, які потрібні для його роботи.
4. **Підвищення ефективності роботи** – доступ до даних структурований, і працівники не витрачають час на інформацію, що не має відношення до їхніх завдань.
5. **Відповідність міжнародним стандартам безпеки** – системи із чітким розмежуванням прав доступу відповідають вимогам GDPR, ISO/IEC 27001 та інших норм, що особливо важливо для компаній, які працюють на глобальному ринку.

Методи авторизації, зокрема розмежування прав доступу на рівні бази даних, є основою сучасної стратегії інформаційної безпеки. Вони дозволяють досягти балансу між зручністю використання системи та надійністю захисту даних. У результаті організація отримує інструмент, що з одного боку запобігає витокам та зловживанням, а з іншого — підтримує ефективність бізнес-процесів.

2 СИМЕТРИЧНИЙ ТА АСИМЕТРИЧНИЙ ЗАХИСТ ВІД DDOS

2.1 Поняття симетрії в телекомунікаціях

Поняття **симетрії** здається інтуїтивно зрозумілим кожному, однак сформулювати його строго здатні не всі. Якщо якийсь об’єкт називають симетричним, то мають на увазі, що він не змінюється під дією певних перетворень, які називають **перетвореннями симетрії**. Це поняття є універсальним і застосовується як у математиці чи фізиці, так і в прикладних галузях, зокрема в інженерії та телекомунікаціях.

Найпростішим і найнаочнішим прикладом симетрії є **геометрична симетрія**. Візьмемо квадрат: якщо його повернути на 90° навколо геометричного центра, він співпаде сам із собою. У такому випадку говорять, що «квадрат симетричний відносно повороту на 90°». Тут поворот є перетворенням симетрії, а множина точок, що утворюють квадрат, — властивістю, яка залишається незмінною. Такі перетворення належать до точкових симетрій, адже під час обертання хоча б одна точка простору (центр повороту) залишається нерухомою.

Інший приклад — **трансляційна симетрія**, коли об’єкт відтворює сам себе після зсуву на певний вектор. Класична ілюстрація — аркуш у клітинку: якщо змістити його на відстань, кратну розміру клітинки, ми отримаємо таку саму структуру. У цьому випадку вже немає нерухомих точок, а елементом симетрії виступає вектор трансляції.

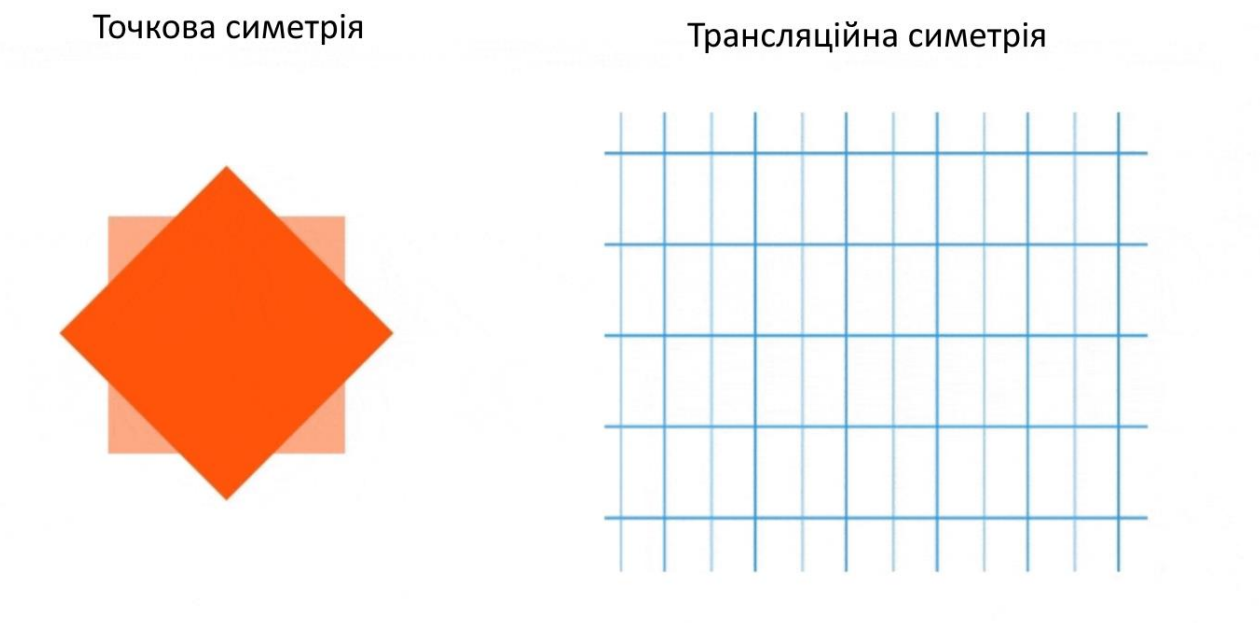


Рис. 2.1 Наозне зображення двох симетрій

Поняття симетрії активно використовується не лише в геометрії, а й у телекомунікаціях та інформаційних технологіях. Зокрема:

- **Симетричні лінії зв’язку.** У кабельних системах симетричними називають пари провідників, які однаково реагують на зовнішні електромагнітні впливи. Завдяки цьому забезпечується високий рівень завадостійкості й мінімізується вплив зовнішніх перешкод. Це критично важливо для стабільної роботи локальних мереж, телефонних систем і навіть сучасних широкосмугових доступів (наприклад, xDSL).
- **Симетрія сигналів.** У цифровій обробці сигналів (DSP) симетричні властивості використовуються для оптимізації алгоритмів, наприклад у перетворенні Фур’є. Сигнали з певними симетричними характеристиками легше аналізувати та передавати, оскільки їхні властивості дозволяють зменшити кількість обчислень і підвищити ефективність системи.
- **Симетрія в антенних решітках.** У телекомунікаційних мережах із багатьма антенами (MIMO, Massive MIMO у 5G/6G) симетричне розташування елементів антенної решітки забезпечує рівномірне формування діаграм спрямованості та оптимальне покриття території. Тут геометрична симетрія напряму впливає на якість зв’язку.

- **Трансляційна симетрія в мережах.** Телекомунікаційні системи часто мають структури, що повторюються. Наприклад, у стільникових мережах базові станції розташовуються за принципом сотової решітки (гексагональної), яка є прикладом трансляційної симетрії. Завдяки цьому вдається забезпечити рівномірне покриття території без «мертвих зон».

Симетрія та асиметрія в телекомунікаційних мережах

У телекомунікаціях поняття симетрії та асиметрії часто використовуються для опису принципів передачі інформації. Якщо говорити спрощено, то симетрія в мережах означає однаковість властивостей каналу незалежно від напрямку руху даних: від клієнта до сервера чи від сервера до клієнта. Це можуть бути пропускна здатність, маршрут, яким рухаються пакети, рівень затримок чи ймовірність втрат пакетів. Якщо ж параметри в одному напрямку відрізняються від іншого, виникає асиметрія.

Ранні симетричні мережі

Історично перші мережі мали повністю симетричну будову. Прикладом є перші телефонні лінії, коли апарати з’єднувалися безпосередньо між собою. Кількість можливих абонентів визначалася числом окремих кабелів, прокладених до інших користувачів. Згодом з’явилися телефонні станції, де оператор вручну поєднував проводи, створюючи канал між абонентами. Незалежно від методу встановлення з’єднання, результат був однаковий — між двома абонентами створювався «провід», який залишався активним на весь час розмови. Такі мережі отримали назву мереж із комутацією каналів.

У 1891 році Елмон Стоуджер запатентував кроковий шукач — електромеханічний комутаційний апарат, який став основою для перших АТС (автоматичних телефонних станцій). Це дало змогу автоматизувати процес комутації, але сам принцип залишався незмінним: встановлювався симетричний канал із однаковими характеристиками у прямому та зворотному напрямках. Головним недоліком таких мереж була низька ефективність використання ресурсів — навіть коли канал не використовувався, він залишався зарезервованим і «з’їдав» потужності мережі.

Поява пакетних мереж та асиметрія

З розвитком електроніки та зростанням кількості користувачів стало очевидно, що комутація каналів не може забезпечити масштабованість і ефективність. Уявімо, що сучасний Інтернет із понад 27 мільярдами пристроїв працював би на базі комутації каналів: реально одночасно використовувалося б менше 1% каналів, тоді як інші залишалися б заблокованими, але невикористаними. Це стимулювало перехід до мереж із комутацією пакетів.

У таких мережах інформація розбивається на пакети, кожен із яких містить службову інформацію (IP-адреси, порти, ідентифікатори протоколів) та може рухатися самостійним маршрутом. Наприклад, протокол UDP передає пакети без встановлення постійного з'єднання, а TCP створює віртуальні з'єднання, які в деякому сенсі імітують комутацію каналів, але при цьому залишаються частиною пакетної архітектури.

Переваги такого підходу очевидні:

- мережа легко масштабується завдяки відсутності необхідності створювати окремий канал для кожної пари абонентів;
- забезпечується відмовостійкість, оскільки дані можна доставляти кількома маршрутами;
- ресурси використовуються максимально ефективно, адже кілька потоків трафіку можуть одночасно ділити одну й ту ж фізичну лінію.

Проте саме пакетна комутація привела до появи явища асиметрії.

Сутність асиметрії в мережах

Асиметрія проявляється у різниці характеристик мережі залежно від напрямку руху трафіку. Вона може виникати на різних рівнях:

- **канальному** — різниця у пропускній здатності (наприклад, у DSL-з'єднаннях швидкість завантаження значно більша за швидкість відвантаження);
- **маршрутизаційному** — пакети у напрямку від вузла А до вузла Б і у зворотному напрямку можуть проходити різними шляхами;

- **рівні QoS** — різні затримки, втрати пакетів чи джиттер у прямому та зворотному каналах.

  У стандартах RFC закріплені відповідні терміни:
- **Forward direction** — прямий напрямок, зазвичай із вищою пропускною здатністю та нижчими затримками.
- **Reverse direction** — зворотний напрямок, який часто поступається за якістю.
- **Upstream link** — аплінк із меншою пропускною здатністю, яким дані надходять «угору» до провайдера.
- **Downstream link** — даунлінк із більшою пропускною здатністю, яким користувач отримує дані.

Причини асиметрії

Причини появи асиметрії можуть бути як технічними, так і економічними. Технічні обмеження залежать від архітектури мережі чи типу обладнання. Проте значну роль відіграють і бізнес-стратегії операторів. Відомою є практика hot-potato routing («гаряча картопля»), коли оператор намагається якомога швидше передати транзитний трафік сусідньому оператору, щоб знизити власні витрати на його обслуговування. Це призводить до того, що маршрути прямого та зворотного трафіку відрізняються, створюючи додаткову асиметрію.

2.2 Видідення потоку та захист асиметричних даних

Тепер доцільно звернутися до прикладів наукових досліджень, які детально вивчають масштаби асиметрії у телекомунікаційних мережах, і спробувати зрозуміти, наскільки суттєво вона впливає на роботу глобального Інтернету. Одним із ключових є дослідження під назвою *«Observing routing asymmetry in Internet traffic»*, у якому автори поставили за мету проаналізувати трафік у прямому та зворотному напрямках і порівняти їхню поведінку на різних рівнях організації магістральних мереж. Особливу увагу вони приділили мережам операторів рівнів Tier-2 і Tier-1, а також маршрутам, що проходять на стиках між їхніми інфраструктурами. Це дало можливість простежити не лише внутрішню симетрію

в межах одного оператора, а й взаємодію між різними провайдерами, що є надзвичайно важливим для розуміння глобальних характеристик Інтернету.

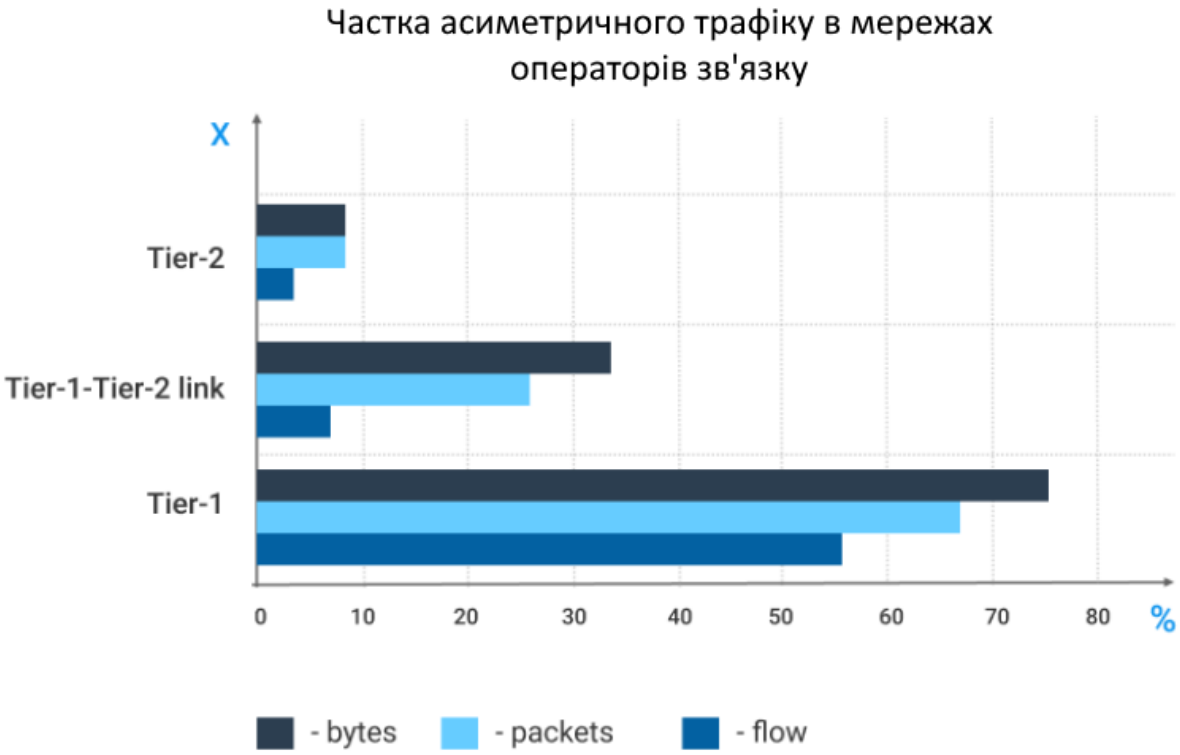


Рис. 2.2 Частка асиметричного трафіку в мережах

Щоб зробити аналіз більш системним, спочатку було введено поняття потоку (Flow). Потоком називається послідовність пакетів, які мають однакові параметри: IP-адресу відправника, IP-адресу отримувача, порт джерела, порт призначення та ідентифікатор протоколу. Якщо поміняти місцями відправника та отримувача, ми фактично отримуємо той самий потік, але в протилежному напрямку. Це визначення дозволяє порівнювати властивості прямого і зворотного напрямків єдиного логічного зв’язку між двома вузлами.

Дослідники розрахували відсоток потоків, які мали передачу пакетів в обох напрямках, після чого всередині кожного такого потоку вони порівняли кількість пакетів та загальний обсяг переданих даних у байтах. Таким чином вони отримали можливість оцінити, наскільки збалансованими є два напрями одного каналу. Для уникнення викривлення результатів із загальної вибірки було вилучено протоколи, які за своєю природою від самого початку є асиметричними і не передбачають підтвердження доставки, наприклад UDP. Це дало змогу зосередитися саме на тих

типах трафіку, які в теорії повинні демонструвати більшу симетричність, а отже, показати реальні відхилення, зумовлені архітектурою мережі та принципами маршрутизації.

Отримані результати показали, що навіть у випадках, коли протоколи орієнтовані на двосторонній обмін і логічно передбачають симетрію, у реальному Інтернеті спостерігаються значні асиметрії. Це проявляється у різниці кількості пакетів у двох напрямах, у відмінностях за обсягом переданих даних, а також у зміні маршрутів, якими йде інформація. Виявилося, що на рівні магістральних операторів, особливо між Tier-1 і Tier-2 мережами, симетрія майже ніколи не досягається повністю, і зворотний трафік часто йде іншим шляхом, відрізняючись за характеристиками затримки, пропускної здатності чи втрат.

Симетрія потоків у дослідженнях Інтернет-трафіку розглядається з кількох позицій, оскільки сама по собі вона може проявлятися як на рівні логічних з'єднань, так і на рівні фактичного розподілу пакетів чи байтів. Найпростіший спосіб оцінити симетричність полягає у визначенні частки потоків, які мають трафік у двох напрямках. Цей показник розраховується як відношення числа двосторонніх потоків до загальної кількості потоків, виражене у відсотках. Чим вищий цей показник, тим більша кількість сеансів зв'язку підтримує обмін у прямому та зворотному напрямках, тобто демонструє більш «збалансовану» структуру.

Проте однієї лише наявності обміну в обох напрямках недостатньо, адже часто трафік буває нерівномірним. Тому вводяться додаткові метрики. Симетрія за кількістю пакетів оцінює, наскільки рівномірно розподіляється навантаження: вона розраховується за формулою $(1 - |Nab - Nba| / |Nab + Nba|) \times 100\%$, де Nab — кількість пакетів у прямому напрямку, а Nba — у зворотному. Аналогічний підхід застосовується і для симетрії за обсягом даних у байтах, де порівнюються сумарні обсяги трафіку в обох напрямах. У цих формулах значення, що наближається до 100%, свідчить про майже ідеальну рівновагу, а значні відхилення вказують на виражену асиметрію.

Результати досліджень показали закономірність: чим ближче трафік до кінцевих користувачів, тим вищий рівень симетрії. Це пояснюється тим, що на рівні

локальних мереж чи edge-маршрутизаторів передача зазвичай носить двосторонній характер: користувач відправляє запит — сервер повертає відповідь, і співвідношення пакетів та байтів у напрямках залишається відносно рівним. Однак у міру просування трафіку в глибину магістральної інфраструктури картина змінюється. На рівні міжоператорських зв'язків і всередині великих автономних систем асиметрія стає значно вираженішою. Це пов'язано з особливостями маршрутизації, політиками обміну трафіком між провайдерами та економічними факторами, які змушують операторів оптимізувати використання своїх ресурсів, іноді навіть ціною симетричності.

Особливий інтерес становить питання симетрії маршрутів. У роботі *«How Asymmetric Is the Internet?»* автори спробували кількісно оцінити, наскільки різними є шляхи прямого і зворотного трафіку на рівні автономних систем (АС). Автономна система — це набір IP-мереж та обладнання, що перебувають під єдиним адмініструванням і мають спільну політику маршрутизації у взаємодії з глобальним Інтернетом.

Для проведення дослідження було використано 4000 вимірювальних зондів платформи RIPE Atlas, розташованих у різних АС по всьому світу. Ці зонди — це невеликі комп'ютери, спеціально призначені для вимірювань характеристик мережі у глобальному масштабі. З їх допомогою дослідники змогли виконати масові експерименти із застосуванням інструмента Traceroute, який визначає маршрут проходження пакетів і фіксує всі проміжні вузли (маршрутизатори), через які проходить трафік. Це дозволило отримати дані як для прямого напряму, так і для зворотного, а далі порівняти ці два набори шляхів між одними й тими ж вузлами.

Результати продемонстрували, що асиметрія маршрутів є нормою, а не винятком. У багатьох випадках прямий і зворотний потоки проходили через абсолютно різні автономні системи, що відразу ж впливало на затримки, пропускну здатність та стабільність зв'язку. Інакше кажучи, навіть якщо трафік починається і закінчується у тих самих точках, то шлях «туди» і шлях «назад» може мати різну довжину, різний набір проміжних вузлів і, як наслідок, різну якість обслуговування.

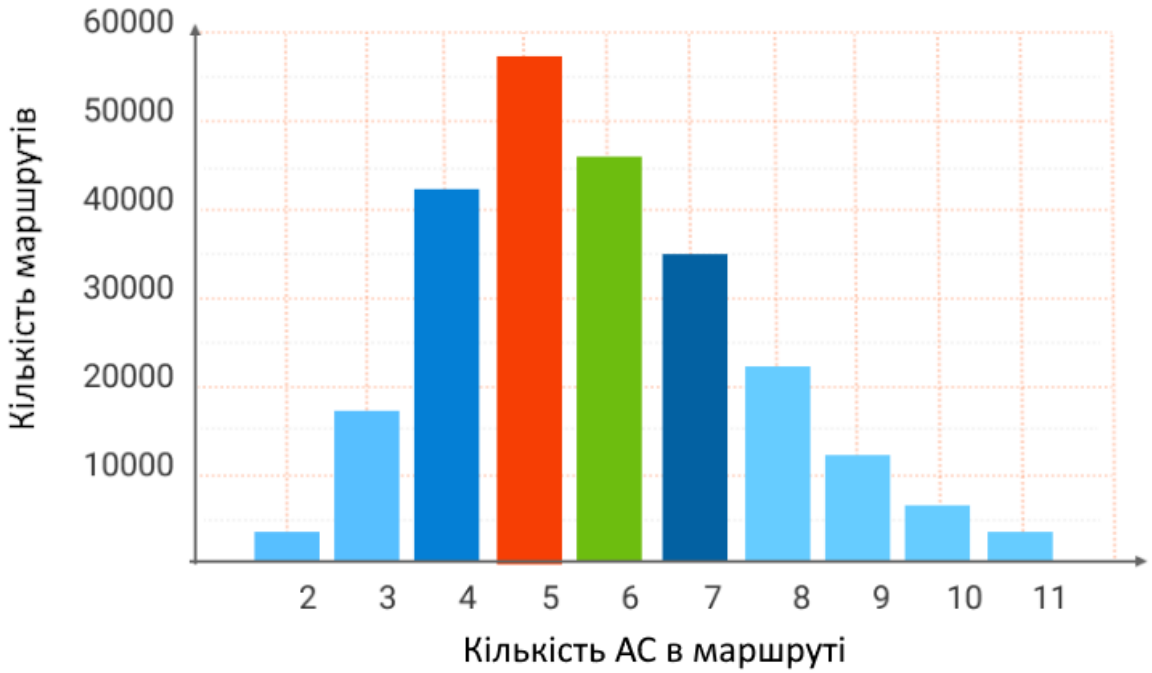


Рис. 2.3 Відомості про всіх проміжних маршрутизаторів

Дослідження показало, що повністю симетричними щодо зміни напрямку передачі даних виявилися лише 12,6% потоків. Для порівняння: у схожому аналізі, проведеному п’ять років тому, цей показник становив лише 10%, тож можна говорити про певне, хоч і незначне, покращення.

