**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**
**ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики
Кафедра математичної інформатики

# Дипломна робота

**на здобуття ступеня бакалавра**
за спеціальністю 122 Комп'ютерні науки
на тему:

# ДЕЦЕНТРАЛІЗОВАНА СИСТЕМА КОНТРОЛЮ ДОСТУПА З ВИКОРИСТАННЯМ КРИПТОГРАФІЧНИХ ПРОТОКОЛІВ

Виконав студент 4-го курсу
Семерак Данило Остапович

(підпис)

Науковий керівник:
професор
Анісімов Анатолій Васильович

(підпис)

Засвідчую, що в цій курсовій роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент _____
(підпис)

Київ – 2021

# РЕФЕРАТ

Обсяг роботи 41 сторінка, 21 ілюстрація, 1 таблиця, 3 джерел посилань.

СКД, ДОСТУП, КОНТРОЛЬ ДОСТУПУ, КРИПТОГРАФІЯ, ДЕЦЕНТРАЛІЗАЦІЯ, ЦИФРОВИЙ ПІДПИС

Об’єктом розроблення є системи контролю фізичного доступу, які відрізняються від традиційних систем.

Метою розробки є спроба створити альтернативні системи контролю фізичного доступу та провести аналіз переваг та недоліків кожної з них.

У процесі формування принципів роботи систем використовувалися існуючі методи створення цифрових підписів та шифрування. Для розроблення демонстраційного програмного забезпечення використовувалася мова програмування python.

У результаті роботи було описано декілька нових систем контролю доступу які відрізняються від традиційних більшою гнучкістю та відсутністю центральної бази з дозволами.

Також у роботі описано можливі сценарії застосування запропонованих систем контролю доступу і вказано переваги й недоліки за умови їх використання у кожній окремій ситуації.

У подальшому доцільним є розробка і створення програмного та електро-механічних прототипів для випробовування запропонованих систем у реальному середовищі.

# ЗМІСТ

# ВСТУП

В офісних центрах, будинках, стоянках чи інших територіях часто виникає потреба у розмежуванні прав фізичного доступу на певні зони. Для задач ідентифікації користувачів, обмеження доступу, ведення статистики активності та інших використовують системи контролю і управління доступом. Такі системи складаються з: ідентифікатора, який є у кожного користувача; зчитувача, який забезпечує зв'язок між ідентифікатором і контролером; контролер, який здійснює перед обробку та забезпечує зв'язок з центральним комп'ютером; центральний комп'ютер який зберігає інформацію про дозволи та може керувати усіма підключеними контролерами. Така система має декілька недоліків, а саме зв'язок контролерів з центральним комп'ютером та сам центральний комп'ютер, який є слабким місцем цієї архітектури.

**Метою і завданням роботи** є проектування нової альтернативної системи контролю і управління фізичним доступом, яка б не мала недоліків традиційних систем. Також завданням є створення програмного забезпечення, яке б демонструвало роботу спроектованих систем.

**Об'єктом розроблення** є децентралізована система контролю доступу. У проектуванні використовуються існуючі методи цифрового підпису та шифрування.

**Сфери застосування** описана система контролю доступу знайде широке застосування у офісах компаній, бізнес-центрах, підприємствах, багатоквартирних будинках та готелях.

# ОСНОВНА ЧАСТИНА

## 1. Постановка проблеми

### 1.1. Огляд існуючих СКД

Система контролю і управління доступом (скорочено СКУД або СКД) — це комплекс технічних та програмних засобів безпеки, що здійснює регулювання входу / виходу та переміщень людей чи транспортних об’єктів на територіях, які знаходяться під охороною, для адміністративного моніторингу та попереджень несанкціонованого проникнення [1].

В загальному випадку топологія такої системи зображена на рисунку (1).

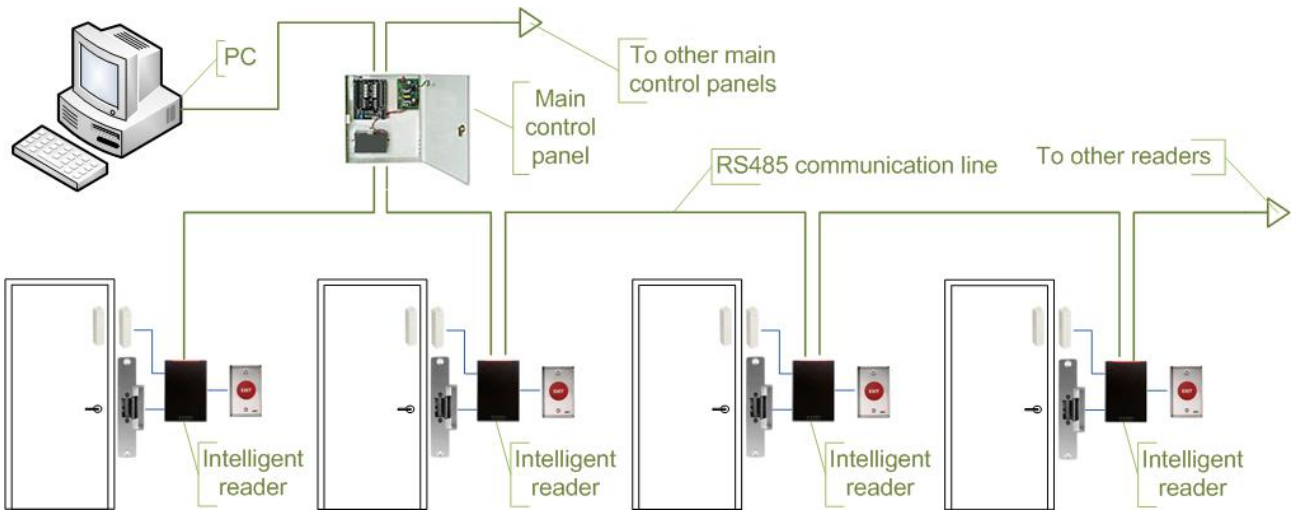



Рисунок 1 – Топологія існуючих СКД [3]

Тобто біля кожних дверей є зчитувач (наприклад карточки RFID чи NFC, або модуль Bluetooth для комунікації з мобільним девайсом), який надсилає інформацію про людину до центрального комп'ютера, де зберігається інформація про доступи. Якщо людина має необхідні права, то надсилається сигнал до дверей, щоб ті відкрилися.

## 1.2. Недоліки існуючих СКД

Уявимо наступну ситуацію. Є офісний центр. Ліфти піднімаються тільки згідно прав доступу працівників (якщо людина працює на 2 поверсі, то ліфт зупиниться тільки там). Аналогічно двері на поверсі відкриваються теж тільки по доступу. Є працівник, який працює на 5 поверсі офісного центру. Він хоче запросити до себе в офіс працівника з 2 поверху. Для цього існує два варіанти:

1) Необхідно внести зміни в СКД, щоб людина з 2 поверху змогла потрапити на 5 поверх. Якщо центральний комп'ютер не має доступу до інтернет, то це потрібно робити вручну.
2) Працівникам необхідно зустрілися у вестибюлі і щоб перший провів другого працівника особисто.

Обидва випадки не є зручні, тому така не гнучкість є недоліком. Виокремимо всі недоліки:

- Складно змінювати права доступу, оскільки потрібно фізично знаходитися біля центрального комп'ютера, який відключений від мережі інтернет.
- Є центральний комп'ютер, який я вразливим місцем. Щоб не відбулося хакерської атаки, комп'ютер часто від'єднують від мережі інтернет.
- Така система потребує проведення кабелів для комунікації між модулями системи і центральним комп'ютером.

- Для забезпечення працездатності системи необхідна велика кількість пристроїв.

### 1.3. Вимоги для нової СКД і задачі, які вона має вирішувати

Метою роботи є теоретичне розроблення системи контролю доступу, яка буде задовольняти наступні вимоги:

1) Забезпечувати функціонал СКД, які були описані вище.
2) Не повинна мати єдиного центру керування, де зберігаються усі дозволи.
3) Забезпечити можливість користувачам надавати доступи іншим користувачам.
4) Повинна бути компактніша за існуючі СКД.

## 2. Теоретична частина

### 2.1. Цифровий підпис

Опишемо загальний принцип роботи всіх алгоритмів підпису з асинхронним ключем [2].

1) При ініціалізації генерується пара ключів: відкритий (pk) та закритий (sk). Закритий ключ користувач зберігає в себе та нікому не передає. Відкритий ключ є загальнодоступним і його повинні знати всі бажаючі.

$$(sk, pk) \coloneqq generate\_keys(keysize)$$

2) Для підпису якого тексту необхідно мати сам текст (message) і секретний ключ (sk). В результаті чого отримаємо певне числове значення, що і буде підписом користувача саме цього тексту.

$$signature \coloneqq sign(sk, message)$$

3) Для перевірки підпису необхідно мати публічний ключ користувача, який підписував (pk), початкове повідомлення (message) та підпис (signature).

$$isValid \coloneqq verify(pk, message, sig)$$

Алгоритми підпису із асинхронним ключем реалізовані таким чином, що знаючи відкритий ключ (pk), початкове повідомлення (message) та підпис (signature) неможливо дізнатися закритий ключ.

Під терміном “неможливо” мається на увазі, що час який необхідний щоб підібрати відповідний ключ незмірно великий.

Таким чином підпис відповідає конкретному користувачу і конкретному тексту. Якщо внести хоч якісь зміни до початкового тексту, то значення підпису зміниться.

# 3. Побудова децентралізованої СКД

У ході роботи було розроблено різні системи контролю доступу, кожна з яких має свої переваги та недоліки. У наступних підрозділах будуть описуватися різні розроблені СКД у порядку схожості до існуючих.

## 3.1. Надсилання запрошень в централізованій СКД

Почнемо з ситуації, розглянутої вище (у розділі 2.2), де потрібно надати дозвіл людині з 2-го поверху на 5 поверх. Нехай на 5 поверсі працює Аліса, а на 2 поверсі Боб.

Кожен працівник має свій номер за яким визначаються права доступу. Тобто ми можемо вважати, що це є їхні відкриті ключі (якщо вводити асиметричне шифрування). І кожен користувач знає ще свій закритий ключ. Тоді якщо Аліса хоче запросити Боба до себе, то вона генерує повідомлення-запрошення (рис. 2) в якому міститься відкритий ключ Аліси, відкритий ключ Боба, двері на які поширюється запрошення та час на який дійсне запрошення. У кінці повідомлення Аліса ставить цифровий підпис.

- Я: [відкритий ключ Аліси]
- Надаю доступ: [відкритий ключ Боба]
- На двері: [ліфт на 5 поверх, двері в кабінет Аліси]
- На час: [час на коли вона хоче щоб прийшов Боб]

Підпис Аліси тексту вище

Рисунок 2 – Запрошення

Після цього Аліса надсилає згенероване запрошення Бобу. Боб підходить до ліфта і показує зчитувачу це повідомлення, а також підтвердження, що він Боб (наприклад ставить підпис на якомусь випадковому тексті). Контролер надсилає повідомлення центральному комп'ютеру, який перевіряє чи має Аліса доступ до зазначених дверей, чи правильний час, чи правильний підпис. Якщо перевірка пройдена, то двері відкриваються.

Таким чином центральний комп'ютер може залишатися без доступу до інтернету, а Аліса може запрошувати Боба і надавати йому доступ до певних дверей.

## 3.2. Рекурсивне надання доступу

В попередньому пункті було проілюстровано можливість надавання доступу іншій людині. Тепер можна розвинути цю ідею і реалізувати можливість послідовного надання доступу. Припустимо що Аліса хоче організувати нараду з відділом маркетологів. Боб – голова відділу маркетологів і в нього є помічник Карл, який теж повинен бути на зустрічі.

Як і раніше, Аліса надсилає запрошення Бобу, а Боб тепер надсилає запрошення Карлу (рис. 3). У запрошені Боба Карлу міститься запрошення, яке отримав Боб від Аліси, щоб підтвердити, що Боб має доступ. Далі записується аналогічна інформація про те що тепер Боб надає доступ Карлу. В кінці Боб ставить свій підпис на всьому отриманому повідомленні.

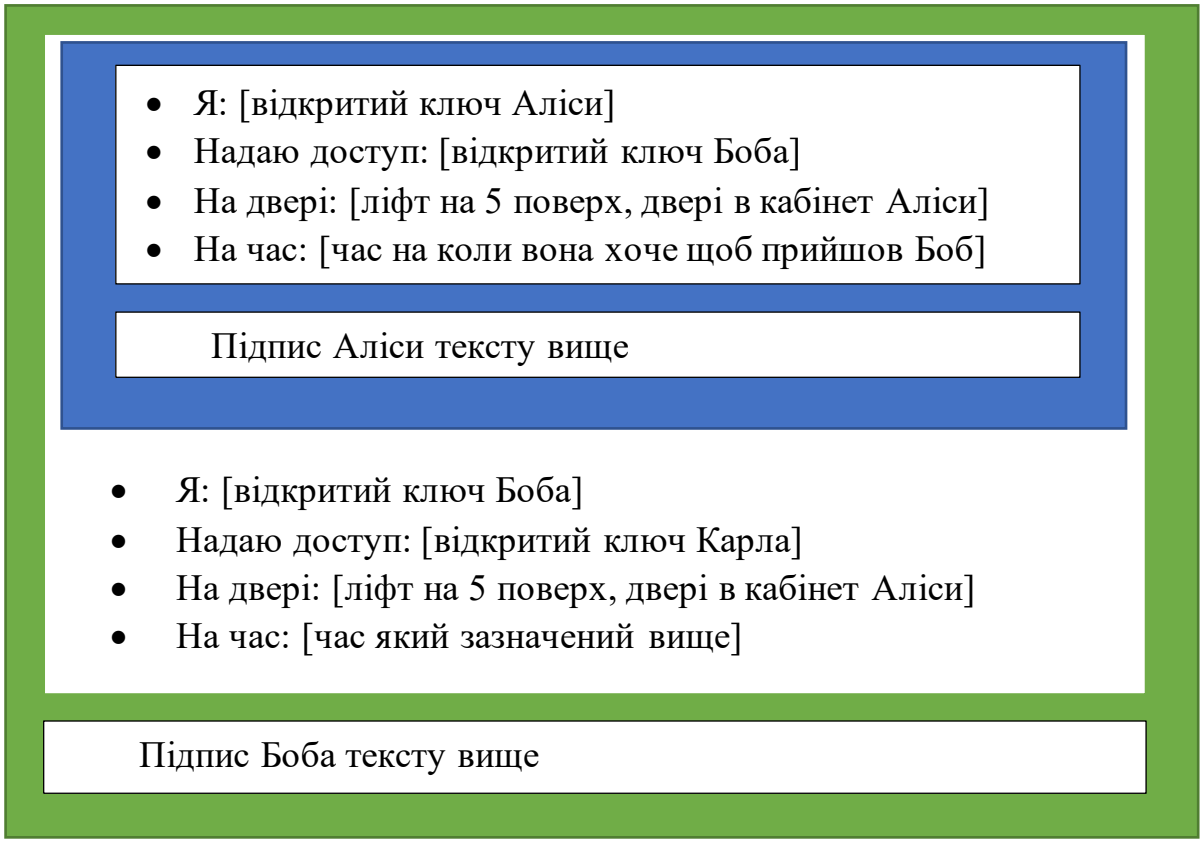



Рисунок 3 – Рекурсивне запрошення

Комп'ютер рекурсивно перевіряє всі данні, а також звіряє обмеження на час та перелік дверей. Обмеження повинні зменшуватися, або залишатися такими ж.

Не важко помітити, що у даному випадку Карл може аналогічно надати доступ іншій людині і тд. Для уникнення подібної ситуації вводимо нове обмеження на глибину повідомлення. Глибина повідомлення – ціле число, яке означає максимальну кількість перенаправлень запрошення після введення цього обмеження. Користувач, який є в базі доступу може створювати запрошення з глибиною яка дорівнює нескінченності. Кожен користувач, який отримав запрошення з обмеженням на глибину n, може надати доступ іншому користувачу, але зі значенням обмеження на глибину, яка менша або дорівнює n-1.

Таким чином якщо Боб отримає від Аліси запрошення з обмеженням на глибину, яке дорівнює 1, то він зможе надати доступ Карлу тільки з обмеженням на глибину, яке дорівнює 0 (рис.4). Карл уже не зможе нікому надати доступ.

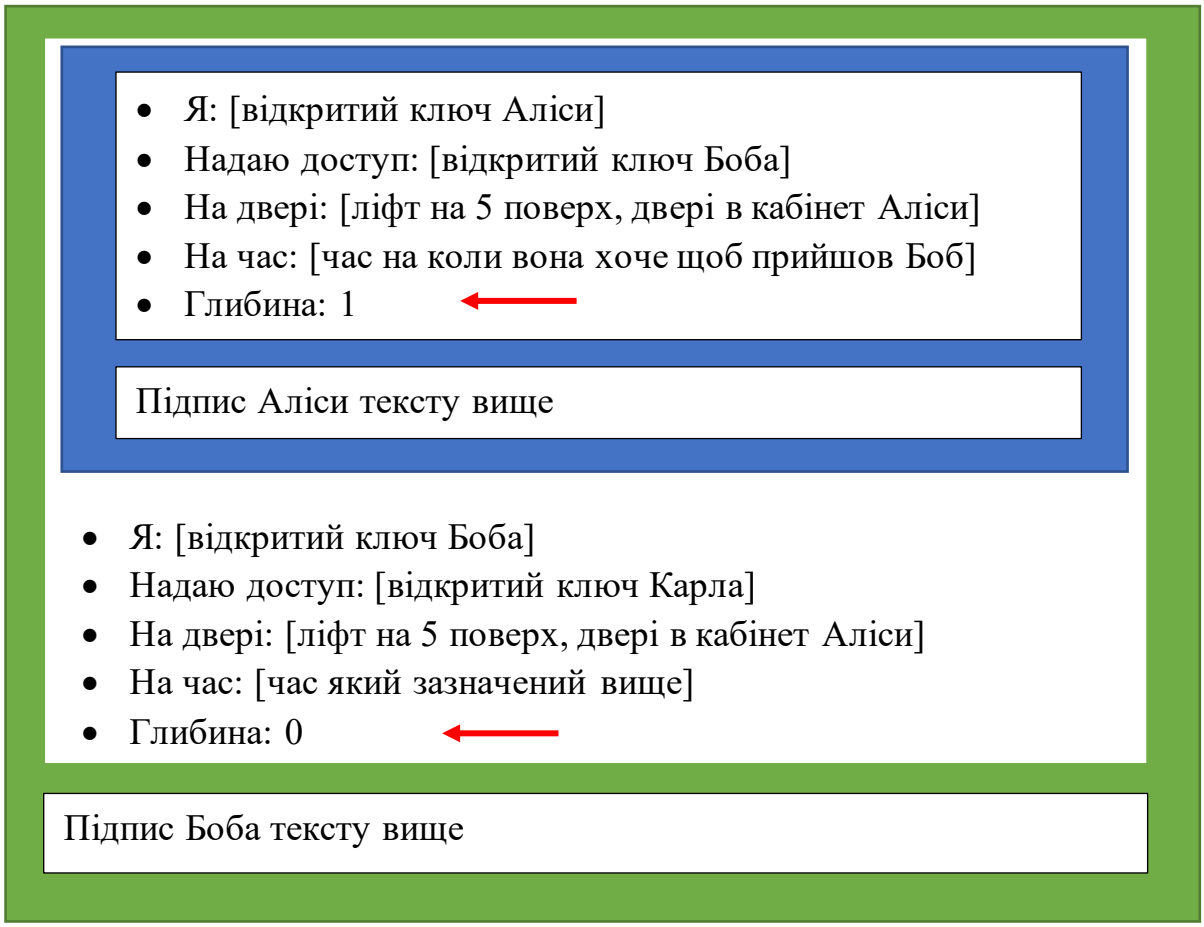



Рисунок 4 - Запрошення з обмеженням на глибину

Коли Карл показує повідомлення зчитувачу, то система перевіряє ще і обмеження на глибину. Якщо задовольняються описані вище правила, то двері відчиняються.