Окрім цього, автори дослідили маршрути однакової довжини у прямому та зворотному напрямках. Виявилося, що навіть за збігу кількості переходів (hop count) самі траєкторії можуть суттєво різнитися й проходити через різні автономні системи або вузли. Це підтверджує, що асиметрія маршрутизації є типовою властивістю Інтернету, яка проявляється навіть там, де на перший погляд спостерігається симетрія:

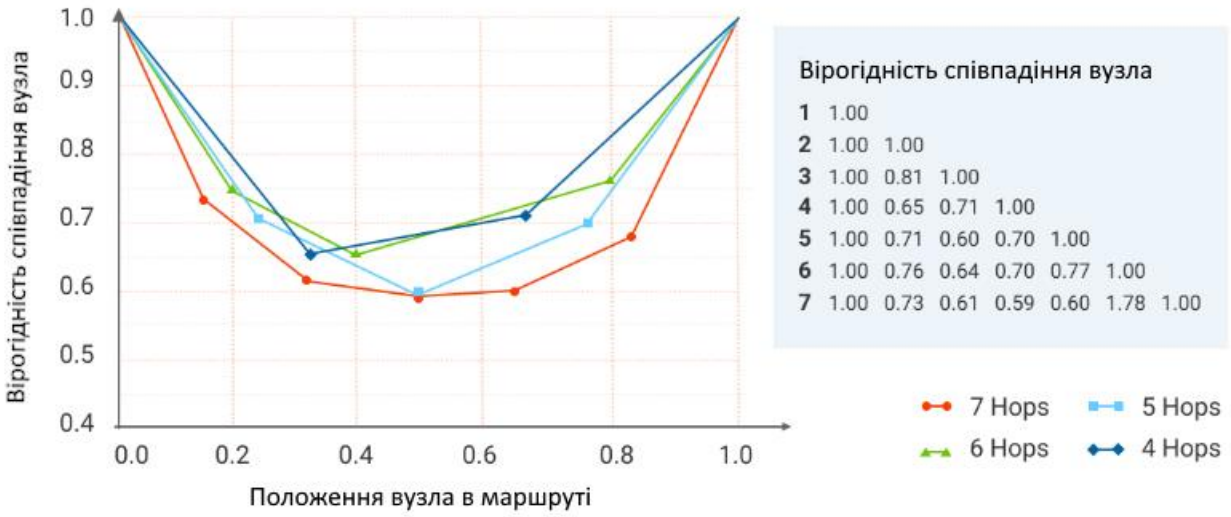


Рис. 2.4 Прояв асиметрії маршрутизації

На осі Y відображено ймовірність того, що вузли (у цьому випадку автономні системи, АС) будуть однаковими при русі з А до Б та у зворотному напрямку з Б до А. На осі X показано позицію вузла в маршруті. Для заданої пари вузлів початкова й кінцева точки завжди збігаються, тому ймовірність дорівнює 1. Водночас чим ближче до центральної частини маршруту, тим менше шансів, що вузли у прямому та зворотному напрямках співпадуть.

Симетрія та асиметрія при підключенні систем захисту від DDoS

У сфері кіберзахисту існує два базові підходи до організації підключення систем фільтрації трафіку: симетричний та асиметричний. Під симетричною схемою мається на увазі такий варіант, коли через захисну інфраструктуру проходить увесь трафік — як вхідний, так і вихідний. Це може бути єдиний фільтрувальний пристрій, кілька спеціалізованих апаратно-програмних комплексів або ціла мережа провайдера послуг захисту. Завдяки цьому будь-яке рішення щодо блокування чи дозволу на проходження пакетів ґрунтується на аналізі повної картини обміну даними. Асиметрична схема, навпаки, передбачає, що система бачить лише один напрямок — зазвичай вхідний трафік, залишаючись «сліпою» до вихідного.



Рис. 2.5 Симетрична та асиметрична схеми підключення захисту

Симетрична схема не завжди означає, що кожен пакет без винятку перевіряється у двох напрямах. Головна ідея полягає в тому, що система має змогу оцінити контекст — взаємозв'язок запитів і відповідей, швидкість обміну, аномальні шаблони комунікацій. Це можна порівняти з тим, як ми читаємо діалог між людьми. Якщо бачити лише половину повідомлень, ризик неправильного тлумачення зростає багаторазово.

Уявімо уривок діалогу:

А: Привіт, ну що, все в силі?

Б: …

А: А хто ще буде?

Б: …

А: Буде щось серйозне!

Б: …

Якщо спиратися лише на ці рядки, можна дійти висновку, що готується конфлікт або навіть насильницька подія. Проте варто додати другу частину розмови — і зміст докорінно змінюється:

А: Привіт, ну що, все в силі?

Б: Привіт. Звичайно. Не забудь взяти ноутбук.

А: А хто ще буде?

Б: Сергій, Іван, Дмитро та Олександр.

А: Буде щось серйозне!

Б: Авжеж.

Повний контекст демонструє, що це зовсім не про бійку, а про дружню зустріч із грою по локальній мережі, як у старі добрі часи.

Ця аналогія точно відображає суть проблеми в телекомунікаціях. Коли система захисту аналізує лише половину трафіку (наприклад, тільки вхідні пакети), вона ризикує неправильно класифікувати подію: видати хибне спрацювання або, навпаки, пропустити атаку. Лише за наявності симетричного огляду — тобто видимості як запитів, так і відповідей — з'являється можливість коректно оцінити ситуацію.

Для прикладу, багато DDoS-атак базуються на техніці відображення та підробки адрес (reflection та spoofing). Якщо система бачить лише вхідний трафік, вона може трактувати масований потік відповідей від зовнішніх ресурсів як легітимні повідомлення. Але якщо аналізується і вихідний напрям, легко виявити, що від внутрішніх клієнтів такі запити ніколи не надходили, отже, відповідь є результатом атаки.

Симетричний підхід до захисту дозволяє:

- точніше виявляти аномалії та відрізняти їх від легітимного трафіку,
- знижувати кількість хибних блокувань, що особливо критично для бізнесу,
- оперативно реагувати на нові методи атак, які часто маскуються під нормальні запити,
- оптимізувати розподіл ресурсів захисної інфраструктури, оскільки рішення ухвалюються на основі повної інформації.

Таким чином, симетричне підключення системи захисту від DDoS не просто технічний нюанс архітектури, а стратегічно важливий підхід, що визначає якість і надійність протидії атакам. Він дозволяє отримати цілісне уявлення про події в мережі й будувати рішення, які справді знижують ризики, а не лише створюють ілюзію безпеки.

Приклад: захист від SYN flood

Відмінність у точності фільтрації між симетричними та асиметричними схемами особливо яскраво проявляється під час роботи з протоколами, які вимагають попереднього встановлення з'єднання, зокрема TCP. Для початку обміну

даними клієнт і сервер повинні пройти процедуру, відому як триетапне рукостискання (3-way handshake). На першому етапі клієнт надсилає SYN-пакет із випадковим номером початкової послідовності A, щоб ініціювати з'єднання. Сервер приймає цей пакет, зберігає номер A та формує SYN-ACK, у якому міститься підтвердження отримання SYN, номер A+1, а також новий випадковий номер послідовності B, згенерований сервером. У відповідь клієнт надсилає ACK-пакет, у якому підтверджує отримання, передаючи значення B+1. Лише після цього TCP-з'єднання вважається відкритим, і сервер вносить відповідний запис до таблиці з'єднань, де воно зберігається до завершення або розриву сесії.

Атака типу SYN flood базується саме на особливостях цього процесу. Зловмисник надсилає велику кількість SYN-запитів, підробляючи IP-адреси джерела. Сервер формує відповідні SYN-ACK пакети й чекає завершального ACK, який ніколи не надходить. У таблиці накопичуються напіввідкриті з'єднання, що поступово вичерпують ресурси пам'яті й призводять до відмови в обслуговуванні. Нові легітимні запити не можуть бути оброблені, адже у сервера просто не залишається вільних ресурсів для створення нових записів.

Захист від подібної атаки значно ефективніший у випадку симетричної схеми підключення, коли фільтрувальна система має доступ як до вхідного, так і до вихідного трафіку. У такому разі можна застосувати спеціалізовані методи, зокрема SYN cookies та SYN proxy. SYN cookies дозволяють уникнути витрат ресурсів на збереження інформації про з'єднання до моменту його підтвердження. Сервер не створює запис у таблиці, поки не переконається, що клієнт легітимний. Замість цього дані про потенційне з'єднання кодуються у номер послідовності B, який сервер передає клієнту у SYN-ACK. Якщо клієнт справді існує й відповідає ACK із значенням B+1, сервер відновлює інформацію та лише тоді створює запис у таблиці. Таким чином ресурси витрачаються лише на реальні з'єднання.

Ще один метод — SYN proxy. Він полягає у використанні проміжного пристрою, який приймає SYN від клієнта, відповідає SYN-ACK, перевіряє ACK і лише після цього встановлює з'єднання із сервером. Для самого сервера це вже виглядає як повністю підтверджене з'єднання, і він не бачить «сирих» запитів. При

цьому SYN proxy підміняє номери послідовностей у пакетах між клієнтом і сервером, щоб забезпечити коректність обміну. По суті, це протидія спуфінгу за допомогою контрольованого «спуфінгу» з боку системи захисту.

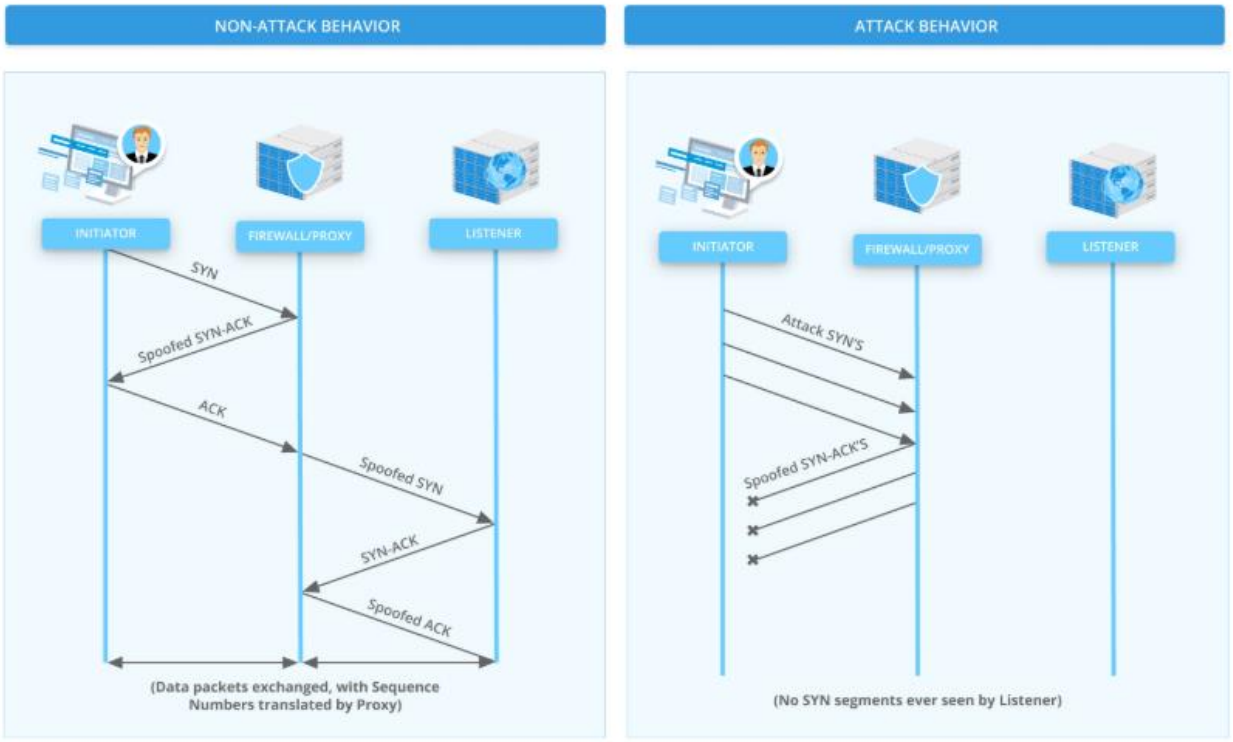


Рис. 2.6 SYN proxy

Зрозуміло, що й при асиметричному підключенні також можна боротися з SYN flood, проте точність і ефективність помітно знижуються. Коли система бачить лише один напрям трафіку, вона не має повного контексту. Це ускладнює відрізнення легітимних клієнтів від зловмисників, особливо якщо використовуються складні методи атак чи динамічна маршрутизація.

Ситуація ускладнюється ще більше для протоколів без встановлення з'єднання, таких як UDP, QUIC чи ICMP. Вони не мають триетапного рукостискання, і це робить класифікацію трафіку ще важчою. Проте навіть у таких випадках дані про вихідний трафік можуть допомогти. Якщо порівняти кількість отриманих і відправлених дейтаграм у рамках одного потоку, можна виявити невідповідності. Це дозволяє перейти від звичайного шейпінгу, який лише обмежує швидкість передачі, до справжньої інтелектуальної фільтрації, що ефективно блокує зловмисні запити.

Таким чином, приклад SYN flood наочно демонструє, чому симетрична схема захисту є більш надійною. Вона дозволяє точніше аналізувати ситуацію, ефективніше розподіляти ресурси та забезпечувати захист не лише від класичних DDoS-атак, а й від нових, більш складних сценаріїв. У свою чергу, асиметричні схеми залишаються компромісом, придатним для певних середовищ, але значно поступаються за глибиною та точністю аналізу.

2.3 Використання фільтруючого обладнання

Іноді виникає потреба самостійно встановити обладнання для фільтрації трафіку, або ж просто цікаво зрозуміти, як подібні системи можуть бути підключені у провайдера послуг захисту.

У випадку, коли в мережі використовується лише один edge-маршрутизатор, усе спілкування з «зовнішнім світом» відбувається саме через нього. Це означає, що трафік у такій конфігурації буде повністю симетричним відносно потоків. Рішення у цьому випадку є простим: достатньо встановити фільтрувальний пристрій на відповідний канал і використовувати його у симетричному режимі, адже саме так досягається найвища точність аналізу та блокування шкідливого трафіку.

Ситуація стає набагато складнішою, якщо у мережі присутні декілька edge-маршрутизаторів. У такій конфігурації трафік розподіляється між різними каналами і набуває асиметричного характеру. Частина запитів може йти одним маршрутизатором, а відповіді — іншим, що створює додаткові труднощі для системи захисту. Саме в таких випадках потрібно враховувати особливості маршрутизації, балансування навантаження та використовувати більш гнучкі схеми підключення фільтрувального обладнання.

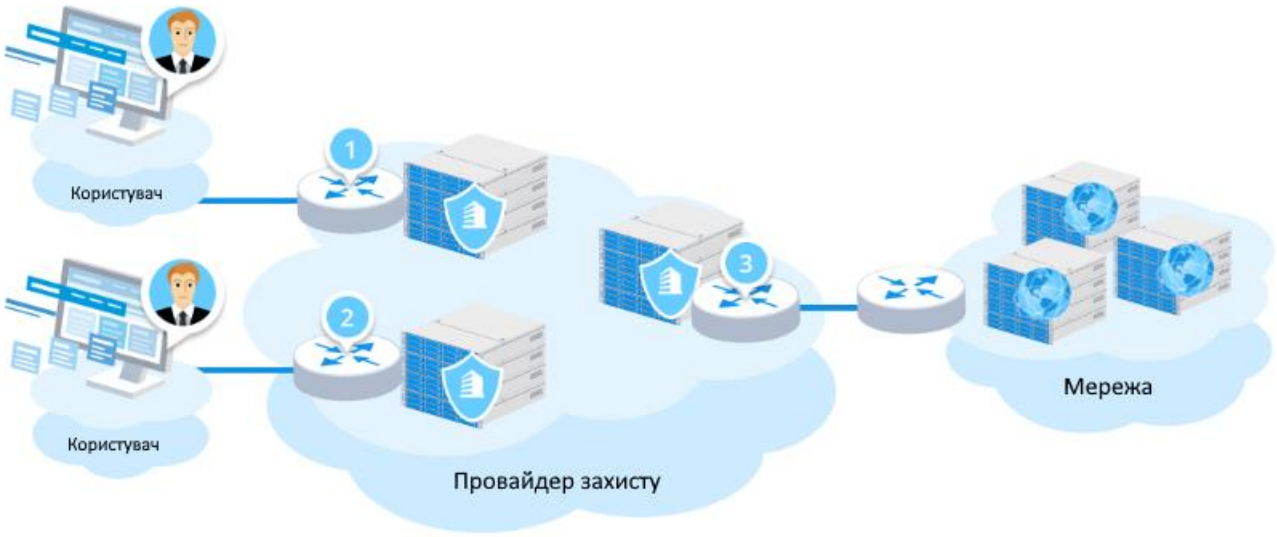


Рис. 2.7 Приклад схеми для встановлення захисту

У випадку, коли мережа має кілька edge-маршрутизаторів, для провайдера значно простіше та економічніше розгорнути кілька окремих фільтрувальних пристроїв, що працюють в асиметричному режимі. Це пояснюється одразу кількома факторами. По-перше, така схема легко масштабується: з’явився новий маршрутизатор на межі мережі — достатньо встановити на нього додатковий фільтр, і він готовий до роботи. По-друге, асиметрична архітектура знижує ризики негативного впливу потужних атак на решту мережі: навантаження розподіляється між кількома пристроями, і відмова одного з них не паралізує усю систему. Однак за ці переваги доводиться розплачуватися точністю. Оскільки кожен пристрій бачить лише частину потоків, виникає небезпека неправильної класифікації трафіку — легітимні пакети можуть бути заблоковані, а шкідливі, навпаки, залишитися непоміченими.

Якщо ж завдання полягає у тому, щоб аналізувати обидва напрямки передачі даних, то постає значно складніша проблема. Із загальної схеми видно, що симетрія зберігається лише на певному маршрутизаторі (умовно — №3). Це дозволяє підвищити точність фільтрації, проте створює одразу дві суттєві загрози. Перша — фінансова: увесь атакуючий трафік доводиться пропускати через власну мережу до

маршрутизатора №3, що веде до значного перевантаження каналів і збільшення витрат. Друга — архітектурна: маршрутизатор №3 стає єдиною точкою відмови, і у випадку його пошкодження або виходу з ладу уся система захисту фактично перестає виконувати свої функції.

Одним із можливих рішень є копіювання всього трафіку, що входить через маршрутизатор №3, на інші фільтрувальні пристрої. Це дає можливість обладнанню, яке працює на маршрутизаторах №1 та №2, бачити ті частини потоків, що раніше залишалися прихованими, і, відповідно, забезпечувати більш точну фільтрацію. Проте таке рішення вимагає високої пропускної здатності внутрішніх каналів і суттєво збільшує навантаження на інфраструктуру.

Більш економним, але й складнішим у реалізації варіантом є синхронізація фільтрувальних пристроїв та вибірковий обмін даними між ними. Для цього не обов’язково передавати усі пакети — достатньо обмінюватися метаданими, що значно знижує навантаження на мережу. Такий підхід дозволяє системі мати повне уявлення про потік навіть тоді, коли пакети одного й того ж з’єднання надходять через різні маршрутизатори (наприклад, одночасно через маршрутизатори №1 та №2). Якщо провести аналогію з розмовою, то в асиметричній схемі ми чуємо лише окремі уривки фраз, у той час як синхронізована система збирає докупи усі репліки та формує цілісну картину діалогу.

Проєктуючи подібні системи, необхідно враховувати фундаментальну властивість Інтернету — його асиметричність. Як було показано у попередніх дослідженнях, прямі й зворотні маршрути часто проходять різними шляхами, а обсяги трафіку в різних напрямах можуть суттєво відрізнятися. З огляду на це, архітектура захисту повинна балансувати між точністю, вартістю та стійкістю. Просте дублювання потоків підвищує якість фільтрації, але збільшує витрати, тоді як синхронізація й обмін метаданими дозволяють зберегти ефективність і водночас оптимізувати ресурси. Саме цей підхід у перспективі вважається найбільш гнучким для провайдерів, які будують багаторівневі системи DDoS-захисту у великих і складних мережах.

Захист від DDoS як сервіс

Сучасний підхід до боротьби з DDoS-атаками все частіше реалізується у вигляді послуги, яку надають спеціалізовані компанії. Коли ви користуєтесь DDoS Protection as a Service (DPaaS), головний тягар відповідальності за фільтрацію та очищення трафіку перекладається на провайдера. Власнику мережі достатньо лише організувати підключення: передати анонси своїх підмереж через протокол BGP, налаштувати тунель або укласти пряме з'єднання з центром захисту. Далі весь процес — від прийому до аналізу і відсіювання шкідливих потоків — виконує інфраструктура постачальника послуг. Такий варіант забезпечує вищу точність обробки запитів, а також дозволяє компанії зосередитися на бізнес-завданнях, не витрачаючи ресурси на будівництво власних систем фільтрації. Здавалося б, ідеальне рішення — це повна симетрія, коли аналізується увесь трафік у двох напрямках. Проте на практиці існують випадки, коли асиметрична схема підключення може виявитися доцільнішою та навіть економічно вигіднішою.

Однією з таких ситуацій є бажання провайдера зберігати максимально широкий контроль над власними маршрутами та вихідним трафіком за допомогою BGP. Це особливо актуально тоді, коли компанію пов'язують угоди про піринг із іншими операторами або договори з аплінками, у яких прописані конкретні вимоги до співвідношення вхідного та вихідного трафіку. У подібних умовах симетричне підключення може створити обмеження для гнучкого управління маршрутами, тоді як асиметрична схема залишає більшу свободу дій.

Інша поширена ситуація — використання одразу кількох провайдерів захисту. На перший погляд, мультивендорна стратегія здається перспективною: у разі відмови одного захисника інший бере на себе навантаження. Проте практика показує, що паралельне використання кількох постачальників часто створює більше проблем, ніж користі. Основні труднощі пов'язані з розподілом зон відповідальності, ускладненням процесу усунення неполадок та необхідністю постійної комунікації з кількома різними компаніями. Раціональніше мати одного основного захисника і ще одного — резервного, якого можна оперативно задіяти у разі виникнення критичних проблем.

Особливої уваги потребують застосунки, чутливі до затримок, наприклад онлайн-ігри, сервіси потокового відео чи платформи для відеоконференцій. Якщо центр очищення трафіку провайдера розташований занадто далеко, це збільшує затримку RTT (Round Trip Time), що негативно впливає на якість сервісу та роботу протоколів із підтвердженням доставки, зокрема TCP. У такому разі асиметрична схема може скоротити затримку майже вдвічі, оскільки трафік не доведеться повністю проганяти через віддалений дата-центр захисника. Це критично важливо для сервісів, де навіть додаткові 20–30 мс затримки можуть суттєво знизити якість обслуговування.

Не менш вагомими є й економічні фактори. У деяких регіонах світу вартість побудови чи підтримки симетричного захисту є надзвичайно високою. Наприклад, у країнах Латинської Америки імпортне обладнання обкладається високими митами та додатковими ліцензійними зборами, що можуть сягати 100% від його вартості. В таких умовах асиметричне підключення стає не просто технічним компромісом, а реальною можливістю заощадити суттєві кошти. При цьому навіть якщо точність фільтрації трохи зменшується, економічний ефект може виявитися значно вагомішим для бізнесу.

Усі ці приклади — контроль трафіку через BGP, мультивендорні рішення, критичність до затримок і висока вартість інфраструктури — є не лише аргументами на користь асиметричного підключення, але й свідченням того, наскільки важливо ретельно підходити до вибору постачальника. Окрім власне якості фільтрації, слід звертати увагу на можливості доставки очищеного трафіку, географічне розташування дата-центрів захисника та фінансові умови співпраці.

Незалежно від того, яку схему підключення ви оберете — симетричну чи асиметричну, — потрібно обов’язково перевірити, чи підтримує ваше обладнання необхідні технології. Жодна система DDoS-захисту не може гарантувати стовідсоткову точність фільтрації. Саме тому у міжнародній практиці використовується термін *mitigation* («пом’якшення», «зменшення наслідків»), який набагато точніше описує суть процесу. Важливо також оцінити запас міцності власної інфраструктури: чи зможе вона витримати зростання навантаження у

випадку, якщо під час масштабної атаки частина шкідливого трафіку все ж таки прорветься всередину. Для асиметричних схем цей запас має бути значно більшим, ніж для симетричних, адже ризик проникнення непрофільованих потоків у таких системах об’єктивно вищий.

Таким чином, захист від DDoS як сервіс — це не просто передача функцій безпеки на аутсорсинг. Це стратегічний вибір між точністю та економічною доцільністю, між мінімізацією затримок і масштабованістю рішень. І лише правильне поєднання технічних можливостей постачальника та характеристик власної інфраструктури дозволяє досягти оптимального балансу, який забезпечить стабільність роботи сервісів навіть під час наймасштабніших атак.

Дві поширені проблеми під час підключення захисту мережі та їх розв’язання

Організація захисту мережі від DDoS-атак завжди пов’язана з додатковими технічними викликами. Навіть у добре спроєктованих інфраструктурах можуть виникати проблеми з коректністю роботи протоколів, передаванням даних або балансуванням навантаження. Є дві найбільш поширені ситуації, які зустрічаються практично у всіх сценаріях підключення захисту, проте вони досить легко виправляються правильними налаштуваннями.

Проблеми з вихідним трафіком при асиметричній схемі маршрутизації

Однією з найчастіших причин некоректної роботи є використання механізму uRPF (Unicast Reverse Path Forwarding). Його завдання полягає у перевірці легітимності пакета: чи дійсно він надійшов із тієї IP-адреси, що вказана у полі *source IP*. Таким чином, uRPF виступає інструментом боротьби зі спуфінгом, коли зловмисники підробляють IP-адреси джерела.

Однак специфіка роботи цього механізму залежить від режиму його налаштування. У strict mode маршрутизатор звіряє, чи збігається інтерфейс, на який надійшов пакет, із тим, який сам маршрутизатор використав би для відправлення відповіді на цю IP-адресу. Якщо збігу немає, пакет відкидається. Такий підхід справді добре працює проти підроблених адрес, проте в умовах асиметричної маршрутизації це призводить до того, що блокуються навіть легальні пакети, які

йдуть через інший інтерфейс. У результаті користувачі можуть стикатися з частковою втратою зв’язку або періодичними збоями.

Вирішення цієї проблеми доволі просте: необхідно змінити режим роботи uRPF зі strict на loose mode. У такому випадку перевірка виконується менш жорстко — достатньо, щоб IP-адреса джерела була присутня у таблиці маршрутизації. Це дозволяє пропускати легальний трафік навіть при асиметричних шляхах проходження пакетів, не знижуючи при цьому рівень захисту від підроблених IP.

Зниження швидкості та втрати трафіку при використанні тунелів GRE / IPIP

Друга поширена проблема стосується підключення захисту через тунелі, наприклад GRE або IPIP. Найчастіше складнощі виникають через некоректно налаштоване значення MTU (Maximum Transmission Unit) — максимального розміру переданої одиниці інформації (PDU, Protocol Data Unit). PDU складається зі службової інформації та корисних даних (*payload*), а параметр MSS (Maximum Segment Size) вказує на максимально допустимий розмір цього корисного навантаження.

При створенні тунелю відбувається інкапсуляція: оригінальний PDU вкладається у payload PDU тунелю. Відповідно, щоб уникнути фрагментації, потрібно зменшити розмір MSS. Якщо цього не зробити, пакети почнуть ділитися на частини, що призводить до втрати швидкості, зростання затримок і навіть випадання з’єднань у пікових навантаженнях.

Щоб уникнути подібних наслідків, необхідно правильно підібрати параметри. Рекомендується встановити MTU не більше 1476 байт та MSS близько 1400–1436 байт (найчастіше використовують 1400). Крім того, слід заборонити фрагментацію на тунельному інтерфейсі, встановивши біт Don’t Fragment. Це забезпечить безперервну передачу даних без падіння швидкості та втрат через подрібнення пакетів.

Важливо враховувати ще один момент: у випадку асиметричної схеми підключення захисту вихідний трафік може йти не лише через тунель, а й безпосередньо через мережу провайдера. Тому параметри MSS необхідно

налаштовувати не тільки на тунельному інтерфейсі, а й на фізичних інтерфейсах, через які здійснюється вихід у глобальну мережу. Це дозволить уникнути проблем із сумісністю та забезпечить стабільну роботу усіх потоків даних.

3 ТЕХНОЛОГІЯ ЗАХИСТУ SRAM PUF

3.1 Технічні поняття технології

Фізично неклонована функція (Physically Unclonable Function, PUF) — це сучасний метод захисту, що базується на використанні унікальних фізичних властивостей конкретного об’єкта. На відміну від класичних підходів, де для автентифікації застосовуються ключі, сертифікати або інші заздалегідь визначені параметри, PUF використовує природну «індивідуальність» мікроелектронних пристроїв. Аналогією в біології можна вважати біометрію людини: відбитки пальців, геометрію обличчя, малюнок райдужної оболонки ока чи тембр голосу.

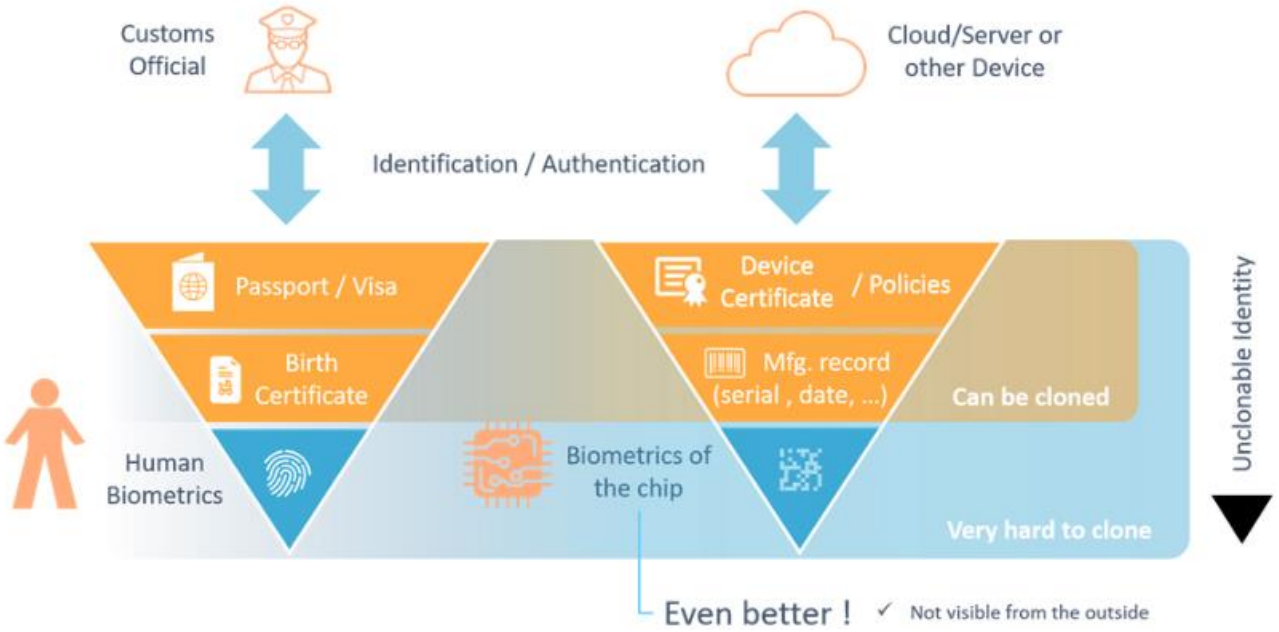


Рис. 3.1 Порівняння біометричних характеристик людини та «біометрії» мікросхеми

Якщо для ідентифікації особи ми користуємося паспортом або свідоцтвом про народження, то слід розуміти, що такі документи можна підробити. Натомість біометричні характеристики неможливо сфальсифікувати або повністю відтворити, адже вони закладені самою природою. І найважливіше — ці дані не потрібно зберігати у відкритому вигляді. Достатньо мати алгоритм, здатний зчитати інформацію й відтворити унікальний «цифровий відбиток» на основі допоміжних параметрів.

Схожий принцип працює і в мікроелектроніці. Кожна мікросхема, навіть якщо вона виготовлена на одному заводі й у тій самій технології, має власні індивідуальні фізичні характеристики. Зокрема, транзистори в кристалі відрізняються пороговими напругами чи параметрами провідності через мікроскопічні технологічні похибки під час виробництва. Ці відмінності практично неможливо відтворити навіть у межах одного технологічного процесу, і саме це робить їх ідеальним матеріалом для побудови PUF.

Одним із найбільш відомих прикладів реалізації є SRAM PUF. SRAM (Static Random Access Memory) — це енергозалежний тип пам’яті, що зберігає інформацію тільки за наявності живлення. Після кожного ввімкнення пристрою комірки SRAM ініціалізуються випадковими значеннями, які зумовлені індивідуальними властивостями транзисторів. Ці випадкові «стартові стани» є унікальними для кожної мікросхеми, утворюючи своєрідний «біометричний відбиток» пристрою. І головне — цей відбиток практично неможливо підробити або клонувати.

Завдяки такій особливості PUF знаходить широке застосування у сфері кібербезпеки та телекомунікацій. Наприклад:

- у криптографії PUF може виступати джерелом унікальних ключів для шифрування даних, без необхідності зберігати їх у пам’яті пристрою;
- у виробництві обладнання він використовується для захисту від підробок і контрафакту, адже кожен чип отримує свій «природний серійний номер»;
- у телекомунікаційних мережах PUF може забезпечувати автентифікацію обладнання та кінцевих пристроїв, знижуючи ризик підключення несанкціонованих вузлів;
- у хмарних сервісах та IoT-системах він застосовується для перевірки справжності сенсорів і контролерів, що особливо важливо при мільярдних підключеннях.

Технологія SRAM PUF базується на властивості статичної оперативної пам’яті (SRAM) формувати унікальний початковий стан при кожному увімкненні живлення. У момент подачі напруги кожна окрема комірка пам’яті випадково приймає значення логічного «0» або «1». Сукупність таких випадкових початкових

значень формує специфічний набір даних, який отримав назву SRAM Startup Data. Цей набір і виступає своєрідним «біометричним відбитком» мікросхеми.

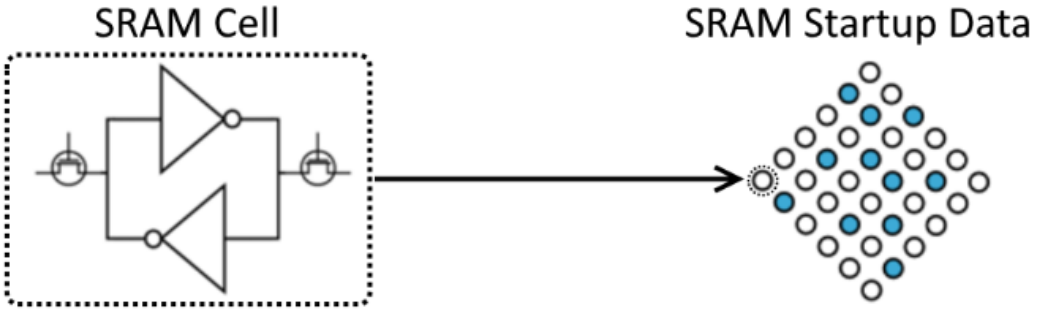


Рис. 3.2. Сукупність SRAM-комірок та їх стартових значень, що формують унікальний, але зашумлений SRAM-відбиток

Однак не всі комірки поводяться однаково стабільно. Частина з них може змінювати свій стан від циклу до циклу. Це пояснюється фізичними відмінностями транзисторів, з яких складається пам’ять: різницею у порогових напругах «плечей» комірки, чутливістю до коливань живлення, впливом температури та ефектами старіння кремнію. Чим меншою є ця різниця, тим більше схильна комірка до коливань і, відповідно, до нестабільності початкових значень. Цей ефект докладно описано у технічному документі *White Paper «The reliability of SRAM PUF»*.

Використовувати такий «сирий» відбиток напряму неможливо, адже він зашумлений. Для того щоб забезпечити надійність і відтворюваність результатів, застосовуються спеціальні алгоритми корекції помилок (Error Correction Codes, ECC). Завдяки їм зашумлений відбиток перетворюється на два основні компоненти:

- **Digital Fingerprint** — відтворюваний цифровий відбиток, унікальний для конкретної мікросхеми;
- **Helper Data** — допоміжна інформація, яка не містить секретів і використовується виключно для того, щоб відновити точний відбиток із «зашумлених» даних.

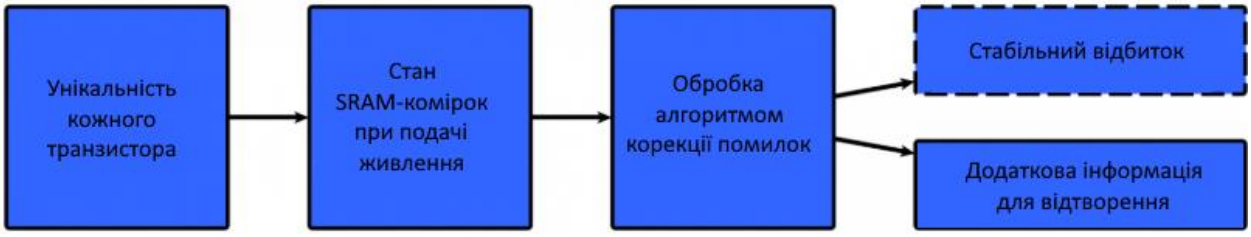


Рис. 3.3 Процес формування основних складових SRAM PUF — Digital Fingerprint та Helper Data

Принцип роботи такий: при кожному включенні пристрою формується набір SRAM Startup Data, який у поєднанні з Helper Data дає можливість коректно відтворити Digital Fingerprint. Цей цифровий відбиток є абсолютно унікальним для кожної мікросхеми, і головне — він ніколи не покидає меж PUF-блоку та не зберігається після вимкнення живлення. Таким чином, ключ існує лише в момент роботи пристрою та відновлюється лише тоді, коли це потрібно.

Digital Fingerprint може виконувати роль кореневого криптографічного ключа, на основі якого формуються симетричні та асиметричні ключі, цифрові сертифікати або інші конфіденційні дані. Це забезпечує високий рівень безпеки, адже навіть у разі фізичного доступу до обладнання зловмисник не зможе відтворити унікальний відбиток.

Завдяки цим властивостям SRAM PUF активно застосовується у сфері телекомунікацій та Інтернету речей (IoT). Наприклад, у мережах 5G і майбутніх 6G технологія дозволяє перевіряти справжність базових станцій, маршрутизаторів та абонентських пристроїв, захищаючи їх від клонування. У IoT ця технологія особливо важлива, адже мільярди сенсорів і контролерів повинні мати унікальну «біометрію», яка підтверджує їхню легітимність у мережі.

Використання SRAM PUF у контролерах

У сучасних мікроконтролерах компанії NXP Semiconductors технологія SRAM PUF реалізована на основі інтелектуальної власності (IP-рішення) від компанії Intrinsic ID, яка є світовим лідером у цій галузі. На сьогодні підтримка PUF-блоку доступна у контролерах серій LPC5400 (ядро Cortex-M4) та LPC5500 (ядро Cortex-M33). Це дає змогу інтегрувати високорівневі механізми апаратної безпеки без необхідності додаткових зовнішніх модулів захисту.

Для практичного розгляду принципів роботи та функціоналу SRAM PUF найчастіше наводять приклад мікроконтролерів сімейства LPC55Sxx, які детально описані у технічному документі AN12324. Саме на цій платформі можна прослідкувати, як функціонує PUF-блок і які методи він підтримує.

Основні методи роботи з PUF-блоком

1. **Enroll** — генерація коду активації (*Activation Code*).
2. **Start** — відтворення цифрового відбитка (*Digital Fingerprint*).
3. **SetKey** — завдання користувацького ключа.
4. **GenerateKey** — генерація випадкового користувацького ключа.
5. **GetKey** — отримання користувацького ключа для подальшого використання.

Кожна з цих команд виконує важливу роль у побудові системи безпеки. Вони дозволяють створювати унікальні ідентифікатори для пристрою, генерувати ключі шифрування та гарантувати, що секретні дані залишаються захищеними навіть у випадку фізичного доступу до обладнання.

Метод Enroll — створення Activation Code

Першим і найбільш важливим кроком у роботі з PUF є виконання операції **Enroll**. Її необхідно запускати лише один раз для кожного пристрою, оскільки саме на цьому етапі формується унікальний «цифровий відбиток» мікросхеми. У результаті роботи алгоритму отримуються два об’єкти:

- **Digital Fingerprint** — унікальний 256-бітний ключ, сформований на основі початкових станів комірок SRAM;
- **Activation Code (Helper Data)** — допоміжні дані, необхідні для подальшого відтворення цифрового відбитка.