## 3.3. Децентралізована СКД

Тепер опишемо СКД без центральної бази дозволів. Контролер біля дверей буде генерувати свої закритий та відкритий ключі. При встановленні системи необхідно задати власника цих дверей. Власник буде мати повний доступ до цих дверей. Для процесу ініціалізації достатньо лише відкритого ключа власника.

У прикладі з компанією, нехай власником буде голова компанії (ГК). Він буде мати доступ до всіх дверей у своїй компанії. Після ініціалізації контролер генерує повідомлення з дозволом для власника (рис. 5). У повідомленні міститься відкритий ключ дверей та відкритий ключ власника. Обмеження на час та глибину набувають значень нескінченність, тобто дозвіл поширюється на будь-який час та на будь-яку глибину.

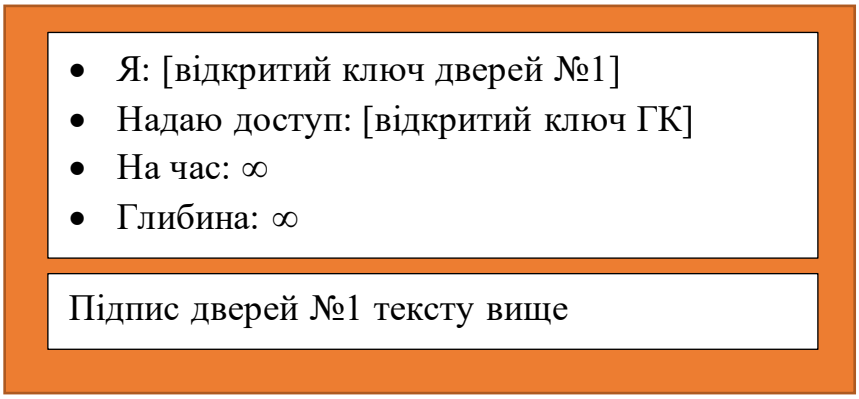



Рисунок 5 – Стартовий дозвіл

Для оптимізації можна вважати, що якщо не вказано обмеження на час і глибину, то їхні значення нескінченність (рис. 6).

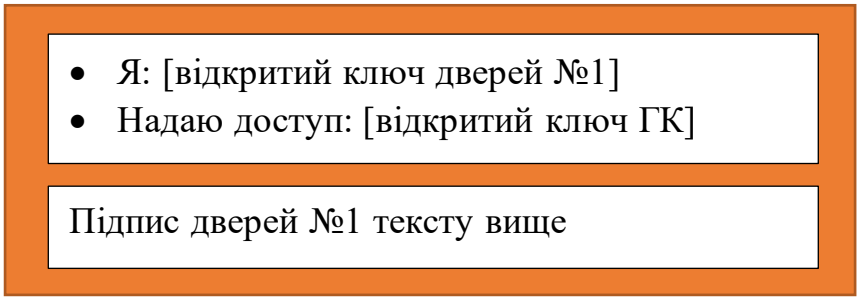



Рисунок 6 – Стартовий дозвіл (скорочений)

Далі ГК може надавати доступи керівникам відділів, а ті в свою чергу працівникам. Таким чином надається доступ до всіх дверей ієрархічно. І кожен

користувач має перелік повідомлень, які дозволяють йому входити у необхідні двері. Для кожних дверей користувач має окреме повідомлення в якому міститься ланцюжок дозволів починаючи від дозволу дверей власнику і аж до самого користувача.

Для ілюстрації поширення дозволів припустимо, що у певної компанії є два відділи (рис. 7): IT та маркетинг. Аліса є крівником відділу IT, а Боб керівником відділу маркетингу. Під керівництвом Боба є два працівника, один з них це Карл.

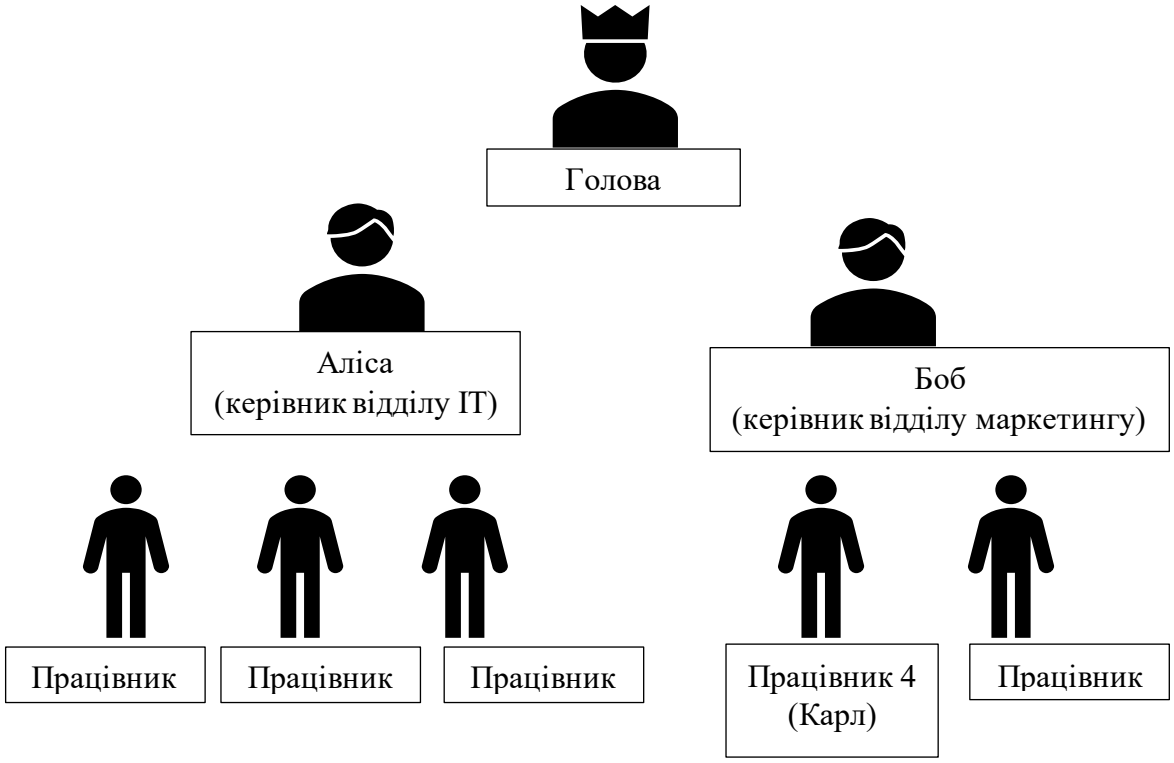



Рисунок 7 – Ієрархія в компанії

У такій ситуації для того щоб Карл зайшов у свій кабінет (наприклад двері №4), він повинен мати повідомлення, яке складається з дозволу дверей № 4 голові компанії, потім дозволу голови компанії Бобу, і врешті Боба Карлу (рис. 8).

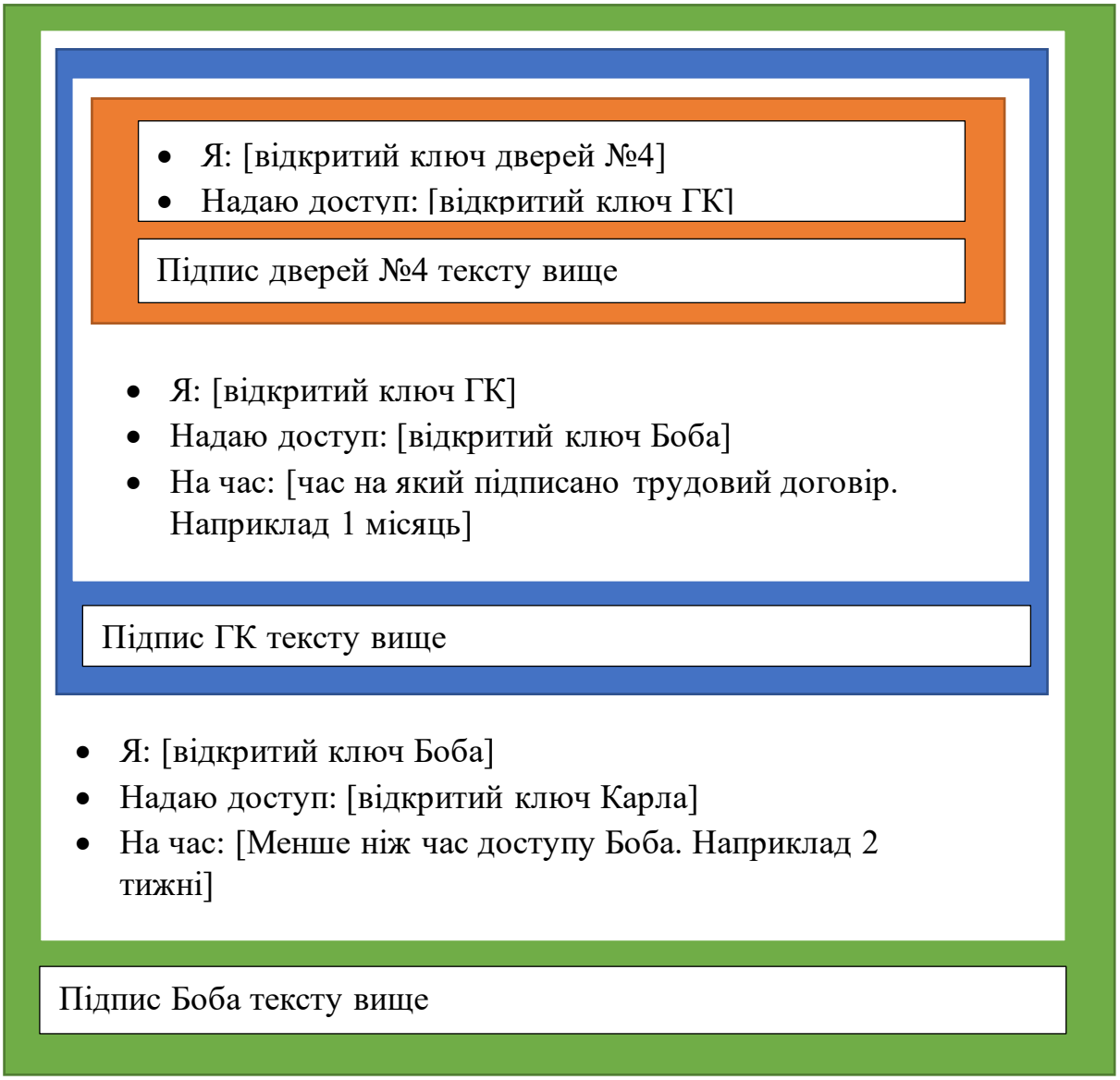



Рисунок 8 – Дозвіл на вхід у децентралізованій СКД

Для перевірки цього повідомлення, контролер дверей повинен пам’ятати тільки свій відкритий ключ і вміти перевіряти підписи. Таким чином кожні двері можуть самостійно рекурсивно перевірити ланцюжок доступу і у позитивному випадку відкрити двері. Для перевірки не потрібно мати зв’язок з якоюсь базою даних чи взагалі зв’язку із зовнішнім світом через інтернет. Вся інформація для підтвердження входу міститься в повідомленні.