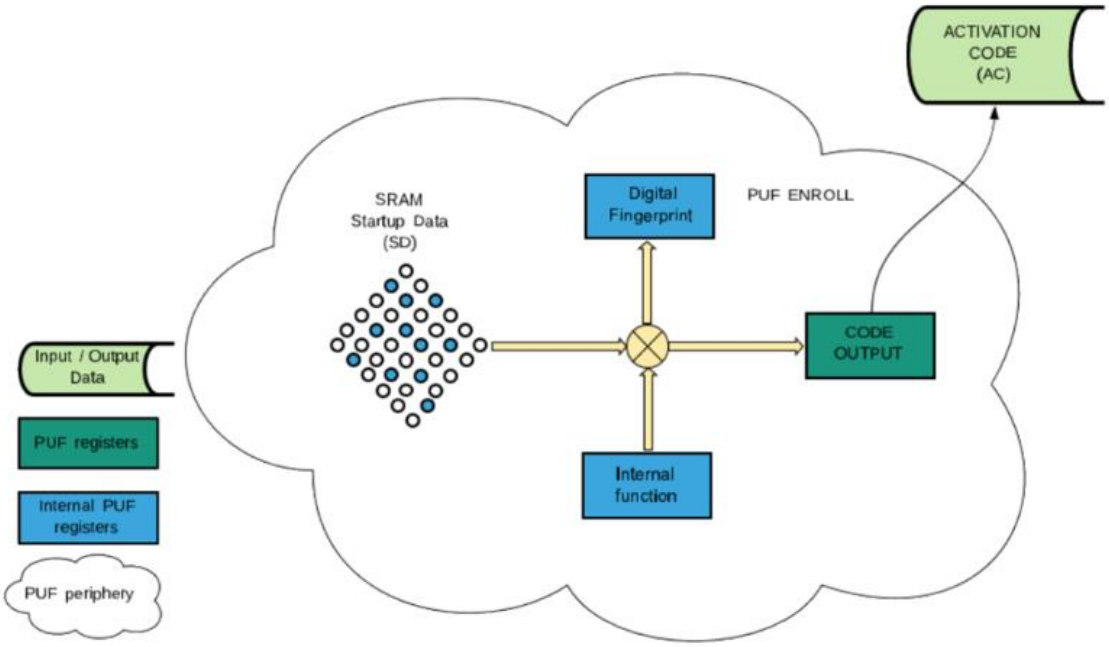


Рис. 4. Метод Enroll: формування Digital Fingerprint і Activation Code

Особливість полягає в тому, що Digital Fingerprint ніколи не залишає меж PUF-блоку й не зберігається після завершення роботи. Натомість Activation Code можна безпечно розмістити у будь-якій енергонезалежній пам’яті (NVM), адже він не містить секретної інформації й сам по собі не дозволяє відновити ключ.

Варто враховувати, що при кожному новому виконанні команди Enroll генеруються інші значення Digital Fingerprint і Activation Code. Це означає, що у разі втрати збереженого Activation Code відновити оригінальний цифровий відбиток буде неможливо. Як наслідок, усі секрети, які базувалися на цьому відбитку, також стануть недоступними. Саме з цієї причини у контролерах передбачено можливість відключення команди Enroll за допомогою спеціального «ф’юза» (fuse), щоб уникнути випадкової або зловмисної перегенерації.

Ще однією важливою перевагою є те, що копіювання Activation Code на інший пристрій не має сенсу. Через відмінності у фізичних характеристиках мікросхем навіть мінімальні зміни у SRAM Startup Data призводять до формування зовсім іншого Digital Fingerprint. Відповідно, ключі чи секрети, пов’язані з оригінальним відбитком, не зможуть бути відновлені на іншому пристрої. Це забезпечує додатковий рівень захисту від клонування обладнання та несанкціонованого використання.

PUF Start – механізм відтворення Digital Fingerprint

Після того як для пристрою виконано операцію Enroll і збережено відповідний Activation Code, технологія SRAM PUF може застосовуватися для захисту ключів і конфіденційних даних. Сам процес починається з того, що живлення для PUF-блоку повністю скидається, а далі виконується команда Start.

У момент запуску в PUF завантажується Activation Code, після чого формується новий набір початкових значень пам'яті SRAM Startup Data. Оскільки ця послідовність не є стабільною і може дещо змінюватися при кожному циклі живлення, спеціальний алгоритм корекції помилок обробляє ці дані та відтворює унікальний Digital Fingerprint, що ідентичний оригінальному. Саме після цього стає можливим виконання наступних операцій: SetKey (встановлення користувацького ключа), GenerateKey (генерація випадкового ключа) та GetKey (отримання ключа для подальшого використання).

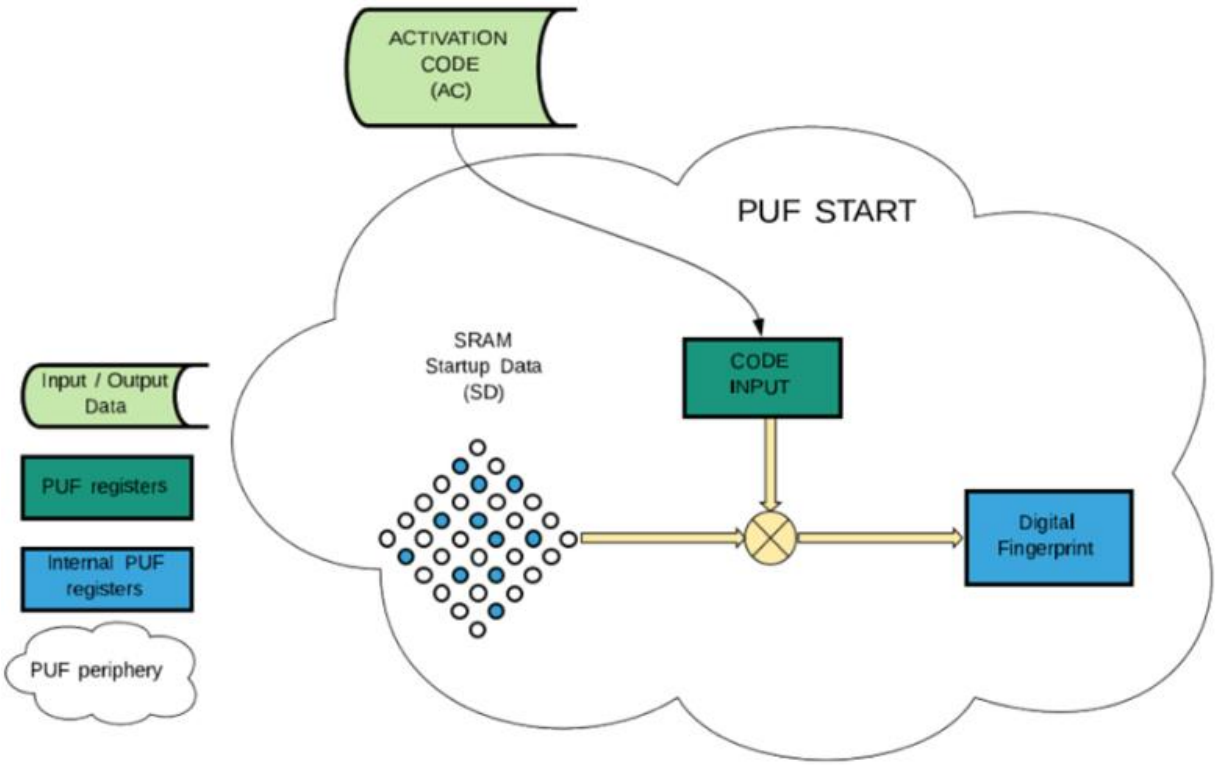


Рис. 3.5. Метод Start: відтворення Digital Fingerprint за допомогою Activation Code

Як працює корекція помилок (приклад Fuzzy Extractor)

Щоб краще зрозуміти принцип роботи, розглянемо спрощений приклад з алгоритмом, що реалізує корекцію помилок у процесі відновлення цифрового відбитка.

Припустимо, що Digital Fingerprint = SRAM PUF Response = SRAM Startup Data. Для прикладу візьмемо відповідь лише з 1 байта. У системі є два можливих кодових слова (*Code Word*), які використовуються для визначення правильного значення:

- C0 = 00000000
- C1 = 11111111

На етапі Enroll пристрій отримує конкретне значення SRAM PUF Response. Далі обирається випадкове кодове слово (наприклад, C0 або C1), і за допомогою операції XOR з відповіддю формується Helper Data (Activation Code). Саме цей набір і зберігається для подальшого використання.

Коли ж виконується команда Start, система знову отримує SRAM PUF Response, проте тепер він може відрізнятися від попереднього через нестабільність окремих комірок. Виконується операція XOR нового значення з Helper Data, після чого аналізується результат. Якщо у результаті більшість бітів збігається з C0, система робить висновок, що правильним є саме це кодове слово; якщо ж із C1 — то воно.

Таким чином, навіть якщо у відповіді виникнуть кілька помилкових бітів (у даному прикладі алгоритм дозволяє виправити до трьох), можна достовірно відновити оригінальний Digital Fingerprint. Заключний етап — це повторна операція XOR між обраним кодовим словом і Helper Data, яка дозволяє реконструювати початковий відбиток.

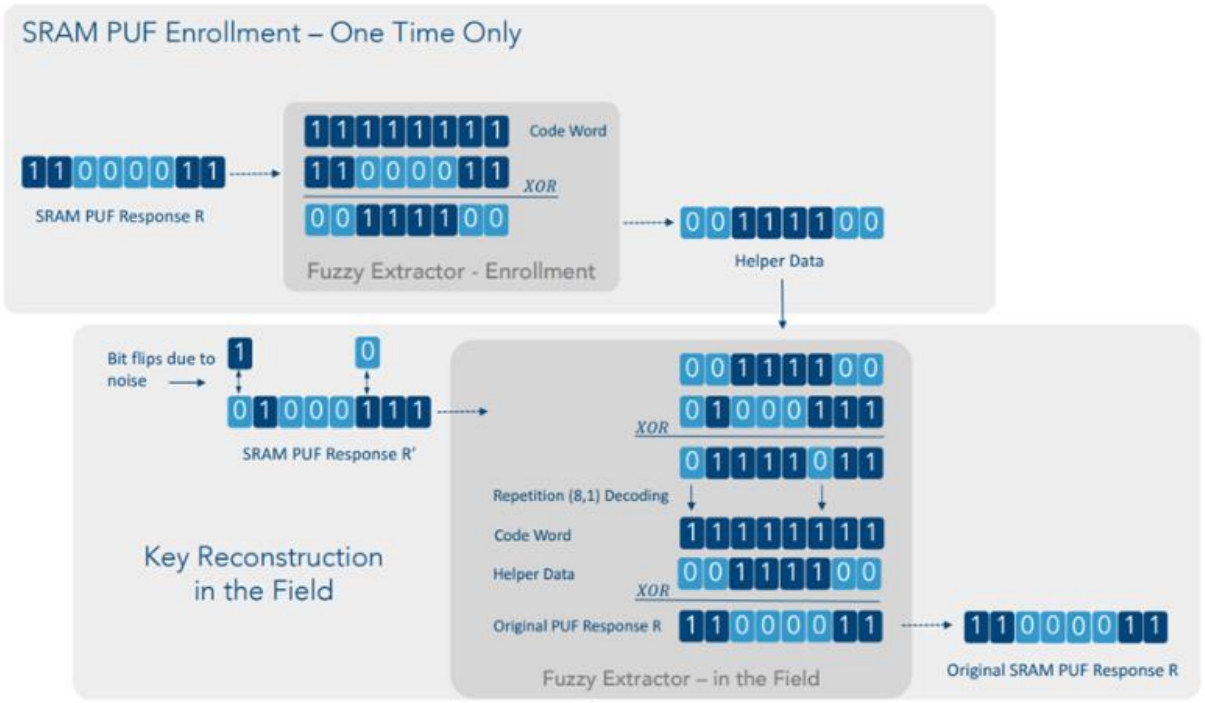


Рис. 3.6. Відновлення оригінального Digital Fingerprint за допомогою Helper Data

Практичне значення методу Start

Метод Start є одним із ключових механізмів у роботі PUF. Він забезпечує надійність та відтворюваність цифрового відбитка навіть у тих випадках, коли «сирі» дані з пам’яті містять похибки. Завдяки цьому будь-який пристрій отримує унікальний, захищений ідентифікатор, який використовується для криптографічних операцій.

Для сучасних телекомунікаційних систем це має особливе значення. Уявімо, що в мережі 5G/6G чи у системі IoT підключено мільйони пристроїв. Кожен із них повинен мати унікальний «паспорт», який неможливо підробити. Саме PUF Start дозволяє отримувати стабільний цифровий відбиток, який може виступати кореневим ключем для захисту трафіку, автентифікації пристрою чи підтвердження довіри на рівні мережевої інфраструктури.

Таким чином, метод Start перетворює випадкові й нестабільні дані пам’яті на стабільний і захищений відбиток, що є базовою основою всієї системи апаратної довіри.

3.2 Робота з ключами користувачів

PUF SetKey – перетворення та захист користувацького ключа

Команда SetKey призначена для того, щоб перетворити введений користувачем ключ (User Key) у спеціальний захищений код — Key Code. Такий код є безпечною послідовністю даних, яку можна зберігати у будь-якому типі пам’яті або навіть передавати разом із прошивкою без ризику розкриття секретної інформації.

Довжина користувацького ключа може змінюватися у дуже широкому діапазоні — від 64 біт до 4096 біт, що дозволяє використовувати цю команду як для легких криптографічних завдань, так і для складних систем безпеки, де потрібні довгі ключі.

Ключовою особливістю є те, що кожен Key Code жорстко прив’язаний до конкретної мікросхеми. Це відбувається завдяки використанню унікального Digital Fingerprint, сформованого PUF-блоком. Відповідно, навіть якщо скопіювати Key Code на інший пристрій, він не матиме жодної цінності, адже відтворити з нього оригінальний ключ можна лише на мікроконтролері, де він був згенерований.

Кожен Key Code отримує свій індекс. При цьому варто звернути увагу на особливий випадок: Key Code з індексом 0. Він не може бути доступним для програмної частини, однак може безпосередньо використовуватися апаратними криптографічними блоками мікроконтролера. Це означає, що ключ з індексом 0 ідеально підходить для інтеграції в AES-блок, який може застосовуватися для шифрування та захисту прошивки пристрою. У такому випадку забезпечується максимальна безпека, оскільки ключ ніколи не з’являється у відкритому вигляді.

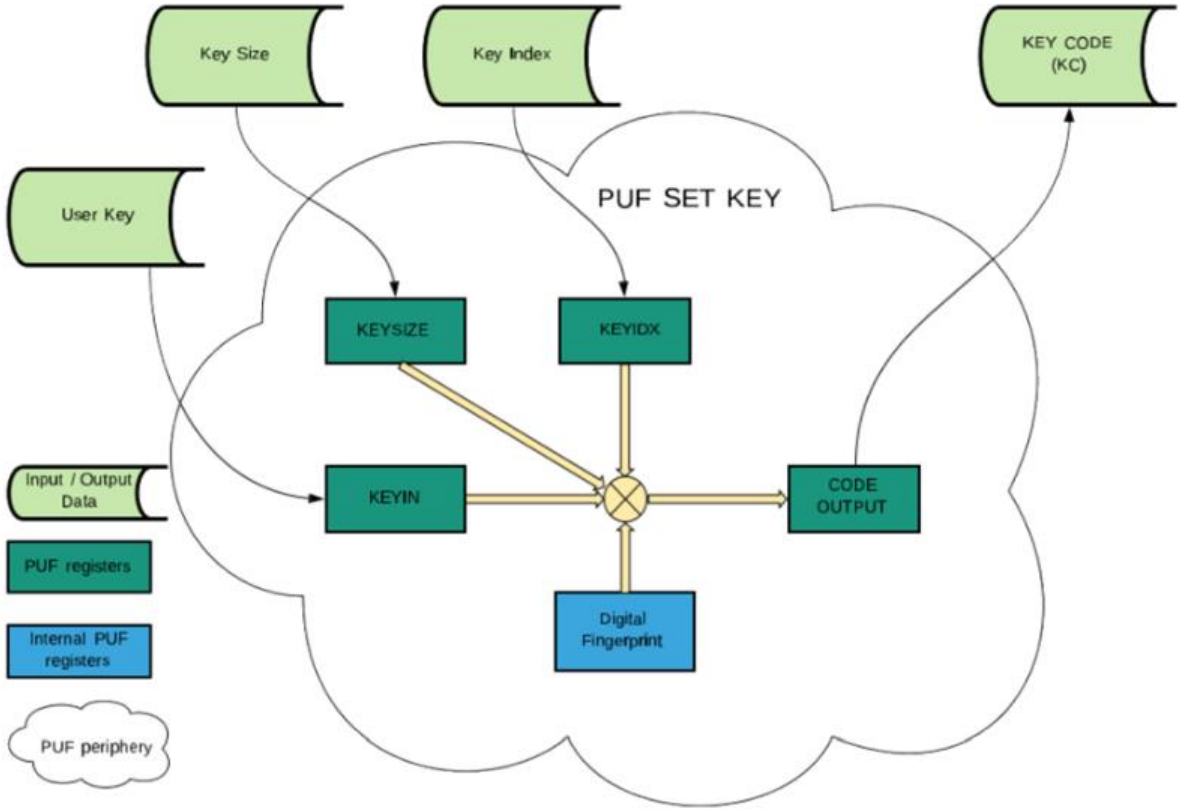


Рис. 3.7. Метод SetKey: процес захисту користувацького ключа за допомогою PUF

PUF GenerateKey – створення випадкового ключа

PUF-блок також підтримує можливість генерації випадкових ключів безпосередньо всередині пристрою. Це реалізується командою GenerateKey, яка формує випадкове значення, а потім автоматично перетворює його у Key Code. Таким чином, ключ ніколи не існує поза межами захищеного PUF-блоку у незашифрованому вигляді, що гарантує найвищий рівень безпеки.

Головна перевага цього підходу полягає в тому, що ключ генерується апаратно, з використанням фізично неклонованих характеристик мікросхеми, а отже, стає унікальним і абсолютно непередбачуваним. Такий метод є оптимальним для сценаріїв, де потрібні короткочасні криптографічні ключі, наприклад у системах автентифікації сеансів у телекомунікаційних мережах чи у протоколах захисту IoT-пристроїв.

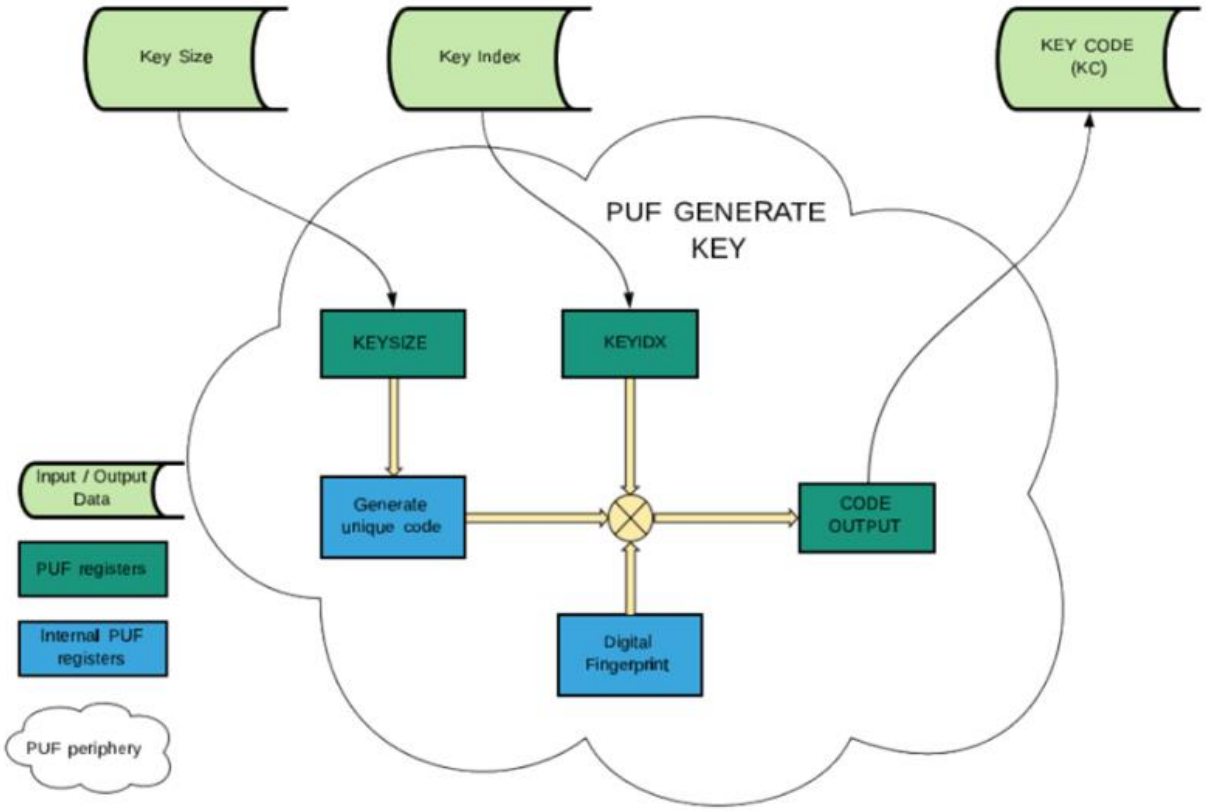


Рис. 3.8. Метод GenerateKey: генерація випадкового ключа та його подальший захист у PUF-блоці

PUF GetKey – відновлення користувацького ключа

Заключний метод — GetKey — виконує зворотну операцію: він перетворює Key Code назад у вихідний користувацький ключ (User Key) або ж безпосередньо направляє його у криптографічні апаратні модулі. Це дозволяє використовувати ключ у подальших криптографічних операціях, не розкриваючи його вміст програмним засобам.

Як уже згадувалося, Key Code з індексом 0 має особливий статус: він недоступний з боку програмного забезпечення й може бути використаний виключно всередині криптографічних блоків. У мікроконтролерах NXP існує можливість інтеграції ключів не тільки з AES-блоком, але й із ще одним вбудованим алгоритмом — PRINCE. Цей симетричний шифр відрізняється високою швидкістю й створений спеціально для сценаріїв «розгортання прошивки на льоту» (on-the-fly decryption).

Особливо важливо це для мікроконтролерів серії LPC5400, у яких немає власної енергонезалежної пам’яті. Такі пристрої працюють за принципом Execute-In-Place (XIP) — прошивка виконується безпосередньо з зовнішньої пам’яті. Використання PRINCE у поєднанні з методом GetKey дозволяє дешифрувати код у режимі реального часу, не знижуючи продуктивності системи.

Для зручності реалізації у методу GetKey передбачений параметр KeySlot, за допомогою якого розробник може визначити, в який саме криптографічний блок (AES або PRINCE) буде завантажено відновлений ключ. Це забезпечує гнучкість налаштувань і дозволяє інтегрувати захист у найрізноманітніші архітектурні рішення.

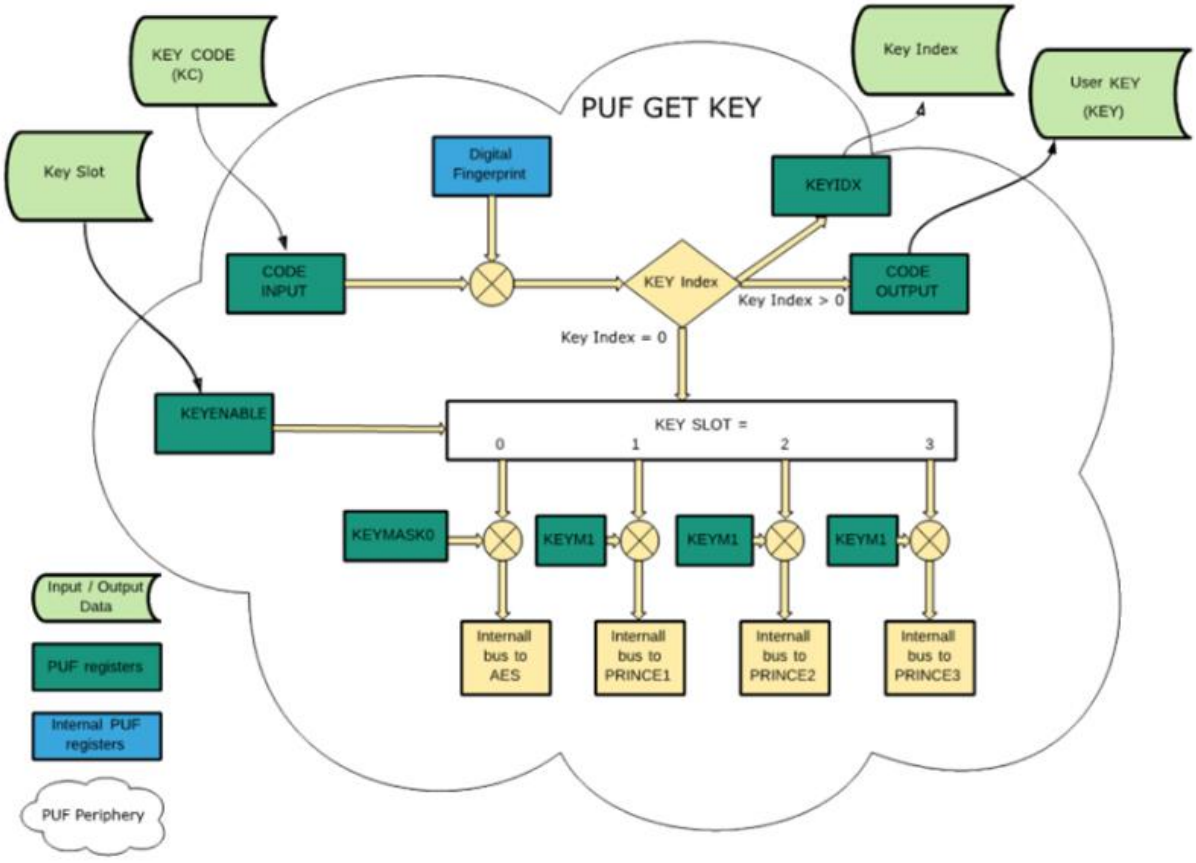


Рис. 3.9. Метод GetKey: процес відновлення користувацького ключа та передача його у криптографічні блоки

3.3 Реалізація SRAM PUF

Сценарії використання SRAM PUF

Технологія SRAM PUF має велике практичне значення в системах безпеки, адже вона дозволяє створювати унікальні апаратні ідентифікатори, які не можна клонувати чи підробити. В офіційній документації AN12292 виокремлюють три основні напрями застосування цієї технології:

- персоналізація пристроїв;
- безпечне завантаження (Secure Boot);
- захист користувацьких ключів і конфіденційних даних.

Персоналізація пристроїв

Персоналізація є ключовим етапом підготовки пристрою до роботи. На цьому етапі PUF-блок проходить процедуру Enroll, яка дозволяє згенерувати унікальний цифровий відбиток та допоміжні дані для подальшої роботи. Для цього використовується спеціальна прошивка — Enrollment Image. Вона виконує кілька завдань:

- тестує працездатність пристрою;
- створює й захищає ключі, необхідні для роботи;
- формує базу ідентифікованих і перевірених пристроїв, які можна вважати автентичними.

Після цього пристрій можна налаштувати так, щоб він працював лише з прошивкою, яка захищена унікальним ключем, згенерованим саме для цього пристрою.

Процес персоналізації може відбуватися у двох середовищах:

1. Довірене середовище (OEM, виробник обладнання). У цьому випадку персоналізація виконується штатними засобами виробника і гарантує повний контроль над процесом.
2. Недовірене середовище (контрактне виробництво). Якщо виробнику не можна довіряти, доводиться будувати складну ланку довіри з використанням

асиметричного шифрування та окремих апаратних модулів безпеки (HSM — Hardware Security Module).

Саме в другому випадку виникає серйозна проблема перевірки достовірності публічного ключа пристрою. Якщо зловмисник виконає атаку «людина посередині» (Man-in-the-Middle), він може підмінити справжній публічний ключ мікроконтролера власним. У результаті прошивка або ключі будуть зашифровані не для оригінального пристрою, а для системи зловмисника.

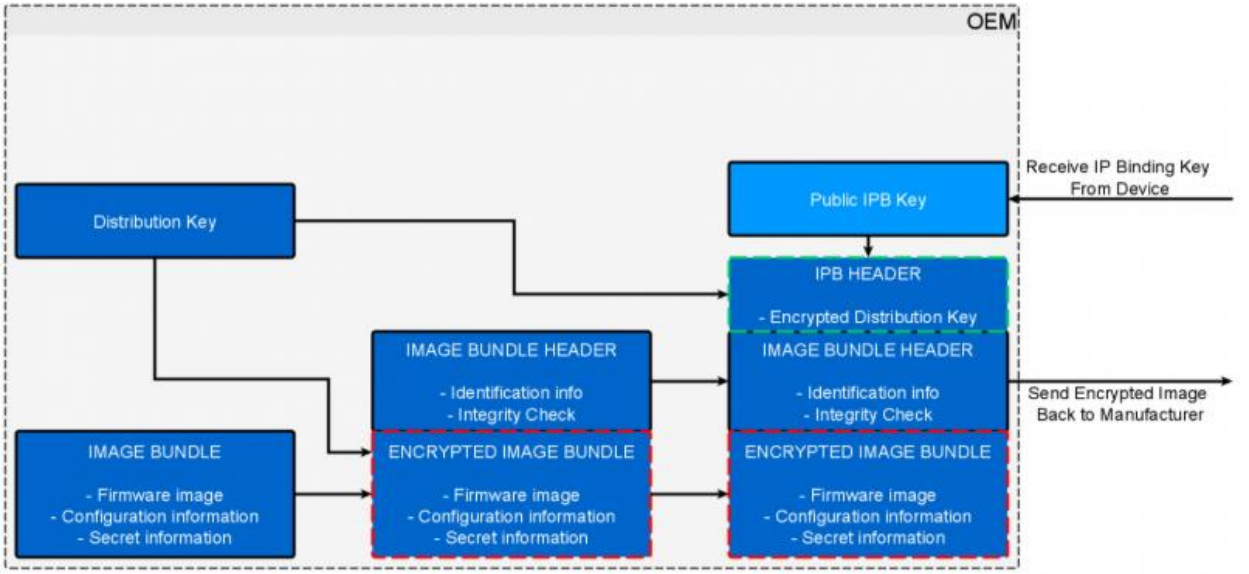


Рис. 3.10 Процес підготовки прошивки для передачі на виробництв

Щоб уникнути таких атак, потрібно підписувати публічні ключі за допомогою довіреного сертифіката виробника (CA). Це гарантує, що ключ дійсно належить пристрою і не був підмінений.

Однак у документі AN12291 немає чіткої відповіді, як саме вирішується ця проблема. Для уточнення компанія NXP Semiconductors рекомендує звертатися безпосередньо до Intrinsic ID — розробників технології. Під час дослідження було знайдено спільний вебінар Intrinsic ID та GlobalSign, де зазначено, що механізм перевірки справжності має інтегруватися ще на рівні виробника мікросхеми.

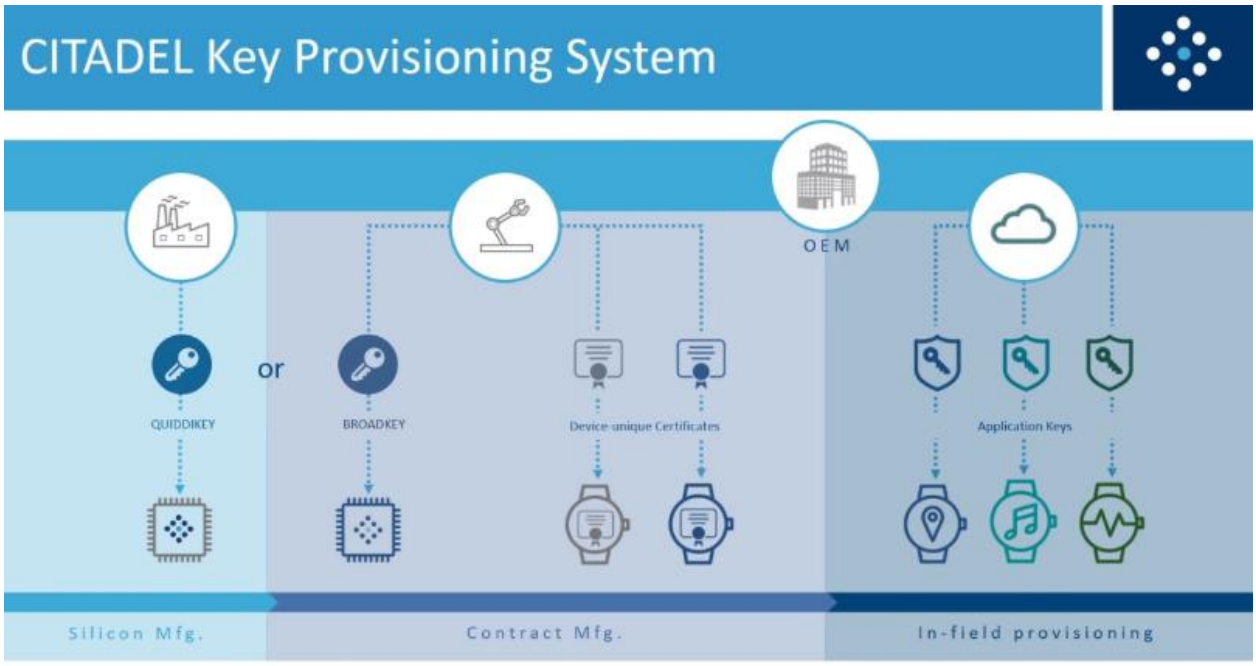


Рис. 3.11 Процес персоналізації: приклад інтеграції сертифікатів на основі SRAM PUF

Цю інформацію підтвердили і представники Intrinsic ID: персоналізація повинна виконуватися або у повністю довіреному середовищі, або мікроконтролери мають пройти процедуру Enroll безпосередньо на фабриці. Для цього використовується приватний ключ сертифікаційного центру (CA private key), що дозволяє формувати надійну базу автентичних пристроїв ще до їх відправлення замовнику.

Чи надає компанія NXP Semiconductors таку послугу офіційно — достеменно невідомо. Однак, імовірно, вона доступна для великих партій виробів, де питання довіри та ідентифікації є критичними. У будь-якому випадку можна зробити висновок: при використанні SRAM PUF у контролерах NXP підготовчий етап Enroll обов’язково повинен виконуватися в довіреному середовищі, щоб уникнути підміни ключів і гарантувати безпечність усієї ланки поставок.

Безпечне завантаження (Secure Boot)

Одним із найбільш важливих сценаріїв використання SRAM PUF у мікроконтролерах NXP Semiconductors є механізм Secure Boot. Його головна мета — гарантувати, що під час увімкнення пристрою запускається виключно перевірене та справжнє програмне забезпечення, яке не було змінене сторонніми особами. Це

критично для телекомунікаційних і промислових систем, де підміна прошивки може призвести до серйозних збоїв або витоку конфіденційних даних

Як зазначалося раніше, у контролерах NXP інтегровані потужні апаратні криптомодулі — AES і PRINCE, які безпосередньо працюють із Key Code. Особливість полягає в тому, що ключ із нульовим індексом передається виключно в ці криптографічні блоки через спеціалізовану внутрішню шину й ніколи не доступний програмним засобам.

Процес реалізації Secure Boot виглядає так:

1. Розробник створює симетричний ключ, який виступатиме в ролі захисного.
2. Прошивка шифрується цим ключем, після чого вона стає недоступною для стороннього аналізу.
3. Симетричний ключ перетворюється на Key Code з індексом 0, що унеможливлює його доступ програмною частиною.

У підсумку ключ існує лише у зашифрованій формі й може бути використаний виключно криптографічними блоками. Це робить процес завантаження прошивки захищеним від підробок, перехоплень і несанкціонованого доступу навіть у випадку, якщо атакувальник має фізичний доступ до обладнання. Такий підхід дозволяє інтегрувати механізм апаратної довіри на базовому рівні роботи пристрою.

Захист користувацьких ключів і даних

Інший не менш важливий сценарій використання SRAM PUF — це захист користувацьких ключів і чутливих даних під час роботи пристрою. Завдяки можливості перетворювати дані у Key Code, будь-яку конфіденційну інформацію можна зробити доступною лише для того пристрою, на якому вона була згенерована.

Це відкриває широкі можливості:

- симетричні ключі можна зберігати у вигляді Key Code, а розгортати їх лише під час шифрування чи розшифрування даних;
- асиметричні ключі можна зберігати частково: публічний ключ — у відкритому вигляді, а приватний — у вигляді Key Code. Це дозволяє

відновлювати його лише тоді, коли потрібно виконати підпис або дешифрування.

Для генерації пар асиметричних ключів у мікроконтролерах NXP передбачено спеціальний апаратний модуль CASPER, оптимізований для виконання складних криптографічних обчислень із високою швидкістю. Це робить систему більш гнучкою і дозволяє застосовувати PUF не лише для симетричного шифрування, а й у більш складних криптографічних протоколах.

Єдине технічне обмеження, яке потрібно враховувати: максимальний розмір User Key становить 512 байт. Проте цього цілком достатньо для більшості сучасних сценаріїв захисту, включаючи зберігання приватних ключів і паролів доступу.

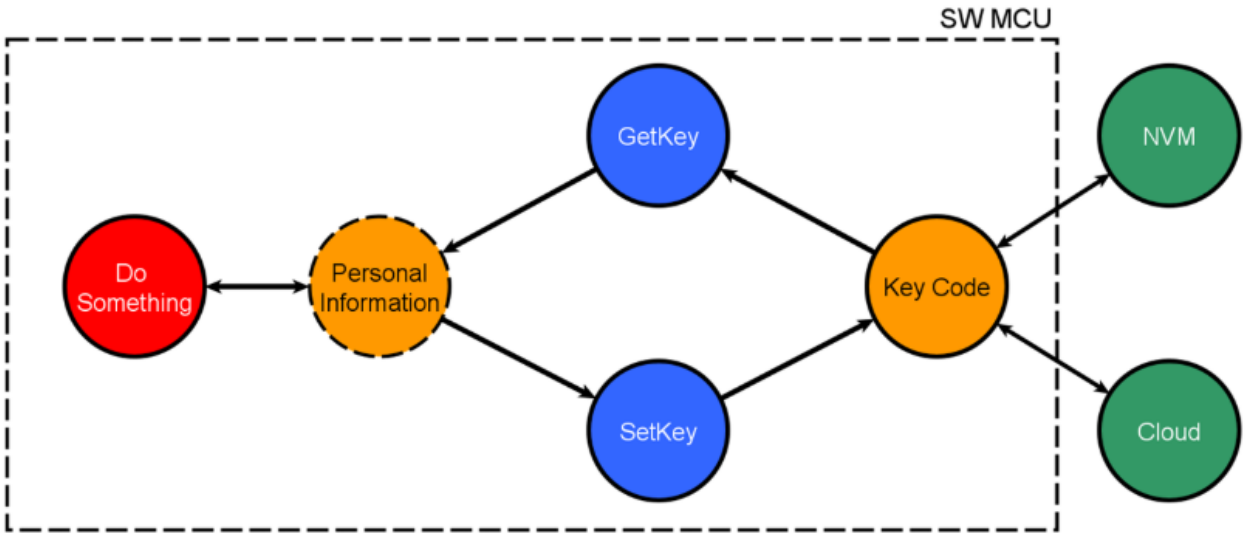


Рис. 3.12 Сценарій використання SRAM PUF для захисту персональних даних і ключів

Які проблеми властиві SRAM PUF?

Хоча технологія SRAM PUF вважається одним із найбільш перспективних рішень у сфері апаратної безпеки, вона не позбавлена низки проблем і потенційних загроз, які необхідно враховувати при її використанні в критично важливих системах, особливо у телекомунікаціях та IoT.

Старіння кремнію як фундаментальний виклик

Головна проблема, із якою стикаються всі пристрої на основі SRAM PUF, — це поступове старіння кремнію, яке з часом призводить до деградації характеристик транзисторів у пам’яті. На цей процес впливають кілька ключових фізичних явищ:

- **NBTI (Negative-Bias Temperature Instability)** — температурна нестабільність переходу, яка викликає зміну порогових напруг у транзисторах;
- **HCI (Hot Carrier Injection)** — інжекція «гарячих носіїв», що виникає внаслідок високих електричних полів і пошкоджує структуру транзистора;
- **TDDB (Time-Dependent Dielectric Breakdown)** — поступовий пробій діелектрика затвора, що з часом призводить до втрати його ізоляційних властивостей.

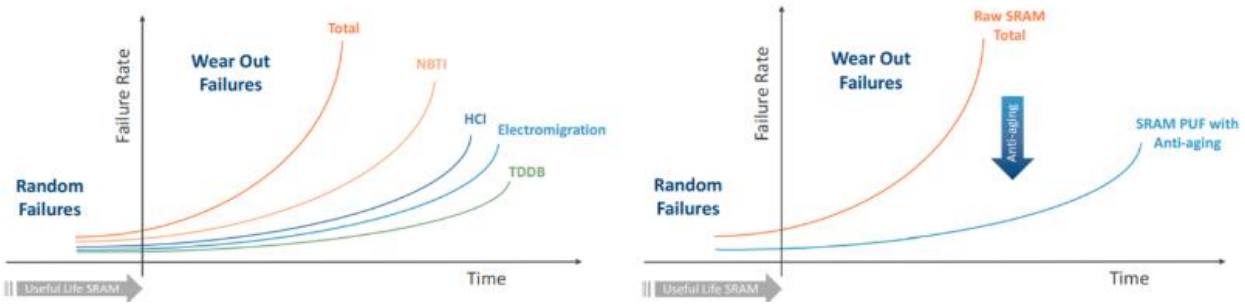


Рис. 3.13 Фактори, що впливають на тривалість життєвого циклу SRAM PUF

Ці явища безпосередньо впливають на стабільність стартових станів пам’яті, від яких залежить точність відтворення Digital Fingerprint. Якщо зміни стають надто суттєвими, пристрій може втратити здатність відтворювати однаковий криптографічний відбиток, а це ставить під загрозу всю систему безпеки.

Для мінімізації цих ризиків виробники впроваджують так звані «антистаріючі» методики, описані у *White Paper «The reliability of SRAM PUF»*. Вони дозволяють компенсувати вплив негативних факторів і суттєво продовжити строк служби пристрою. За наявними оцінками, за умови комплексного використання таких технологій мікросхеми з PUF-блоком можуть функціонувати стабільно не менше 25 років, що відповідає типовому життєвому циклу промислових і телекомунікаційних систем.

Криптографічні загрози та машинне навчання

Інший потенційний виклик пов’язаний не стільки з фізикою, скільки з розвитком інтелектуальних технологій. У майбутньому методи машинного навчання можуть бути використані для проведення криптографічних атак на PUF.

Якщо атакувальник зможе зібрати достатній обсяг даних про реакції конкретного пристрою, то існує ймовірність побудови математичної моделі, здатної імітувати поведінку цього PUF-блоку.

Подібні ризики вже підтверджені експериментально. Наприклад, на конференції РусКрипто 2020 компанія *Криптонит* представила дослідження, в якому зазначалося, що прості реалізації PUF можуть бути скомпрометовані буквально за кілька годин або навіть хвилин.