## 3.4. Реалізація обміну повідомленнями між користувачем та контролером

Розглянемо реалізацію обміну повідомленнями між користувачем та контролером біля дверей. Користувач спілкується з контролером через зчитувач. Для описуваної системи потрібен двосторонній зв’язок між користувачем та контролером, а також він повинен мати середню пропускну здатність, щоб швидко надсилати повідомлення з доступом. Таким вимогам відповідає зв’язок через Bluetooth. Тоді у користувача є телефон з функцією Bluetooth (ідентифікатор), а зчитувач це модуль Bluetooth у контролері. Для спрощення будемо вважати, що користувач спілкується напряму з контролером. Коли користувач підходить до дверей, то йому необхідно дізнатися відкритий ключ саме цих дверей (контролера), щоб знати яке повідомлення з доступом відправити. Потрібно провести процес аутентифікації користувача для контролера, а також аутентифікацію контролера дверей для користувача (щоб користувач був впевнений, що це не зловмисний пристрій). При цьому необхідно мінімізувати кількість повідомлень між контролером та користувачем, адже для комфортного користування СКД потрібно, щоб час пропуску був мінімальний.

Ініціалізувати процес повинен користувач, оскільки він підходить до дверей (рис. 9).

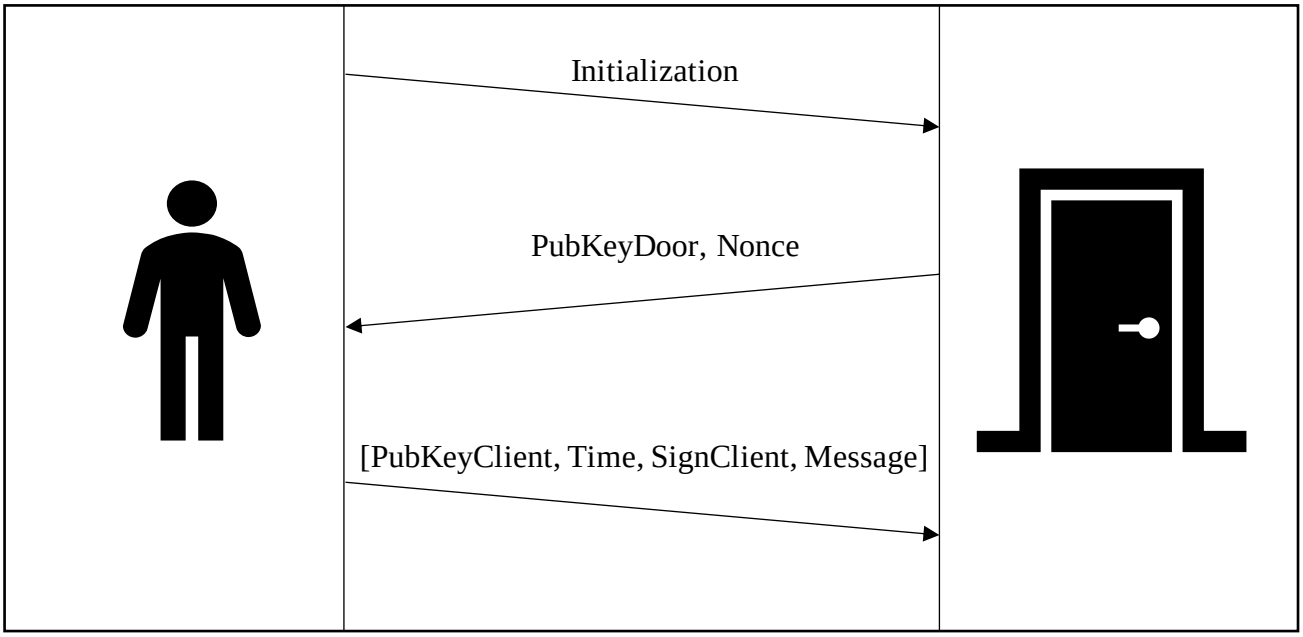



Рисунок 9 – Обмін повідомленнями між користувачем та дверми

Після того, як контролер отримав стартове повідомлення, він відправляє повідомлення, яке містить відкритий ключ дверей (PubKeyDoor) та випадково згенерований текст (Nonce).

Користувач перевіряє, чи має він необхідний дозвіл для цих дверей. У позитивному випадку він формує повідомлення з свого публічного ключа (PubKeyClient), публічного ключа дверей (контролера) (PubKeyDoor), поточного часу (Time) і випадково згенерованого тексту, отриманого від дверей (контролера) (Nonce) і підписує його (SignClient), як показано у рівнянні (1).

$$SignClient = sign_{PrivKeyClient}(PubKeyClient, PubKeyDoor, Time, Nonce) \quad (1)$$

Після цього користувач формує повідомлення зі свого відкритого ключа (PubKeyClient), часу, який він записав раніше (Time), підпису (SignClient) і повідомлення, яке надає дозвіл на відкриття (Message). Користувач шифрує повідомлення за відкритим ключем контролера дверей (PubKeyDoor) і надсилає.

Таким чином контролер дверей зможе прочитати повідомлення тільки якщо він справді має закритий ключ. Тобто відбулася аутентифікація контролера дверей.

З іншого боку, користувач підписав повідомлення, частину з якого надіслав контролер. Отже контролер знає всі компоненти повідомлення, яке підписав користувач і може перевірити валідність підпису. Тим самим відбувається аутентифікація користувача.

Крім цього у контролера залишається підпис користувача повідомлення, яке підтверджує, що клієнт намагався зайти саме у цей час, у ці двері. Ця інформація може зберігатися у контролера для збереження історії входу.

Наступним кроком контролер перевіряє частину повідомлення з доступом. Якщо вона відповідає усім вимогам, то двері відчиняються.

## 3.5. Історія СКД

В попередньому розділі було розглянуто те що контролер дверей має інформацію, яка підтверджує те що конкретний користувач намагався зайти у конкретні двері у конкретний час. Проте раніше було означано, що контролер дверей не підключений до інших пристроїв, а тому інформація про вхід у двері залишається локально. У деяких випадках інформація про вхід у двері може бути корисною. Тому у цьому розділі розглянемо СКД, яка працює так само, як і розглянута у попередньому розділі, але у якій контролери дверей підключені до мережі інтернет.

Вважатимемо, що є певне сховище куди мають доступ усі. Це може бути наприклад хмарне сховище, до якого будуть підключені усі контролери, а також зможуть зайти (але лише читати) користувачі. Контролери будуть надсилати інформацію про те хто зайшов у зашифрованому вигляді. Так щоб розшифрувати можна було 1 з n закритих ключів (тих людей які надали доступ цій людині). Тобто

коли захоче зайти Карл до своєї кімнати, надішлеться запис який зможуть розшифрувати тільки Карл, Боб та голова компанії. Якщо ж зайде Боб, то про це дізнається тільки він і голова компанії. Аліса ж не зможе розшифрувати ці повідомлення.

Таким чином Голова компанії зможе побачити все що відбувається в його компанії. Керівники відділу зможуть побачити лише те що відбувається у їхніх підлеглих, але не зможуть побачити те що відбувається у підлеглих іншого відділу.

Отже топологія СКД на даний момент зображена на рисунку (2). Сині стрілки ілюструють передачу дозволів між користувачами, а також двері куди з цими дозволами можна зайти. Присутня властивість транзитивності (якщо користувач А надав доступ користувачу В, а В має доступ до дверей, то і А має доступ до цих дверей). Жовті пунктирні стрілки вказують на інформацію, яку надають контролери дверей про те хто зайшов.