Хоча наразі немає відомих успішних атак на комерційні реалізації від Intrinsic ID (ключового гравця у сфері PUF), це не гарантує абсолютної безпеки. З розвитком обчислювальних потужностей та штучного інтелекту такі атаки можуть стати практично здійсненними. Особливо небезпечними вони можуть бути у великих телекомунікаційних мережах, де сотні тисяч пристроїв генерують величезний масив статистичних даних, доступних для аналізу.

4 ЗАХИСТ НА РІВНІ BACKEND

4.1 Проблематика на рівні Backend

У сучасному цифровому просторі, де веб-додатки стали невід'ємною частиною щоденної діяльності мільйонів користувачів, питання безпеки інформації та функціоналу систем набуває критично важливого значення. Стрімкий розвиток технологій, таких як віртуалізація сервісів, мобільні технології, хмарні обчислення та інтернет речей (IoT), зумовив значне зростання кількості веб-додатків і перетворив їх на ключові інструменти для бізнесу, державних структур і кінцевих споживачів. Водночас із новими можливостями з'явилися й нові загрози, що ставлять під сумнів конфіденційність, цілісність та доступність даних, а також стабільність бізнес-процесів.

Конфіденційність у сфері інформаційної безпеки означає такий стан даних, за якого доступ до них мають лише визначені особи або групи осіб, яким ці дані довірено. Це досягається через комплекс заходів, серед яких обмеження доступу, захист від несанкціонованого втручання, використання криптографічних методів (зокрема шифрування під час зберігання та передавання), управління правами доступу користувачів, а також постійне навчання персоналу. Важливу роль відіграє і фізична безпека: доступ до приміщень, де зберігаються або обробляються дані, має контролюватися замками, системами ідентифікації та іншими засобами.

Не менш важливою є цілісність даних. Вона означає, що інформація має залишатися незмінною та достовірною протягом усього життєвого циклу — від створення й обробки до зберігання та передачі. Порушення цілісності може призвести до фатальних наслідків: помилкових рішень, спотворення інформації, втрати критично важливих бізнес-процесів. Для гарантування цілісності застосовуються такі методи, як хеш-функції, цифрові підписи, контрольні суми, а також механізми контролю версій і аудит, які дозволяють відстежувати всі зміни та запобігати маніпуляціям.

Третім базовим принципом є доступність. Вона означає, що інформація та сервіси повинні бути доступними користувачам у будь-який момент, коли це необхідно, без затримок і збоїв. Втрата доступності може призвести до серйозних наслідків: фінансових збитків, втрати клієнтів, погіршення репутації компанії. Для забезпечення цього принципу впроваджуються системи резервного копіювання й відновлення, відмовостійкі архітектури, балансування навантаження та моніторинг продуктивності. Особливу увагу приділяють захисту від атак типу DDoS, які можуть вивести з ладу цілі сервіси.

Дані в контексті інформаційних технологій розглядаються як фрагменти інформації, що підлягають запису, зберіганню, обробці й передачі. Вони можуть мати різні формати та ступінь чутливості. Наприклад, особисті дані користувачів потребують максимальної захищеності, тоді як технічні журнали чи службові записи можуть бути менш критичними. З огляду на те, що дані є основою аналітики, штучного інтелекту та машинного навчання, їх захист стає стратегічним завданням.

Поняття захисту охоплює весь комплекс заходів, спрямованих на забезпечення безпеки, стабільності та збереження даних і систем. Серед основних етапів — попередження загроз, їх виявлення, своєчасне реагування та відновлення після інцидентів. Попередження включає впровадження політик безпеки, навчання персоналу, застосування сучасних технологій моніторингу. Виявлення загроз відбувається за допомогою систем аналізу поведінки, журналювання й аудиту. Реагування охоплює швидке усунення виявлених уразливостей, а відновлення передбачає повернення систем до нормального стану, використання резервних копій і детальний аналіз причин інцидентів.

Важливо відрізняти поняття «загроза» та «атака». Загроза — це потенційний чинник, який може завдати шкоди системі або даним, тоді як атака — це вже конкретна дія, спрямована на реалізацію цієї загрози. Загрози можуть бути внутрішніми (помилки персоналу, технічні збої) і зовнішніми (хакерські атаки, DDoS, природні катастрофи). Вони класифікуються за напрямами: загрози конфіденційності, цілісності, доступності, автентичності та авторизації. Атаки ж

реалізуються у вигляді зламу паролів, зараження систем шкідливим ПЗ, підробки даних або блокування доступу до сервісів.

Окремої уваги заслуговує backend веб-додатків — серверна частина, яка відповідає за обробку запитів користувачів, взаємодію з базами даних, реалізацію бізнес-логіки, автентифікацію та авторизацію, а також за захист даних. Backend може надавати API для взаємодії з іншими сервісами, виконувати перевірку даних, відстежувати підозрілу активність і реалізовувати заходи безпеки. Від його стабільності й захищеності залежить безперебійна робота усього додатка.

Захист веб-додатків — це не лише технічний обов'язок, а й стратегічна потреба бізнесу. Невчасне виявлення уразливостей, втрата даних чи збій у роботі сервісів можуть призвести до втрати клієнтів, падіння довіри й репутаційних ризиків. Саме тому захист має охоплювати як технічні аспекти (шифрування, контроль доступу, моніторинг), так і організаційні (політики безпеки, аудит, навчання персоналу).

4.2 Основні загрози

SQL-ін'єкції (*SQL Injection*)

Веб-додатки активно використовують бази даних для зберігання та обробки інформації, а користувачі взаємодіють із цими даними через різноманітні інтерфейси — від форм входу до сторінок відображення результатів пошуку. У типовому сценарії автентифікації користувач вводить логін та пароль, після чого сервер перевіряє їх у базі даних і визначає рівень доступу. Проте небезпека виникає тоді, коли розробники нехтують перевіркою коректності введених даних. У такому разі будь-яке значення, введене користувачем у форму, може бути інтерпретоване сервером не лише як дані, а й як частина SQL-запиту. Цим і користуються зловмисники, впроваджуючи вразливість у вигляді SQL-ін'єкцій.

Приклад:

Сервіс авторизації *auther* працює з базою *users*, що має поля *username* і *password*. Для входу користувач вводить:

Login: admin

Password: password

Стандартний запит до бази виглядатиме так:

SELECT * FROM users WHERE username='admin' AND password='password';

Якщо ж зловмисник замість пароля введе рядок 'password' OR '1'='1', то запит набуде вигляду:

SELECT * FROM users WHERE username='admin' AND password='password' OR '1'='1';

Оскільки вираз '1'='1' завжди істинний, система дозволить авторизацію без справжнього пароля. Це надає змогу увійти в систему під обліковим записом адміністратора, отримати доступ до критичних даних і навіть змінювати їх.

Основні причини вразливості:

- відсутність валідації та фільтрації введених даних;
- недостатнє екранування символів у SQL-запитах;
- розкриття допоміжної інформації через системні повідомлення про помилки, які можуть слугувати підказкою для складання робочого шкідливого запиту.

XSS (*Cross-Site Scripting* — міжсайтовий скриптинг)

Уразливість типу XSS дозволяє впровадити шкідливий код (переважно JavaScript) у веб-сторінку, яка потім виконується у браузерах інших користувачів. Це створює серйозну загрозу для безпеки, оскільки код виконується у довіреному контексті веб-додатка, а отже, має доступ до cookies, локального сховища та інших конфіденційних даних користувача.

Види XSS-атак:

- **Рефлектований (Reflected XSS)**. Шкідливий код міститься в URL-посиланні й відображається сервером у відповіді користувачеві.
- **DOM-based XSS**. Код виконується на клієнтській стороні через маніпуляцію елементами DOM у браузері.
- **Збережений (Stored XSS)**. Вразливий скрипт зберігається на сервері й виконується кожного разу, коли користувач відкриває сторінку.

Приклад:

У формі авторизації сервісу *auther* відсутня фільтрація введених даних. Якщо замість логіна користувач введе:

<script>alert('XSS attack');</script>

то цей код буде впроваджений у HTML і виконаний у браузері жертви. У реальному сценарії це може призвести не лише до появи повідомлення, а й до викрадення cookies сесії, перенаправлення користувача на шкідливий сайт або завантаження небезпечного контенту.

Переповнення буфера (*Buffer Overflow*)

Переповнення буфера виникає тоді, коли програма намагається записати в буфер більше даних, ніж йому відведено. Це може призвести до перезапису пам’яті, некоректної роботи програми та створює можливості для виконання довільного коду.

Наслідки:

- **Відмова в обслуговуванні (DoS)**. Переповнення призводить до збою і система припиняє обробку запитів.
- **Повний контроль над сервером**. Якщо зловмиснику вдається впровадити виконуваний код, він може здобути доступ адміністратора до всієї системи.

Приклад:

Сервіс *keySaver* використовує хеш-таблицю для зберігання паролів довжиною до 10 байт. Функція збереження не перевіряє довжину введеного рядка:

```
function savePassword(password) {
   passwordTable = allocateHashTable(10);
   copyData(password, passwordTable);
}
```

Якщо замість коректного пароля користувач введе рядок довжиною понад 10 байт, наприклад "AAAAAAAAAAcodecode", надлишкові символи перезапишуть пам’ять за межами виділеного простору. Це може викликати збій або дозволити виконати шкідливий код.

CSRF (*Cross-Site Request Forgery* — міжсайтове підроблення запиту)

CSRF-атака змушує автентифікованого користувача виконати небажані дії на веб-сайті, на якому він уже авторизований. Така уразливість особливо небезпечна, якщо користувач має підвищені права доступу.

Можливі наслідки:

- зміна особистих даних (пароль, email);
- виконання небажаних транзакцій;
- використання адміністративних прав для видалення чи модифікації критичних даних.

Приклад:

Уявімо інтернет-магазин *buyOnline*, де є форма для зміни пароля:

```
<form method="post" action="http://victim-site.com/change_password">
   <input type="password" name="new_password">
</form>
```

Зловмисник створює свій сайт із прихованою формою:

```
<form action="http://victim-site.com/change_password" method="post">
   <input type="hidden" name="new_password" value="hacked_password">
   <input type="submit" value="Отримати приз">
</form>
```

Якщо авторизований користувач відвідає цей сайт, браузер автоматично відправить запит до *buyOnline*, і його пароль буде змінено на hacked_password. Користувач навіть не усвідомить, що відбулася атака, оскільки все виглядатиме як звичайна взаємодія із сайтом.

4.3 Можливі рішення

З огляду на багатогранність загроз, що супроводжують сучасні веб-додатки, розробка систем захисту набуває особливої актуальності. Сьогодні веб-сервіси стали основою бізнес-процесів, електронної комерції, банківських операцій, державних послуг та повсякденного спілкування мільйонів користувачів. У такій ситуації поява нових методів атак та інструментів зловмисників — лише питання

часу. Тому надійний захист має ґрунтуватися на комбінації перевірених практик, сучасних технологій та систематичному аналізі ризиків. Нижче детально розглянемо ключові методи захисту від найпоширеніших атак: SQL-ін’єкцій, XSS, переповнення буфера та CSRF.

Захист від SQL-ін’єкцій

SQL-ін’єкції — одна з найбільш небезпечних та поширених атак на веб-додатки, яка дозволяє отримати несанкціонований доступ до бази даних, модифікувати її вміст або навіть повністю видаляти таблиці. Сучасні методи протидії ґрунтуються на багаторівневому підході.

1. **Параметризовані запити (Prepared Statements).**

   Це базова й водночас найбільш ефективна техніка. Значення, введені користувачем, передаються як окремі параметри, а не як частина SQL-запиту. Це виключає можливість того, що зловмисник впровадить свій код.

   Приклад:

   cursor.execute("SELECT * FROM users WHERE user_id=?", (user_id,))

Навіть якщо ввести шкідливий рядок на кшталт 1; DROP TABLE users;, він буде розпізнаний як дані, а не як команда.

2. **Фільтрація та валідація даних.**

   Вхідні дані мають відповідати чітко визначеним правилам.

- *Білий список* дозволяє лише ті символи чи значення, які відповідають очікуваному формату (наприклад, тільки цифри у полі для ID).
- *Чорний список* забороняє небезпечні символи (', ;, --), однак менш ефективний, оскільки зловмисники знаходять обхідні шляхи.

3. **Використання ORM (Object-Relational Mapping).**

   ORM-бібліотеки створюють проміжний рівень між програмою та базою даних. Вони автоматично будують запити й мінімізують ризик SQL-ін’єкцій, хоча й вимагають обережності у налаштуванні.

4. **Обмеження прав доступу.** Бази даних повинні мати чітко визначені ролі користувачів. Обліковий запис додатку повинен мати лише ті права, які необхідні для його роботи, без зайвих адміністративних повноважень.

Захист від XSS (*Cross-Site Scripting*)

XSS-атаки небезпечні тим, що дозволяють виконувати шкідливий код у браузері користувача. Наслідками можуть бути викрадення cookies, обхід автентифікації або повне захоплення сесії.

1. **Екранування введених даних.**

   Будь-які дані, введені користувачем, мають бути оброблені перед відображенням на сторінці. Символи <, >, ", ' замінюються на безпечні аналоги, що унеможливлює їх інтерпретацію як HTML чи JavaScript.

2. **Встановлення заголовків безпеки.**

- *Content-Security-Policy (CSP)* дозволяє визначати, звідки можуть завантажуватися скрипти. Наприклад:
- Content-Security-Policy: default-src 'self'; script-src 'self' https://example.com
- *X-XSS-Protection* може блокувати підозрілу активність на стороні браузера.

3. **Валідація та санітизація даних.**

   Важливо не лише перевіряти дані на клієнтській стороні, а й дублювати ці перевірки на сервері. Це зменшує ризик впровадження шкідливого коду.

4. **Використання безпечних бібліотек та API.**

   Сучасні фреймворки (наприклад, Angular чи React) мають вбудовані механізми захисту від XSS, автоматично екрануючи введені користувачем дані.

Захист від переповнення буфера

Атаки цього типу спрямовані на експлуатацію помилок у роботі з пам’яттю. Вони можуть призвести як до відмови у обслуговуванні, так і до виконання довільного коду зловмисника.

1. **Перевірка довжини даних.**

   Кожна операція запису повинна супроводжуватися перевіркою максимально допустимого розміру буфера.

2. **Використання безпечних функцій.**

   Сучасні мови програмування та бібліотеки містять функції, які автоматично контролюють розмір буфера, зменшуючи ризик помилки.

3. **Механізми операційних систем.**

- *ASLR (Address Space Layout Randomization)* — ускладнює прогнозування розташування областей пам’яті.
- *DEP (Data Execution Prevention)* — блокує виконання коду в сегментах пам’яті, що призначені лише для даних.

4. **Статичний та динамічний аналіз коду.**

   Регулярне тестування програмного забезпечення дозволяє виявляти помилки ще на етапі розробки.

Захист від CSRF (*Cross-Site Request Forgery*)

CSRF-атаки примушують користувача виконувати небажані дії на сайті, де він авторизований.

1. **CSRF-токени.**

   Для кожної сесії генерується унікальний токен, який додається до запитів. Сервер перевіряє відповідність токена, і сторонній зловмисник не може його підробити.

2. **SameSite Cookies.**

   Налаштування cookies з параметрами Strict, Lax або None дозволяє контролювати їхню поведінку під час крос-доменних запитів.

- *Strict* — максимальний захист, куки працюють лише в межах одного домену.
- *Lax* — компромісний варіант із винятками для безпечних запитів.
- *None* — використовується для складних сценаріїв, але лише з HTTPS.

3. **Перевірка заголовка Referer.**

   Сервер може переконатися, що запит був виконаний із довіреного джерела.

4. **Подвійна передача токена (Double Submit Cookie).**

   У запит передається два токени: один зберігається в cookies, інший у полі форми. Якщо вони збігаються — запит визнається легітимним.

Рекомендації та найкращі практики для розробників і адміністраторів телекомунікаційних мереж

Забезпечення безпеки веб-додатків — це багаторівневий процес, що включає як технічні, так і організаційні аспекти. Оскільки сучасні веб-сервіси є критично важливими для бізнесу, державних структур та повсякденного життя користувачів, то вимоги до їх захисту постійно зростають. Зловмисники постійно вдосконалюють методи атак, тому розробники та адміністратори повинні впроваджувати комплексні та системні заходи, які враховують усі етапи життєвого циклу програмного забезпечення: від проєктування до експлуатації.

Аутентифікація та авторизація

- Використовувати сучасні протоколи автентифікації, такі як **OAuth 2.0** та **OpenID Connect**, які забезпечують безпечний обмін обліковими даними.
- Реалізувати багаторівневу систему авторизації з чітким розмежуванням прав доступу.
- Впроваджувати політику обов’язкової зміни паролів, застосовувати **мультифакторну аутентифікацію (MFA)**.
- Використовувати принцип **Zero Trust** для постійної перевірки довіри навіть до авторизованих користувачів.

Захист від ін’єкцій

- Використовувати параметризовані запити або підготовлені вирази для захисту від SQL-ін’єкцій.
- Виконувати валідацію введених даних за допомогою білих списків.
- Використовувати **ORM-бібліотеки** як проміжний рівень між програмою та базою даних.
- Приховувати внутрішні повідомлення про помилки від користувачів.

Управління сеансами

- Використовувати криптографічно захищені випадкові токени для ідентифікації сесій.
- Зберігати токени у cookies з атрибутами **HttpOnly**, **Secure** та **SameSite**.
- Регулярно оновлювати сеансові токени та завершувати неактивні сесії.

- Шифрувати передані дані за допомогою **TLS/SSL**.

Захист від CSRF

- Використовувати **CSRF-токени** для кожної сесії користувача.
- Використовувати атрибут **SameSite** у cookies.
- Реалізувати метод **Double Submit Cookie** для додаткової перевірки.
- Використовувати перевірку заголовків **Referer** або **Origin**.

Контроль доступу

- Дотримуватися принципу **найменших привілеїв (Least Privilege)**.
- Регулярно переглядати права доступу та оновлювати політики безпеки.
- Використовувати механізм розділення обов'язків (Separation of Duties).

Моніторинг і аудит безпеки

- Використовувати системи **SIEM** для аналізу журналів подій.
- Впроваджувати **IDS/IPS** для виявлення та запобігання вторгнень.
- Регулярно проводити **пентести** та аудити безпеки.
- Використовувати методи машинного навчання для виявлення аномалій.

Оновлення та патчі

- Регулярно оновлювати сервери, бази даних, бібліотеки та фреймворки.
- Використовувати централізовані системи управління оновленнями.
- Інтегрувати безпеку в CI/CD-процеси за допомогою підходу **DevSecOps**.

Навчання персоналу

- Проводити регулярні тренінги з безпеки для розробників і адміністраторів.
- Використовувати практичні завдання для виявлення та усунення вразливостей.
- Формувати культуру кібербезпеки в команді.

Тестування

- Використовувати **unit-тести та інтеграційні тести**, що перевіряють сценарії безпеки.
- Проводити **фаззинг-тестування** для виявлення неочікуваних помилок.
- Використовувати **Red Team/Blue Team вправи** для перевірки готовності системи до атак.

ВИСНОВКИ

У ході проведеного дослідження на тему «Методи захисту персональних даних у телекомунікаційних мережах» було проаналізовано сучасний стан інформаційної безпеки, окреслено ключові проблеми, що виникають при збереженні конфіденційності, цілісності та доступності даних, а також розглянуто широкий спектр методів і технологій, спрямованих на протидію кіберзагрозам.

Особливу увагу було приділено вивченню таких типових атак, як SQL-ін’єкції, XSS, переповнення буфера, CSRF, DDoS та інші методи несанкціонованого доступу, що становлять серйозну небезпеку для веб-додатків і телекомунікаційних мереж. У дослідженні було показано, що ці атаки не лише порушують стабільність роботи систем, але й можуть призвести до витоку персональних даних, їх модифікації або повного знищення. Зловмисники активно використовують уразливості в архітектурі додатків, слабкі паролі, недосконалі механізми авторизації та неефективне управління сеансами для досягнення своїх цілей.

Було встановлено, що для захисту телекомунікаційних мереж потрібен багаторівневий підхід, який поєднує як технічні, так і організаційні заходи. Серед найбільш ефективних методів було виділено: використання параметризованих запитів і ORM для запобігання SQL-ін’єкціям; застосування політик Content-Security-Policy та санітизації введених даних для протидії XSS; контроль довжини вхідних даних і використання захисних механізмів операційних систем (ASLR, DEP) для запобігання переповненню буфера; реалізацію CSRF-токенів та механізму Double Submit Cookie для протидії CSRF-атакам.

Не менш важливим елементом є захист каналів зв’язку. Використання SSL/TLS-протоколів забезпечує безпечну передачу даних між клієнтом і сервером, унеможливлюючи перехоплення або модифікацію інформації під час її транспортування. Додатково було розглянуто роль протоколів OAuth та OpenID Connect у безпечній автентифікації та управлінні доступом, що дозволяють мінімізувати ризики, пов’язані з передачею облікових даних стороннім сервісам.

Окреме місце у роботі посіло питання організації доступу на рівні баз даних, зокрема впровадження політик найменших привілеїв, розмежування прав доступу залежно від ролей користувачів та регулярний аудит політик безпеки. Також було наголошено на важливості моніторингу подій, журналювання, впровадження SIEM-систем та проведення пентестів, що дозволяють виявляти вразливості до того, як вони будуть використані зловмисниками.