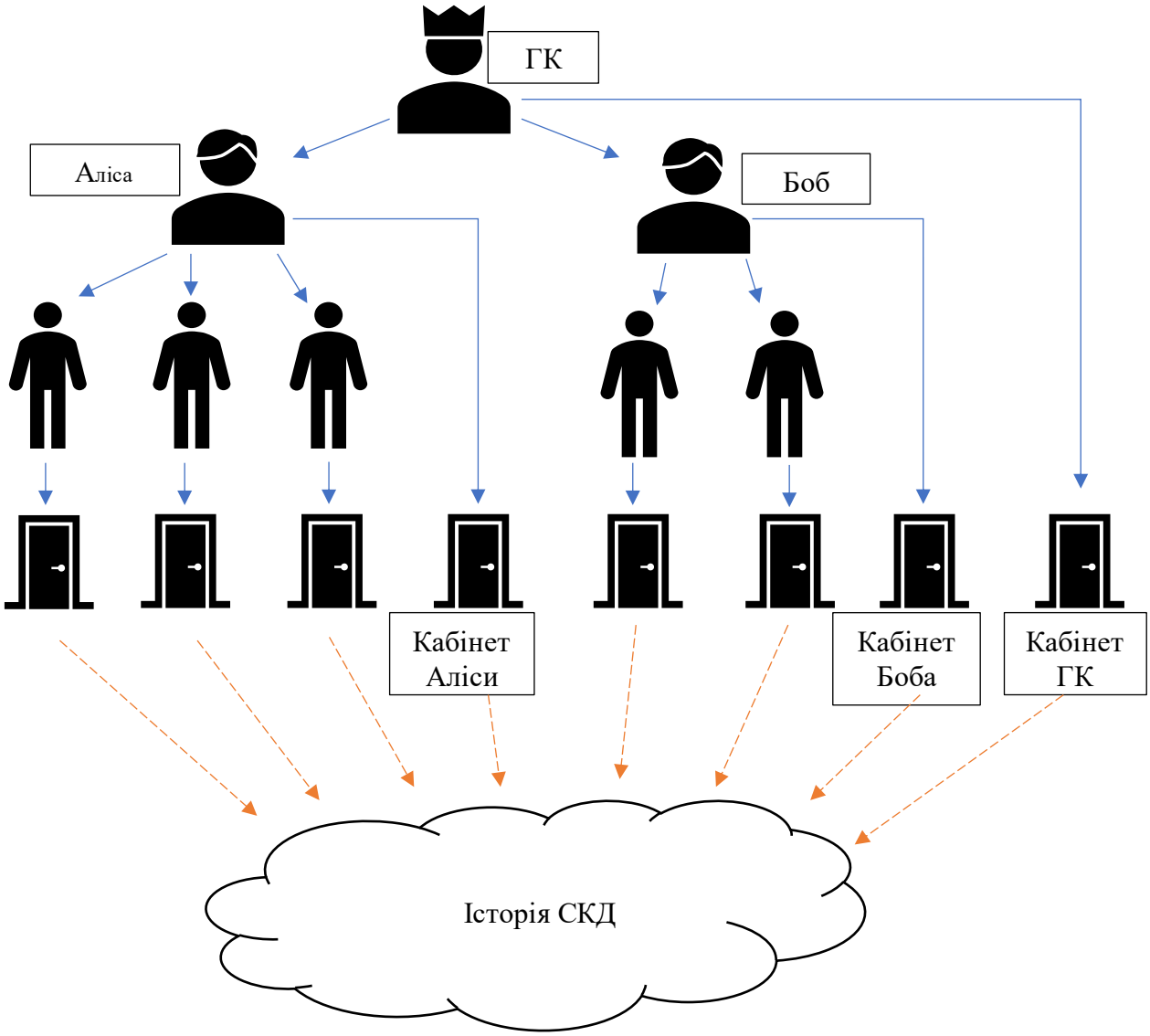



Рисунок 10 – Топологія СКД з історією

## 3.6. NFC картки

Описані вище архітектури передбачають використання телефону. Часто це зручно, оскільки не потрібно думати де картка. Використання картки не можливе в описаній конфігурації, адже потрібно постійно отримувати нові запрошення, та і розмір запрошень надто великий.

Тим не менше можна зробити наступні вдосконалення в СКД для можливості використання NFC карток. Для цього також необхідний доступ до інтернет.

Користувач, після отримання дозволу на відкривання дверей, може згенерувати дозвіл на ім’я своєї NFC картки. Та надіслати його контролеру дверей у зашифрованому вигляді. Тоді контролери формують свою базу дозволів і коли до них підходить користувач з відповідною карткою, то двері відчиняються.

Проте така СКД втрачає свої переваги у надійності та відсутності з’єднання контролерів до зовнішнього світу. Також у разі використання карток не можливо отримати потрібного підпису під час відкривання дверей, який би підтверджував, що була спроба відкривання дверей.

## 3.7. Типи обмежень

У СКД описаних вище ми описували типи обмежень пов’язані з часом та глибиною. У цьому розділі ми розглянемо детальніше типи обмежень, а також наведемо інші.

### *3.7.1. Час*

У описаних СКД не можливо забирати доступ у користувачів, які уже його мають. Тобто якщо Аліса надіслала запрошення Бобу, то забрати його у Боба уже не можливо. Саме тому необхідно встановлювати обмеження по часу і бажано робити його мінімальним. Тобто Алісі потрібно надавати доступ Бобу, наприклад, на 1 тиждень. Тоді якщо Аліса захоче забрати доступ у Боба, то їй достатньо буде не надсилати нове запрошення через тиждень.

### *3.7.2. Глибина*

Обмеження на глибину повідомлення необхідне щоб унеможливити надсилання запрошень лишнім людям. У випадку якщо користувач хоче надіслати запрошення гостю і користувач не хоче, щоб з гостем прийшов ще хтось, то потрібно вибирати значення 0 глибини повідомлення. Звичайно, що ми не враховуємо випадок, коли за одним пропуском можуть пройти декілька людей.

Якщо ж гість має привести з собою ще якихось людей, то значення обмеження повинно бути 1. Проте варто пам’ятати, що це обмеження не забороняє гостю надсилати запрошення будь-якій кількості людей.

### *3.7.3. Ширина*

Одне запрошення може використовуватися тільки на одні конкретні двері. А тому можна ввести обмеження на кількість користувачів, які можуть проходити за певним запрошенням. Уявимо ситуацію, що в компанії на нараду Аліси повинна прийти група працівників з іншого відділу. Ми знаємо що група складається з організатора та ще 3-х людей. Ми знаємо організатора, проте не знаємо хто саме ті 3 інших працівника. Тоді було б зручно мати змогу обмежити не тільки глибину повідомлення доступу, а і кількість людей яким нададуть доступ у майбутньому. Якщо уявити граф (рис. 11) надання доступу між користувачами, то це виходить обмеження на ширину в під графі.

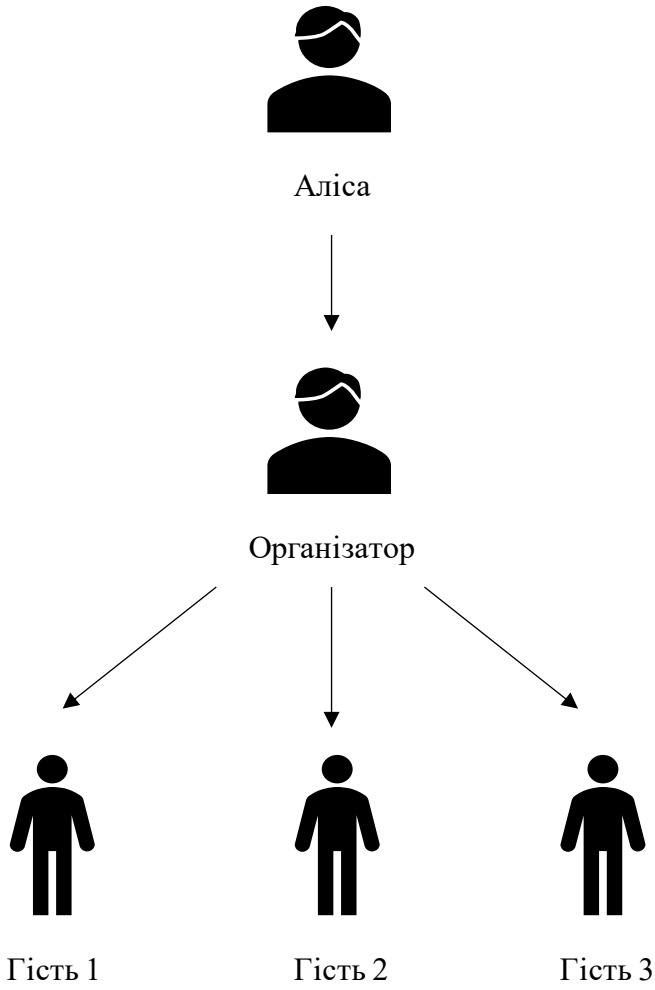



Рисунок 11 – Група гостей

Для такого обмеження контролери дверей повинні запам’ятовувати хто ввійшов уже за запрошенням з обмеженням на ширину.

Повертаючись до ситуації, Аліса генерує запрошення Організатору, де вказує обмеження на ширину, яка дорівнює 3. Тоді Організатор надсилає доступ трьом іншим працівникам. Коли підходить перший з трьох працівників до дверей, то контролер його впускає і запам’ятовує, що за обмеженням на ширину пройшов уже один користувач, а також запам’ятовує відкритий ключ цього користувача. Аналогічно контролер записує собі двох інших користувачів. Записані користувачі можуть повторно входити, якщо це не забороняють інші обмеження. Якщо ж організатор дасть доступ більшій кількості користувачів, то зможуть пройти лише

перші троє. Саме тому якщо користувачу надали запрошення з таким обмеженням, то він не може бути впевнений, що запрошення пройде валідацію.

Окремо варто зазначити випадок, коли Гість 2 надасть доступ Гостю 4 (рис. 12). Для обмеження на ширину Гість 4 рівносильний іншим гостям. А тому якщо він зайде в першій трійці, то його двері впустять, але не впустять того хто прийде четвертим, навіть якщо це Гість 2.

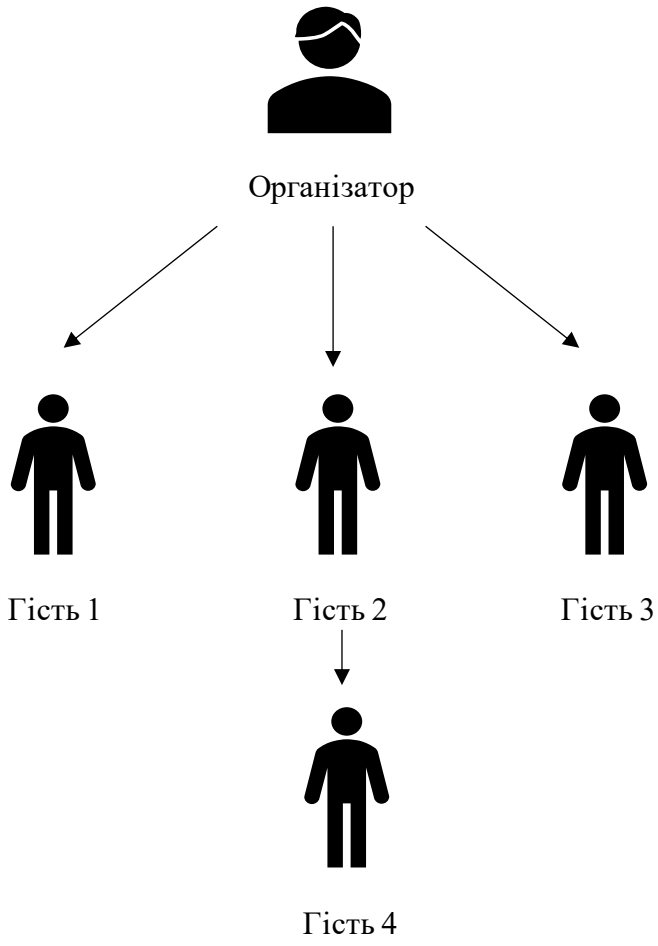
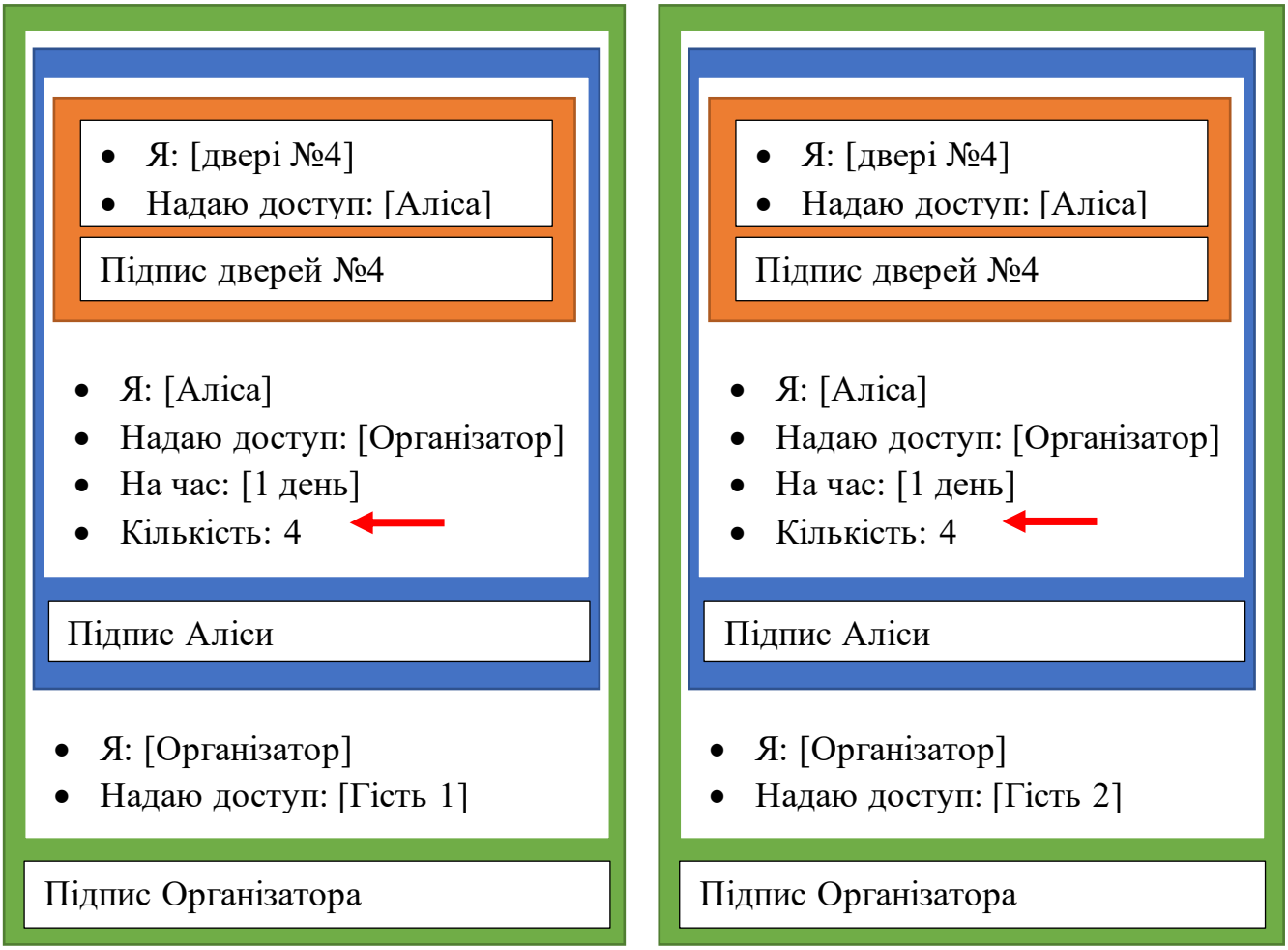



Рисунок 12 – Група з 4-х гостей

Тому варто поєднувати це обмеження з іншими. Наприклад, якщо Організатор зробить обмеження на глибину 0 для гостей, то Гість 4 уже не зможе отримати доступ.

### *3.7.4. Кількість використань*

У деяких ситуаціях може бути зручно встановити обмеження на кількість використань обмеження. Як і обмеження на ширину, воно потребує, щоб контролери дверей записували події, що відбуваються. У ситуації, розглянутій у розділі 3.7.3, де Аліса запрошує групу людей, Аліса може надіслати Організатору дозвіл з обмеженням на кількість використань. Тоді будь-які дозволи, які будуть утворені з даного, будуть враховуватися в це обмеження. На рисунку (13) зображено два дозволи, які отримають Гість 1 та Гість 2. Вважаємо, що Аліса є власником тих дверей, куди потрібно надати дозвіл.




а) б)

Рисунок 13 – Дозвіл з обмеженням на кількість використань: а – Гість 1, б – Гість 2

Коли хтось з користувачів використає одне з запрошень зображених на рисунку (13), то контролер дверей почне перевіряти запрошення рекурсивно згори. Коли перевірка дійде до обмеження на кількість, то контролер перевірить скільки уже було використано запрошень, які мали такий же вміст від початку і до місця застосування обмеження (у нашому випадку це частина запрошення у синій рамці). Таким чином запрошення Гостя 1, Гостя 2 і Організатора рахуються разом. Також варто зазначити, що ця перевірка відрізняється від перевірки на ширину тим, що враховуються повторні використання запрошень. Тобто якщо Організатор зайде в двері 4 рази, то більше ніхто з гостей не зможе зайти.

Як і з обмеженням на ширину, при використанні обмеження на кількість, користувач не може бути впевнений, що двері відкриються, адже не відомо чи ще хтось має доступ з цим обмеженням.

## 3.8. Порівняльна характеристика

У попередніх розділах було запропоновано і описано різні СКД. Зробимо порівняльну характеристику цих СКД. У розділі 1.1 було описано існуючу централізовану СКД, в розділі 3.2 було описано централізовану СКД з додатковою можливістю запрошень. У розділі 3.3 описано децентралізовану СКД, яка не потребує єдиної бази даних дозволів і не потребує підключення між дверями. У розділі 3.5 описано додаткові можливості, які стають доступними при підключенні децентралізованої СКД до мережі інтернет. У таблиці 1 зібрана порівняльна інформація, яка була у відповідних розділах.

| | Централізована СКД | Централізована + запрошення | Децентралізована СКД | Децентралізована + інтернет |
|---|---|---|---|---|
| Відсутність централізованої бази | - | - | + | + |
| Відсутність з’єднань | - | - | + | - |
| Можливість надання запрошень | - | + | + | + |
| Можливість отримання інформації про відкриття дверей | + | + | - | + |

Таблиця 1 — Порівняльна характеристика

# 4. Можливі схеми передач дозволів

Схема передач дозволів у децентралізованій СКД може приймати різного вигляду залежно від ситуації в якій вона використовується. У цьому розділі описуються різні можливі сценарії передач дозволів від дверей і до інших користувачів. Також описуються переваги та недоліки кожного з сценаріїв.

## 4.1. Рекомендована

Цей сценарій уже описувався вище. Власником усіх дверей є одна людина – голова компанії. Він надає дозволи керівникам відділів, а ті у свою чергу надають доступ працівникам, які їм підпорядковуються (рис. 14). На рисунку (14) помаранчевими стрілками позначено надання доступу від дверей власнику, а чорними стрілками – надання доступу між користувачами. При такому сценарії голова компанії точно знає, що всіма дозволами керує він, а також може побачити усю активність у своїй компанії. Недоліками є те що голова компанії, як власник усіх дверей, повинен бути фізично присутнім при ініціалізації кожного з контролерів, що є незручним у випадку великої компанії.

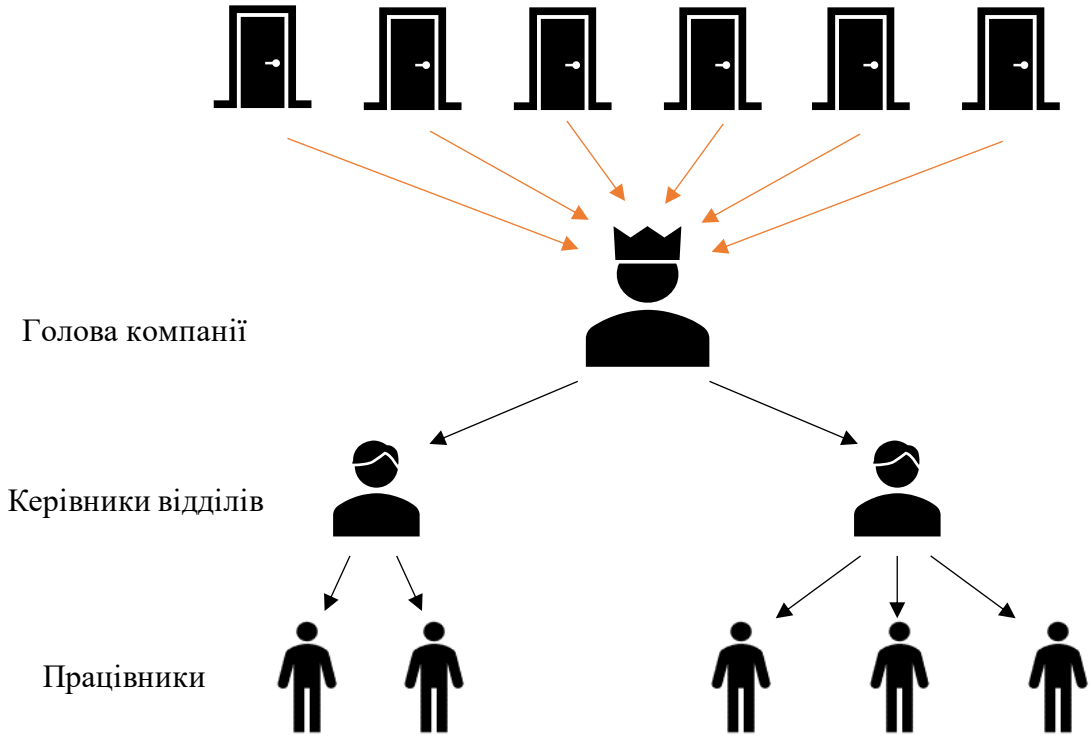



Рисунок 14 – Рекомендована схема передач дозволів

## 4.2. Оптимальна

У великих компаніях може бути соті чи більше дверей, тому логічним є те що пропусками повинна займатися охорона компанії, а не голова компанії. У такому випадку зручною буде ситуація, де власником усіх дверей є спеціальна людина (рис. 15). Ця людина надає усі права доступу голові компанії, який уже в свою чергу ієрархічно роздає дозволи усім іншим працівникам, як на схемі, зображеній вище.