У дослідженні підкреслено, що сучасні методи захисту повинні бути доповнені організаційними заходами: навчанням персоналу правилам кібергігієни, формуванням культури безпеки у компаніях і державних структурах, впровадженням регулярного тестування, оновленням програмного забезпечення та застосуванням підходів DevSecOps.

Таким чином, у результаті виконаної роботи можна зробити висновок, що забезпечення захисту персональних даних у телекомунікаційних мережах можливе лише за умови комплексного підходу, що об’єднує сучасні криптографічні методи, системи фільтрації та виявлення атак, політики управління доступом і постійний моніторинг. В умовах стрімкого зростання обсягів даних і поширення технологій 5G/6G, IoT та хмарних сервісів, питання безпеки стає ще більш актуальним. Тому перспективними напрямами подальших досліджень є розвиток методів на основі штучного інтелекту для виявлення аномалій, впровадження Zero Trust-архітектури у телекомунікаційні мережі та вдосконалення протоколів шифрування для забезпечення стійкості до атак майбутнього покоління.

ПЕРЕЛІК ПОСИЛАНЬ

1. Terahertz Wireless Communication Components and System Design. Springer, 2021. – книга про компоненти і архітектуру THz-систем.
2. Eichler, T.; Ziegler, R. White Paper: Fundamentals of THz Technology for 6G. Rohde & Schwarz, 2022. – огляд фундаментальних властивостей THz-спектру.
3. Tripathi, S.; Sabu, N. V.; Gupta, A. K.; Dhillon, H. S. Millimeter-Wave and Terahertz Spectrum for 6G Wireless. arXiv, 2021. – про характеристики каналу, моделі, застосування THz.
4. Polese, M., et al. Toward End-to-End, Full-Stack 6G Terahertz Networks. 2020. – охоплює повний стек: від фізичного до мережевого рівня.
5. Yang, F., et al. Terahertz Reconfigurable Intelligent Surfaces (RISs) for 6G Communication Links. PMC, 2022. – про RIS у THz-зв’язку.
6. Singh, A.; Petrov, V.; Guerboukha, H.; Reddy, I. V. A. K.; Knightly, E. W.; Jornet, J. M. Wavefront Engineering: Realizing Efficient Terahertz Band Communications in 6G and Beyond. arXiv, 2023. – стаття про форму хвиль (wavefront) та beamforming у THz.
7. Cai, X.; Cheng, X.; Tufvesson, F. Toward 6G with Terahertz Communications: Understanding the Propagation Channels. arXiv, 2022. – про моделі каналу та вимірювання в THz.
8. Shafie, A., et al. Terahertz Communications for 6G and Beyond Wireless. ACM / IEEE, 2023. – про системи THz-зв’язку по шарах (PHY, link, network).
9. Langde, P., et al. A Journey of Terahertz Communication: An IRS Integration. 2025. – IRS + THz, інтеграція.
10. Lee, J. Y., et al. 6G Communications Through Sub-Terahertz CMOS Power Amplifiers and Hardware Designs. ScienceDirect, 2025. – про суб-THz апаратні рішення.
11. IEEE Wireless Communications – Special Issue: Terahertz Communications and Sensing for 6G and Beyond. IEEE, 2024. – збірка поточних досягнень.
12. Rohde & Schwarz. White Paper: Fundamentals of THz Technology for 6G. 2022. – індустрійний огляд властивостей THz.

13. D’heer, C.; Reynaert, P. THz and Sub-THz CMOS Electronics for High-Speed Telecommunication: Architectures and Circuits for Future 6G Transceivers. Springer, 2024. – про дизайн CMOS схем над 100 GHz.
14. Thomas, S.; Virdi, J. S.; Babakhani, A.; Roberts, I. P. A Survey on Advancements in THz Technology for 6G: Systems, Circuits, Antennas, and Experiments. arXiv, 2024. – огляд систем і антен THz.
15. Lee, J. Y., et al. 6G Communications Through Sub-Terahertz CMOS Power Amplifiers: Design Challenges and Trends. Heliyon / arXiv, 2025. – про підсилювачі потужності в sub-THz.
16. Rasilainen, K.; Phan, T. D.; Berg, M.; Pärssinen, A.; Soh, P. J. Hardware Aspects of Sub-THz Antennas and Reconfigurable Intelligent Surfaces for 6G Communications. IEEE JSAC, 2023. – про антени і RIS у sub-THz.
17. Jiang, W.; Zhou, Q.; He, J.; Habibi, M. A.; Melnyk, S.; El Absi, M.; Han, B.; Di Renzo, M.; Schotten, H.; Luo, F.-L.; El-Bawab, T. S.; Juntti, M.; Debbah, M.; Leung, V. C. M. Terahertz Communications and Sensing for 6G and Beyond: A Comprehensive Review. arXiv, 2023. – огляд комунікацій і сенсингу на THz.
18. Xue, Q.; Ji, C.; Ma, S.; Guo, J.; Xu, Y.; Chen, Q.; Zhang, W. A Survey of Beam Management for mmWave and THz Communications Towards 6G. arXiv, 2023. – про управління променями для THz.
19. Han, C.; Wang, Y.; Li, Y.; Chen, Y.; Abbasi, N. A.; Kürner, T.; Molisch, A. F. Terahertz Wireless Channels: A Holistic Survey on Measurement, Modeling, and Analysis. arXiv, 2021. – вимірювання і моделювання каналів THz.
20. Silva, J. D. de S. E.; et al. A Survey on the Impact of Intelligent Surfaces in THz Communication Systems. PMC, 2023. – про використання поверхонь RIS/IS у THz.
21. Singh, R. C. Security of Terahertz Communications: Challenges & Solutions. TechRxiv, 2024. – про безпеку THz каналів.
22. Power Modeling in mm-Wave and Terahertz CMOS Transmitters. arXiv, 2024. – модель енергоспоживання в mmWave/THz передавачах.
23. A Survey of 6G Wireless Communications: Emerging Technologies. arXiv, 2020. – розглядає THz як одну з ключових технологій 6G.

24. Toward 6G Communication Networks: Terahertz Frequency Challenges and Open Research Issues. ResearchGate / IJECE, 2025. – проблеми і відкриті питання THz.

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ТЕЛЕКОМУНІКАЦІЙ

КВАЛІФІКАЦІЙНА РОБОТА

МЕТОДИ ЗАХИСТУ ПЕРСОНАЛЬНИХ ДАНИХ У ТЕЛЕКОМУНІКАЦІЙНИХ МЕРЕЖАХ

Виконав:
студент групи ТСДМ-62
Роман КАШПУР

МЕТА РОБОТИ ТА АКТУАЛЬНІСТЬ

- Мета роботи – дослідження сучасних методів захисту персональних даних у телекомунікаційних мережах та аналіз ефективності симетричних і асиметричних підходів до побудови систем протидії DDoS-атакам, з подальшим формуванням рекомендацій щодо підвищення рівня безпеки інформаційних систем.
- Об’єкт дослідження – процеси забезпечення інформаційної безпеки у телекомунікаційних мережах, зокрема захист персональних даних і стабільності роботи мережевої інфраструктури в умовах кіберзагроз.
- Предмет дослідження – методи, алгоритми та технології захисту даних і мережевих ресурсів — зокрема, засоби автентифікації, авторизації, шифрування, а також механізми симетричного та асиметричного DDoS-захисту.

Актуальність.

- У сучасних умовах глобальної цифровізації питання забезпечення кібербезпеки та захисту персональних даних є одним із найпріоритетніших напрямів розвитку телекомунікаційних систем. Масове впровадження технологій 5G/6G, хмарних сервісів та інтернету речей (IoT) призводить до зростання обсягів оброблюваної інформації та підвищення ризику її компрометації.
- Тема роботи є надзвичайно актуальною, оскільки охоплює як захист баз даних і веб-додатків від типових атак (SQL-ін’єкцій, несанкціонованого доступу, зламу паролів), так і сучасні методи протидії DDoS-атакам у симетричних та асиметричних мережах. Це безпосередньо відповідає сучасним викликам у сфері кіберзахисту телекомунікаційних інфраструктур.

Основні загрози

- SQL-ін’єкції (SQL Injection).

Це атака, під час якої зловмисник вставляє шкідливий код у SQL-запит через поля введення. У результаті він може отримати несанкціонований доступ до бази даних або змінити її вміст. Причини — відсутність фільтрації введених даних і неправильне екранування символів.
Захист: параметризовані запити, фільтрація даних, використання ORM-систем, обмеження прав доступу.

- XSS (Cross-Site Scripting).

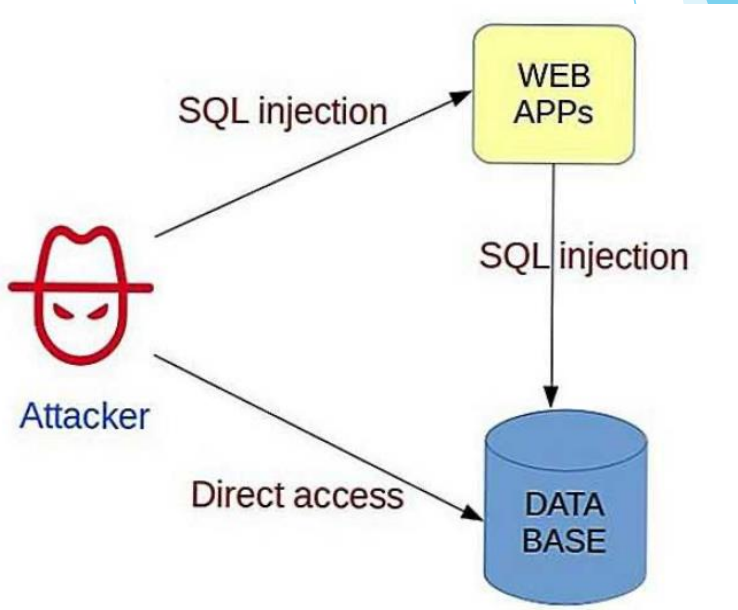
Дозволяє впровадити шкідливий JavaScript у сторінку, що виконується у браузері користувача, крадучи cookies або дані.
Захист: екранування введених даних, політика Content-Security-Policy, серверна валідація, безпечні фреймворки.

- Переповнення буфера.

Виникає при записі надлишкових даних у пам’ять, що може дати змогу виконати шкідливий код.
Захист: перевірка довжини введення, безпечні функції, механізми ASLR і DEP.

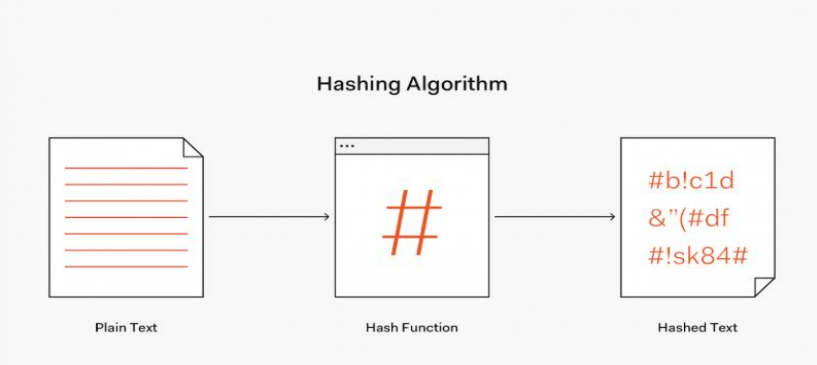
- CSRF (Cross-Site Request Forgery).

Атака, що змушує авторизованого користувача виконувати небажані дії.
Захист: CSRF-токени, атрибут SameSite cookies, перевірка заголовка Referer.

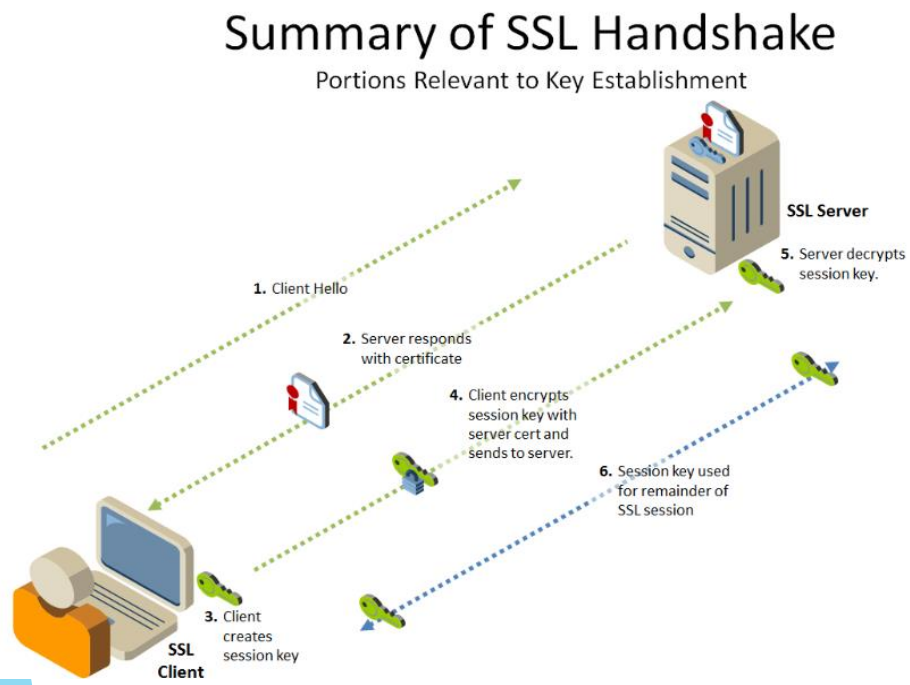


Ілюстрація роботи SQL-ін'єкції

Методи автентифікації



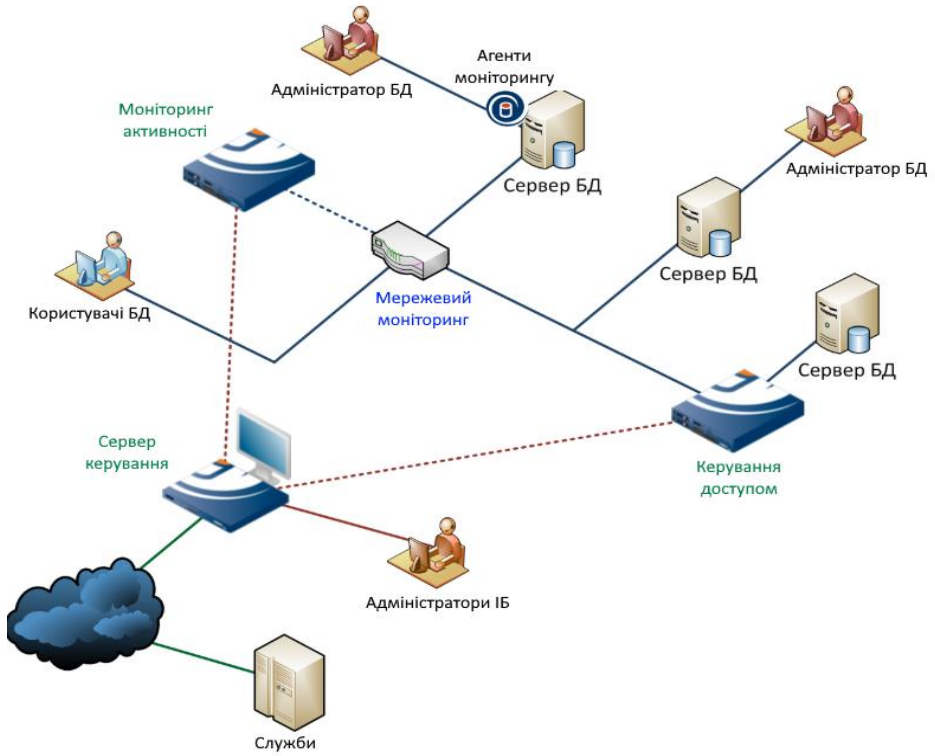
Процес хешування



Процес SSL/TLS-шифрування

- Хешування паролів — ключовий механізм захисту даних, який перетворює пароль у незворотний хеш, зберігаючи лише його цифровий відбиток. Це унеможливлює відновлення пароля навіть у разі витоку бази. Для підвищення безпеки використовують **bcrypt, scrypt, Argon2** і додавання **salt**, що ускладнює атаки словниками чи rainbow tables. Важливим доповненням є **SSL/TLS-шифрування**, яке захищає дані під час передавання в мережі. Разом із багатофакторною автентифікацією та оновленням сертифікатів такі методи формують основу сучасної системи кіберзахисту, мінімізуючи ризик викрадення або компрометації паролів користувачів.

Авторизація та права доступу

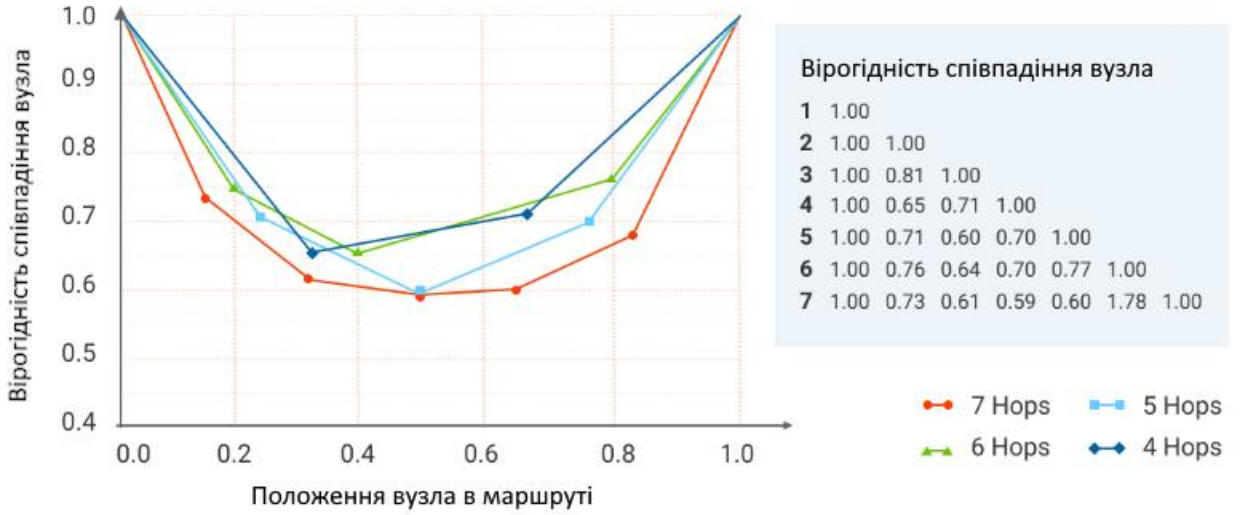


Умовна структура компанії

- Методи авторизації в базах даних визначають, хто і до якої інформації має доступ, забезпечуючи конфіденційність і стабільність роботи системи. Розмежування прав доступу дозволяє призначати користувачам лише необхідні повноваження: аналітики бачать узагальнені дані, відділ продажів — контакти клієнтів, бухгалтерія — фінансові показники. Такий підхід реалізує принцип найменших привілеїв, мінімізує ризики витоку інформації та підвищує ефективність роботи. Авторизація на рівні бази даних відповідає міжнародним стандартам безпеки (GDPR, ISO/IEC 27001) і є ключовим елементом сучасної системи захисту корпоративних даних.

Видідення потоку та захист асиметричних даних

- Дослідження «Observing routing asymmetry in Internet traffic» показало, що навіть у двосторонніх протоколах Інтернету спостерігається значна асиметрія трафіку. Потоки даних у прямому та зворотному напрямках часто відрізняються за кількістю пакетів, маршрутом і затримками. Аналіз із використанням платформи RIPE Atlas підтвердив, що лише близько 12% з'єднань мають повну симетрію маршрутів. Асиметрія посилюється на рівні міжоператорських зв'язків Tier-1 і Tier-2 через політики маршрутизації та економічні чинники. Це впливає на якість зв'язку й ефективність систем захисту, зокрема при організації симетричного DDoS-захисту.