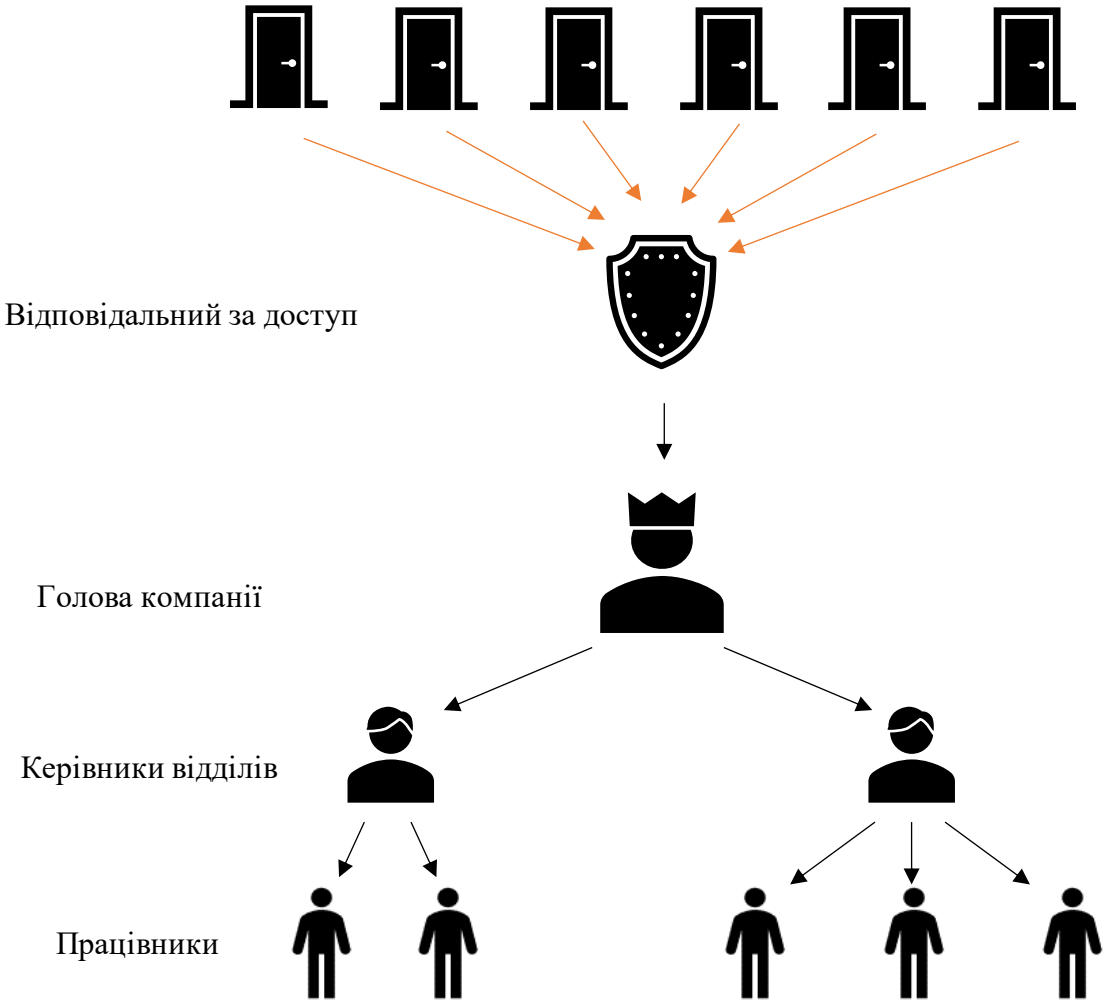



Рисунок 15 – Оптимальна схема передач дозволів

Недоліками такої схеми є те що голова компанії не може знати точно, чи відповідальний за доступ не надав доступ стороннім людям. Проте такої впевненості немає і при використанні традиційних СКД.

## 4.3. Централізована

Запропонована технологія дуже гнучка. За допомогою неї можна відтворити навіть схему надання дозволів у існуючих централізованих СКД (рис. 16). У таких СКД є одна центральна точка, яка напряму надає доступ усім користувачам.

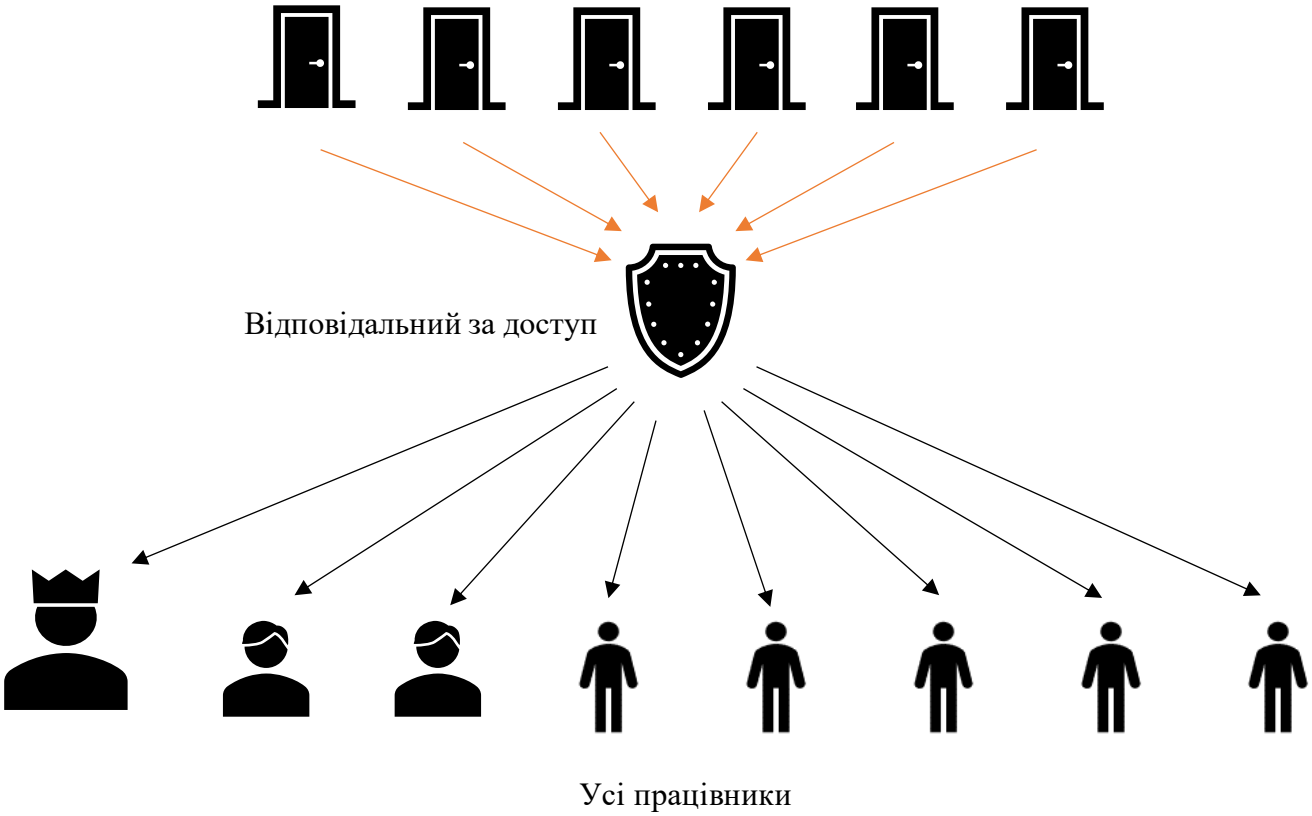



Рисунок 16 – Централізована схема передач дозволів

Така схема має ряд недоліків, які мають традиційні СКД:

1) Весь доступ знаходиться в однієї людини, яка повинна постійно керувати одночасно усіма доступами.
2) Керівники відділів не можуть напряму побачити активність своїх підлеглих. Вони це можуть зробити лише через відповідального за доступ.

Проте, на відміну від традиційних СКД, така система дає можливість надавати запрошення між працівниками. Перевагами використання такої схеми є простота у використанні для працівників, адже керівникам відділів не потрібно контролювати надання доступів для своїх підлеглих.

Описана схема передач дозволів може бути зручною у використанні для готелів. Де є один адміністратор, який має доступ до всіх дверей. Коли поселяється людина у номер, то їй надають дозвіл на двері в номері на період проживання.

Клієнт може надати дозвіл іншим людям які з ним живуть. Коли приходить час здачі номера, то дозвіл перестає бути дійсним.