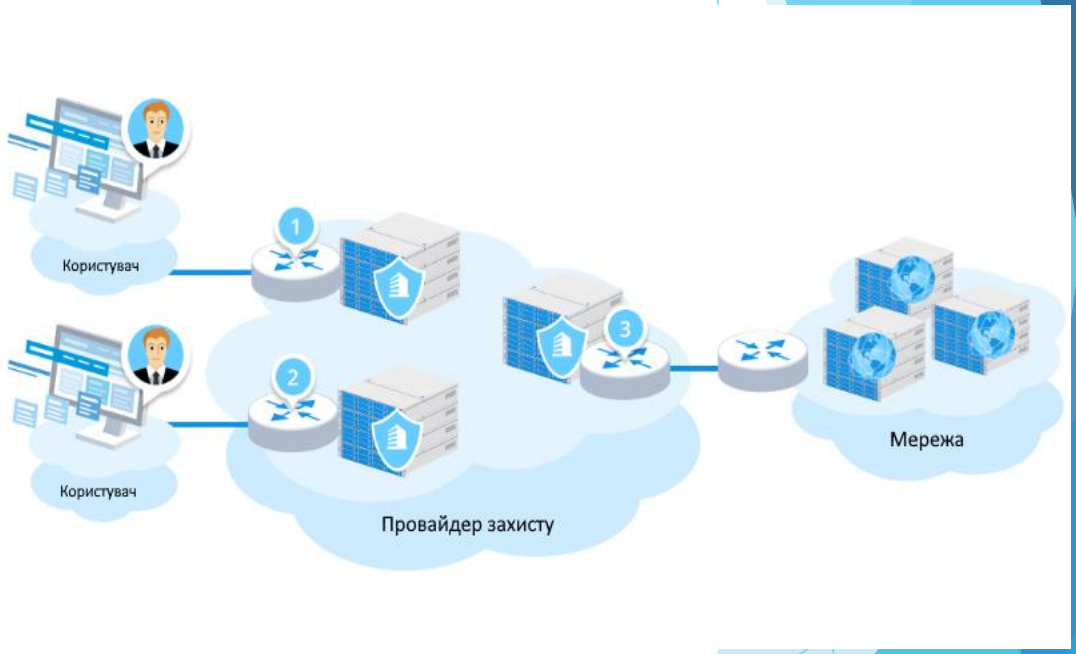
Прояв асиметрії маршрутизації



Симетрична та асиметрична схеми підключення захисту

Використання фільтруючого обладнання

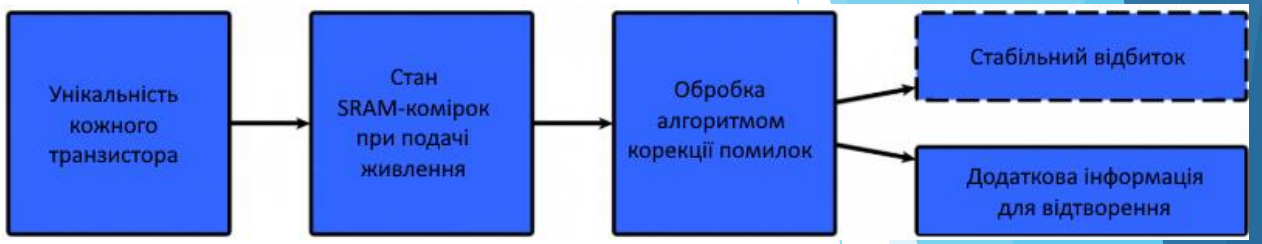
- Використання фільтруючого обладнання залежить від архітектури мережі. Якщо є один edge-маршрутизатор, достатньо встановити фільтр у симетричному режимі — він забезпечує повний аналіз трафіку в обох напрямках. У складніших мережах із кількома маршрутизаторами застосовують асиметричні схеми: кожен пристрій фільтрує свою частину потоків. Для підвищення ефективності використовують синхронізацію або обмін метаданими між фільтрами. У провайдерів часто впроваджують DDoS-захист як сервіс (DPaaS), де фільтрація виконується зовнішнім обладнанням. Головне завдання — баланс між точністю фільтрації, вартістю рішень та стійкістю мережі.



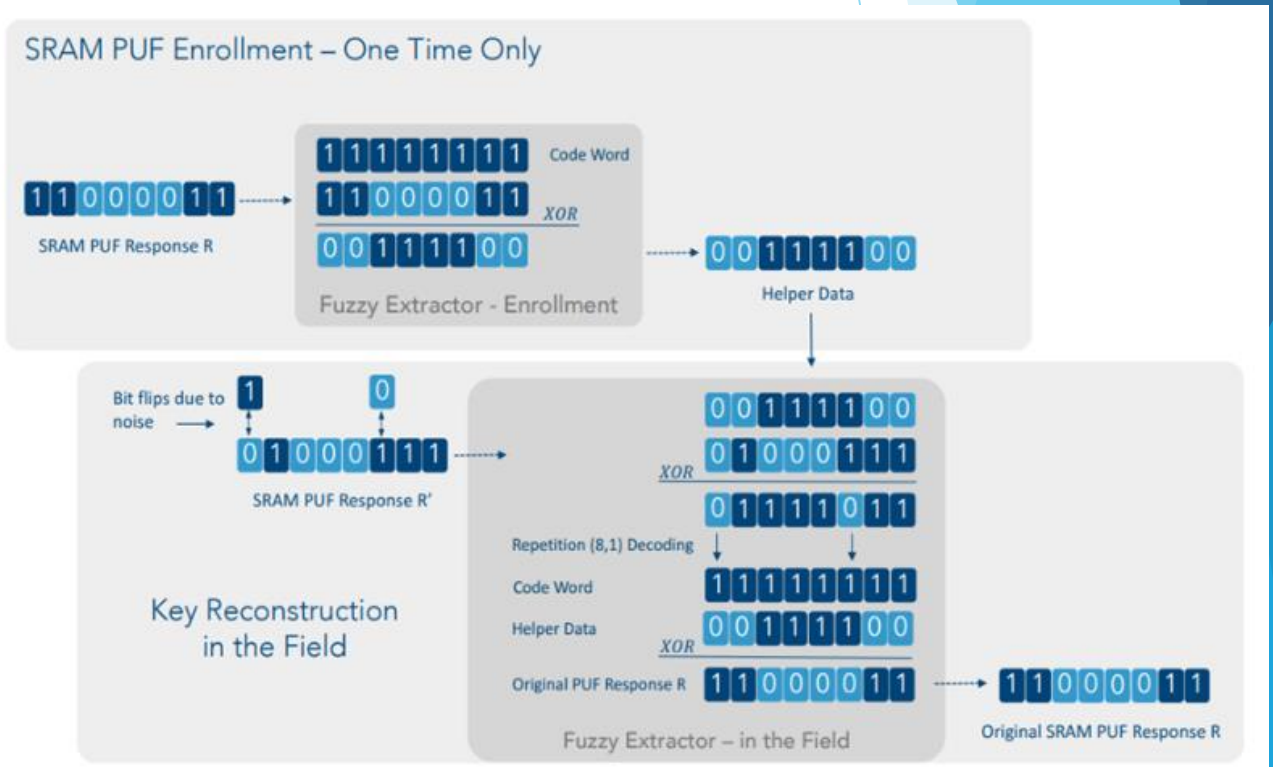
Приклад схеми для встановлення захисту

ТЕХНОЛОГІЯ ЗАХИСТУ SRAM PUF

- **Фізично неклонована функція (PUF)** — це метод апаратного захисту, що використовує унікальні фізичні властивості мікросхем, подібно до біометрії людини. Кожен чип має власні мікроскопічні відмінності, які неможливо клонувати, що дозволяє створити унікальний «цифровий відбиток» — **Digital Fingerprint**.
- Найпоширенішою реалізацією є **SRAM PUF**: при кожному ввімкненні пристрою комірки пам’яті приймають випадкові значення, з яких формується унікальний набір даних. Алгоритми корекції помилок перетворюють цей шумовий відбиток у стабільний і відтворюваний **Digital Fingerprint**, а допоміжна **Helper Data** дозволяє його точно відновити.
- **SRAM PUF** використовується для створення ключів шифрування, автентифікації обладнання, захисту IoT-пристроїв і перевірки справжності компонентів у мережах **5G/6G**. У мікроконтролерах **NXP LPC55Sxx** технологія реалізована на базі рішень **Intrinsic ID**, забезпечуючи надійний рівень апаратної безпеки без зовнішніх модулів.



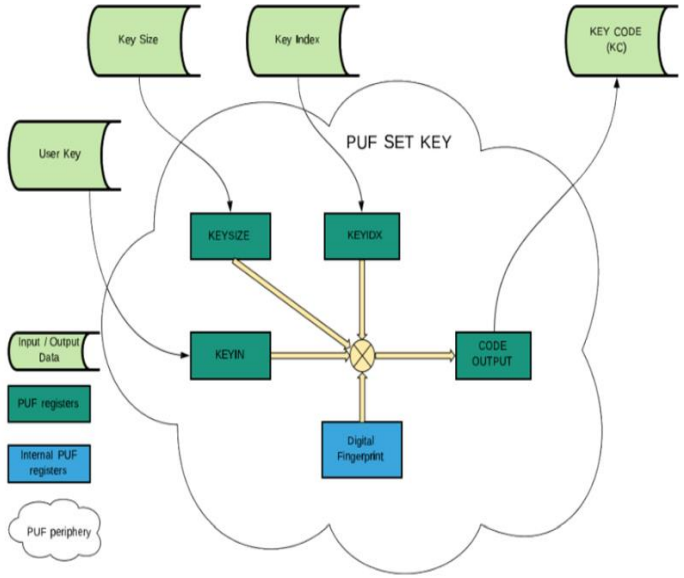
Процес формування основних складових SRAM PUF — Digital Fingerprint та Helper Data



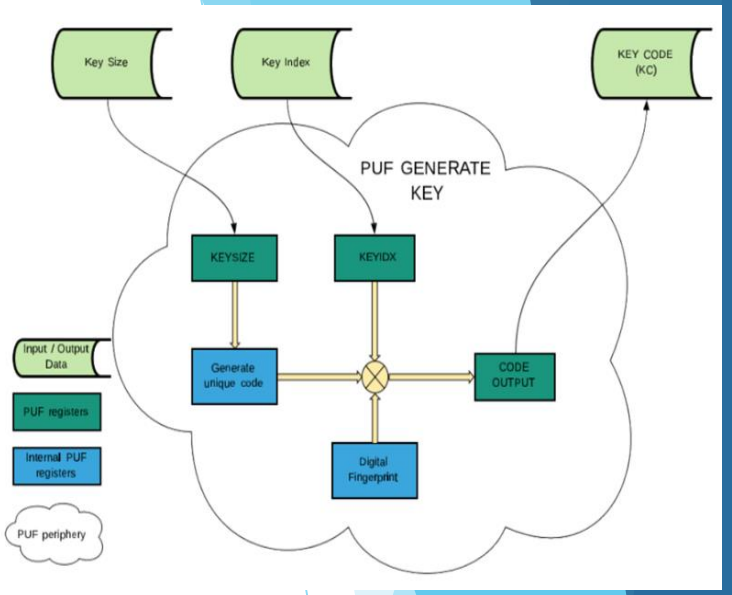
Відновлення оригінального Digital Fingerprint за допомогою Helper Data

Робота з ключами користувачів

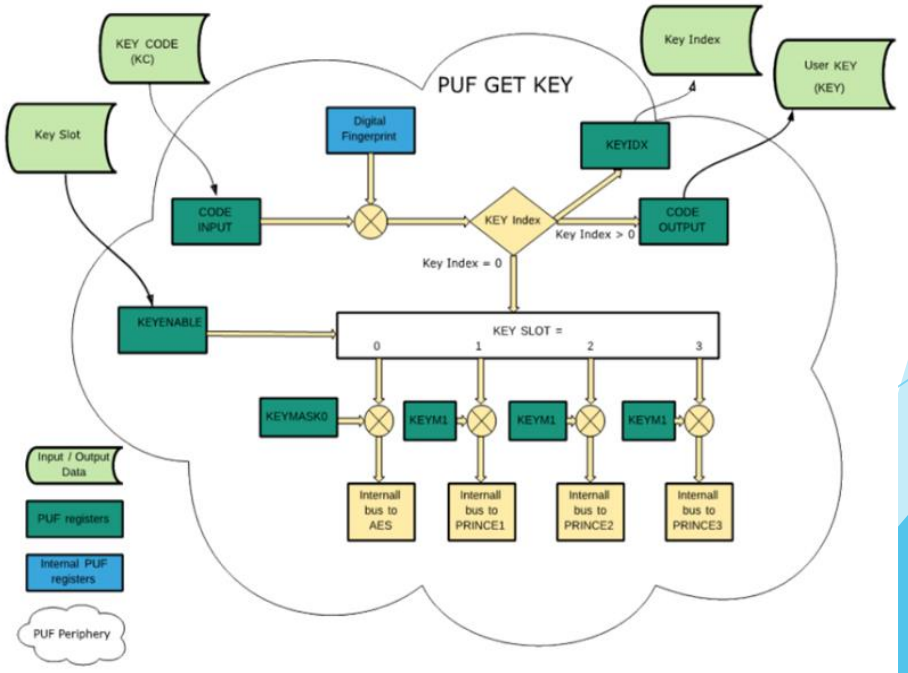
- **PUF SetKey – перетворення та захист ключа** Команда **SetKey** перетворює користувацький ключ у захищений **Key Code**, який можна безпечно зберігати чи передавати. Кожен код прив’язаний до конкретної мікросхеми через унікальний **Digital Fingerprint**, тому його неможливо використати на іншому пристрої. Особливий **Key Code з індексом 0** доступний лише апаратним модулям (AES, PRINCE) і ніколи не з’являється у відкритому вигляді, що гарантує максимальну безпеку.
- **PUF GenerateKey – генерація випадкового ключа** Команда **GenerateKey** створює унікальний випадковий ключ усередині PUF-блоку та одразу перетворює його у Key Code. Ключ ніколи не існує у відкритій формі, що робить метод ідеальним для автентифікації сеансів і захисту IoT-пристроїв.
- **PUF GetKey – відновлення ключа** Команда **GetKey** відновлює користувацький ключ або передає його безпосередньо у криптомодулі AES чи PRINCE. Це забезпечує безпечне дешифрування даних «на льоту» (XIP) без втрати продуктивності та гнучку інтеграцію в системи безпеки.



Метод SetKey: процес захисту користувацького ключа за допомогою PUF



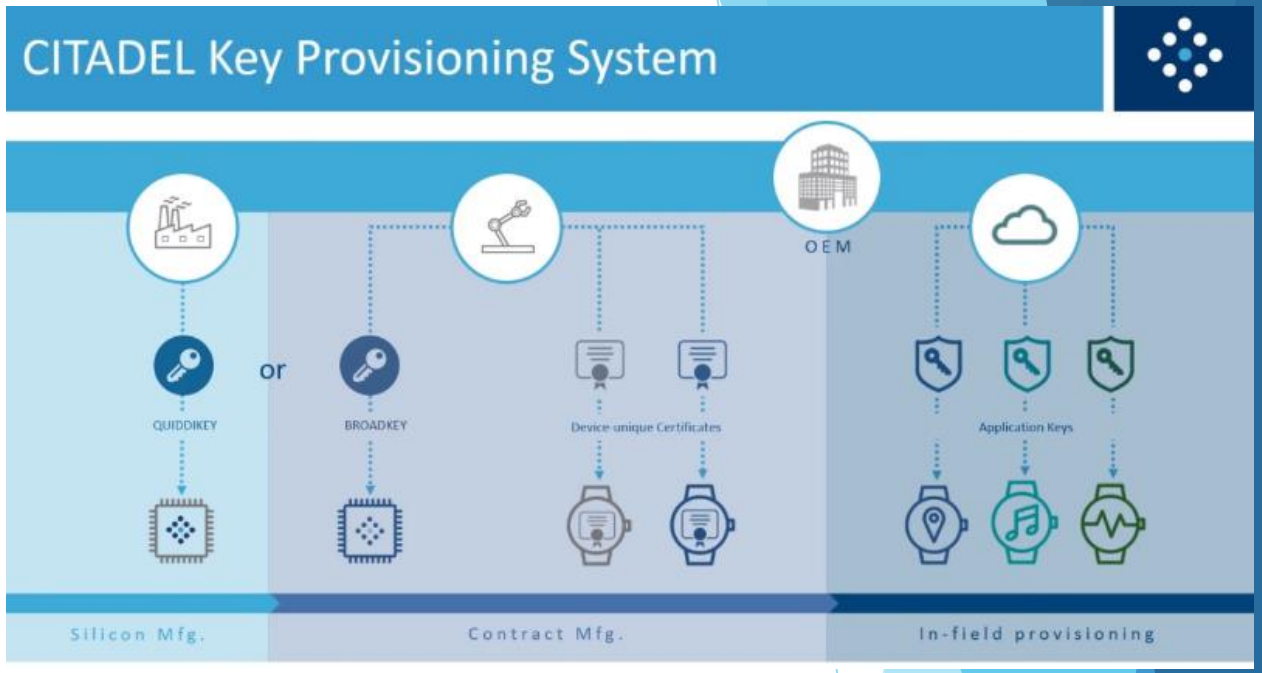
Метод GenerateKey: генерація випадкового ключа та його подальший захист у PUF-блоці



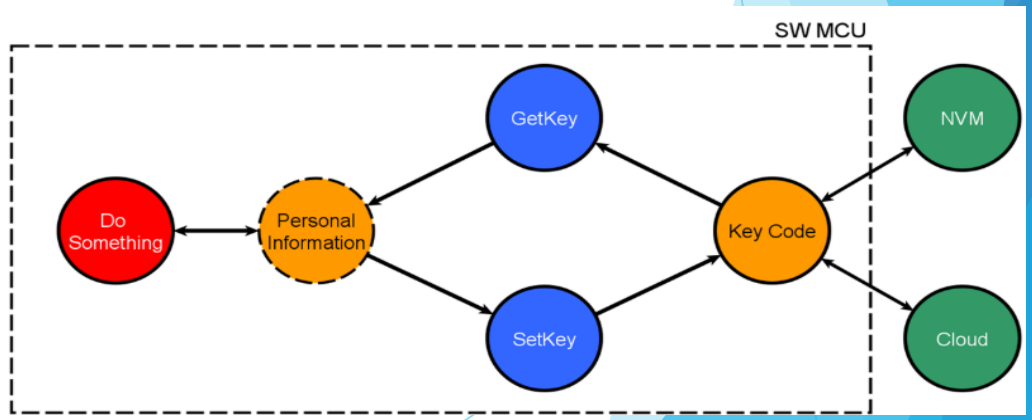
Метод GetKey: процес відновлення користувацького ключа та передача його у криптографічні блоки

Реалізація SRAM PUF

- SRAM PUF використовується для створення унікальних апаратних ідентифікаторів, що забезпечують високий рівень безпеки пристроїв. Основні напрями застосування: **персоналізація**, **безпечне завантаження (Secure Boot)** та **захист ключів і даних**.
- **Персоналізація пристроїв** відбувається через процедуру *Enroll*, яка формує унікальний цифровий відбиток. У довіреному середовищі це гарантує автентичність, а в недовіреному — потребує криптографічного захисту та сертифікатів CA для уникнення атак типу *Man-in-the-Middle*.
- **Secure Boot** забезпечує запуск лише перевіреного ПЗ. Ключі зберігаються у зашифрованому вигляді як *Key Code* і доступні лише апаратним криптомодулям AES/PRINCE.
- **Захист даних** здійснюється шляхом зберігання користувацьких ключів у форматі *Key Code*, що робить їх придатними лише для конкретного пристрою.
- Основні ризики: **старіння кремнію**, що впливає на стабільність PUF-відбитка, та **можливість атак із використанням машинного навчання**, які можуть відтворювати поведінку PUF.



Процес персоналізації: приклад інтеграції сертифікатів на основі SRAM PUF



Сценарій використання SRAM PUF для захисту персональних даних і ключів

Рекомендації з безпеки для розробників і адміністраторів

Захист веб-додатків — це комплексний процес, що охоплює технічні та організаційні заходи на всіх етапах життєвого циклу системи. З метою підвищення безпеки слід дотримуватися таких практик:

- **Аутентифікація та авторизація:** застосовувати протоколи OAuth 2.0, OpenID Connect, MFA та принцип Zero Trust.
- **Захист від ін'єкцій:** використовувати параметризовані запити, ORM-бібліотеки, фільтрацію введених даних і приховувати повідомлення про помилки.
- **Управління сесіями:** створювати захищені токени, зберігати їх у cookies з атрибутами HttpOnly, Secure, SameSite, шифрувати трафік TLS.
- **CSRF-захист:** впроваджувати унікальні CSRF-токени, перевірку заголовків і атрибут SameSite.
- **Контроль доступу:** дотримуватися принципу найменших привілеїв і регулярно перевіряти права користувачів.
- **Моніторинг і оновлення:** використовувати SIEM, IDS/IPS, DevSecOps, пентести та навчання персоналу для підтримки кіберстійкості системи.

ВИСНОВКИ

- У кваліфікаційній роботі проаналізовано сучасний стан інформаційної безпеки та основні кіберзагрози.
- Розглянуто типові атаки — SQL-ін’єкції, XSS, CSRF, DDoS, а також ефективні методи протидії: параметризовані запити, фільтрація введення, SSL/TLS, OAuth, OpenID і політика найменших привілеїв.
- Підкреслено важливість організаційних заходів — підготовки персоналу, аудиту безпеки, оновлення ПЗ та DevSecOps-підходу.
- Зроблено висновок, що надійний захист персональних даних можливий лише за умови комплексного підходу, який поєднує технічні, криптографічні та організаційні засоби. Перспективи — використання ШІ, Zero Trust і нових протоколів шифрування.

Дякую за увагу!

Протокол аналізу звіту подібності науковим керівником

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Роман КАШПУР

Назва: МЕТОДИ ЗАХИСТУ ПЕРСОНАЛЬНИХ ДАНИХ У ТЕЛЕКОМУНІКАЦІЙНИХ МЕРЕЖАХ

Координатор: Юрій МЕЛЬНИК

Підрозділ: ННІТ

Коефіцієнт подібності 1:0.45

Коефіцієнт подібності 2:0

Тривога: 0

Після аналізу Звіту подібності констатую наступне:

☐ виявлені в роботі запозичення є сумлінними і не мають ознак плагіату. Тому робота визнається самостійною і допускається до захисту;

☐ виявлені в роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і самостійності її автора. Роботу направити на доопрацювання;

☐ виявлені в роботі запозичення є недобросовісними і мають ознаки плагіату або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень. У зв'язку з чим, робота не допускається до захисту.

…… ……

Дата *Підпис Наукового керівника*