## 4.4. Багатоквартирний будинок

Запропонована СКД може використовуватися не тільки в офісних центрах, а і в житлових будинках. Кожен мешканець є власником своїх дверей, а охоронець є власником вхідних дверей у будинок і надає доступ до вхідних дверей кожному мешканцю (рис. 17). Якщо ж до якогось з мешканців приходить гість, то мешканець надсилає тимчасовий доступ до своїх дверей і до вхідних дверей гостю. Коли людина заходить у вхідні двері будинку, то охоронець (як власник) отримує повідомлення за яким увійшла людина і може побачити хто надав дозвіл та зрозуміти у яку квартиру іде ця людина.

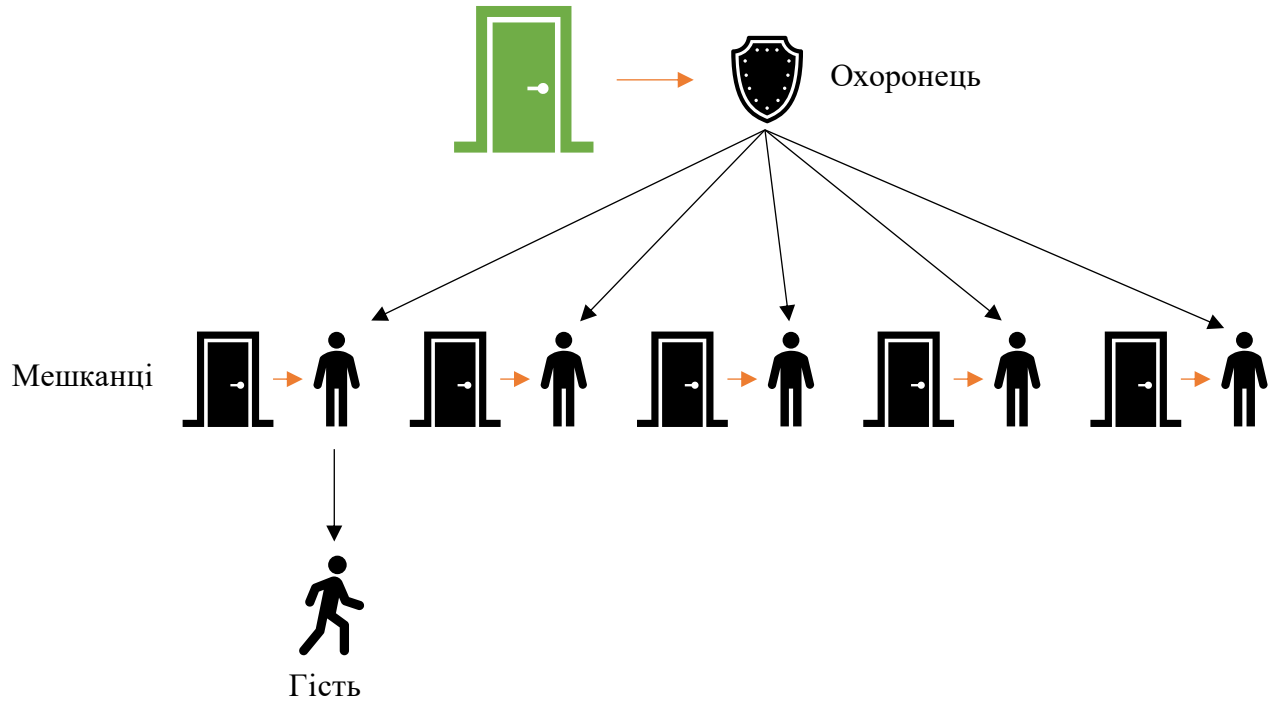



Рисунок 17 – Схема передач дозволів у багатоквартирному будинку

Перевагами такої системи є:

1) Мешканці можуть надсилати запрошення гостям та бачити коли гості заходять у будинок.

2) Охоронець бачить в яку квартиру іде людина.

Недоліками такої системи є те що гостю потрібно встановити мобільний додаток, щоб потрапити у будинок. Проте ніщо не заважає поєднувати цю систему з відомими домофонами чи іншими існуючими системами.

# 5. Програмна реалізація

В процесі роботи було створене програмне забезпечення, яке демонструє роботу децентралізованої СКД. Також у розробленій програмі можна моделювати описані вище схеми передач доступів та відслідковувати вміст повідомлень з доступами у кожного з користувачів. Ще однією можливістю є моделювання дверей і перевірки доступів користувачів.

Для криптографічних протоколів вибрана еліптична криптографія. Для цифрових підписів використовується ECDSA, а для шифрування ECDH. Програма написана на мові програмування Python. Для графічного інтерфейсу використана бібліотека Tkinter.

## 5.1. Головне вікно

У головному вікні є можливість створення користувача та двері (рис. 18).

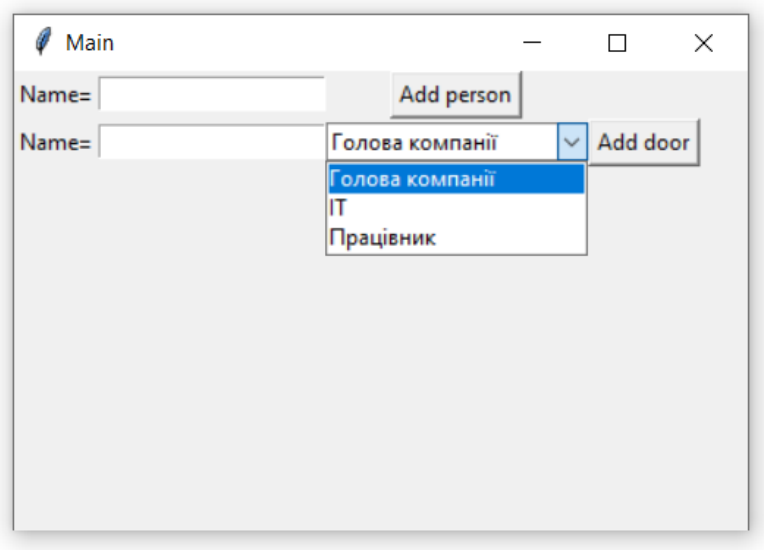



Рисунок 18 – Головне вікно

Для створення користувача потрібно ввести ім’я, яке буде закріплено за цим користувачем. Після натиснення кнопки «Add person» згенеруються відкритий та закритий ключі та створиться об’єкт класу Person.

Для створення нових дверей необхідно ввести назву дверей та вибрати з випадаючого списку користувача, який буде власником цих дверей. Після натиснення кнопки «Add door» згенеруються відкритий та закритий ключі, створиться об’єкт класу Door та надішлеться повідомлення з доступом власнику.

## 5.2. Вікно користувача

Кожному користувачу у симуляції відповідає окреме вікно (рис. 19). У лівому верхньому куті вікна вказується ім’я користувача. Нижче знаходиться рядок для надання доступу іншому користувачу. Ще нижче знаходиться рядок для здійснення спроби зайти в двері. Внизу вікна розміщується поле для відображення поточних дозволів, які має користувач.

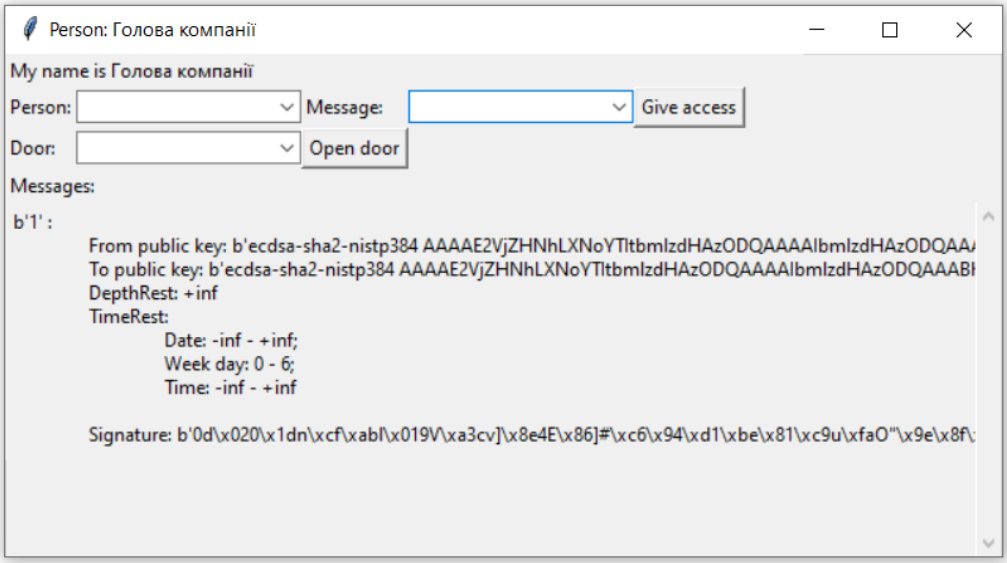



Рисунок 19 – Вікно користувача (Голова компанії)

Для надання доступу іншому користувачу потрібно з випадаючого списку вибрати ім’я користувача, кому потрібно надати доступ. Аналогічно потрібно вибрати повідомлення яке потрібно поширити. Після натиснення кнопки «Give access» відкриється нове вікно де можна налаштувати обмеження, з якими буде надаватися дозвіл.

Щоб здійснити спробу відкриття дверей, необхідно з випадаючого списку вибрати двері і натиснути кнопку «Open door». Після цього надішлеться запит дверям. Якщо запит пройде успішно, то у вікні дверей появиться запис про те що двері відкрилися для цього користувача. Якщо ж запит не пройде перевірку, то появиться запис про відхилення доступу.

На рисунку (19) зображено вікно користувача Голова компанії. Він є власником дверей «1». Тому Голова компанії має повідомлення від дверей «1» яке складається з відкритого ключа дверей, відкритого ключа Голови компанії, обмеження на глибину яка дорівнює нескінченності (без обмежень) та обмежень на час, який вказує на те що за цим запрошенням двері відкриються в будь-який день, в будь-який час. Вкінці повідомлення стоїть підпис дверей над усім повідомленням.

## 5.3. Вікно налаштування обмежень

При натисканні кнопки «Give access» відкривається вікно налаштування обмежень (рис. 20). Спершу розташований блок обмежень на час. Після нього блок обмежень на глибину. Вкінці знаходиться блок для перевірки результату та відправки повідомлення користувачу.

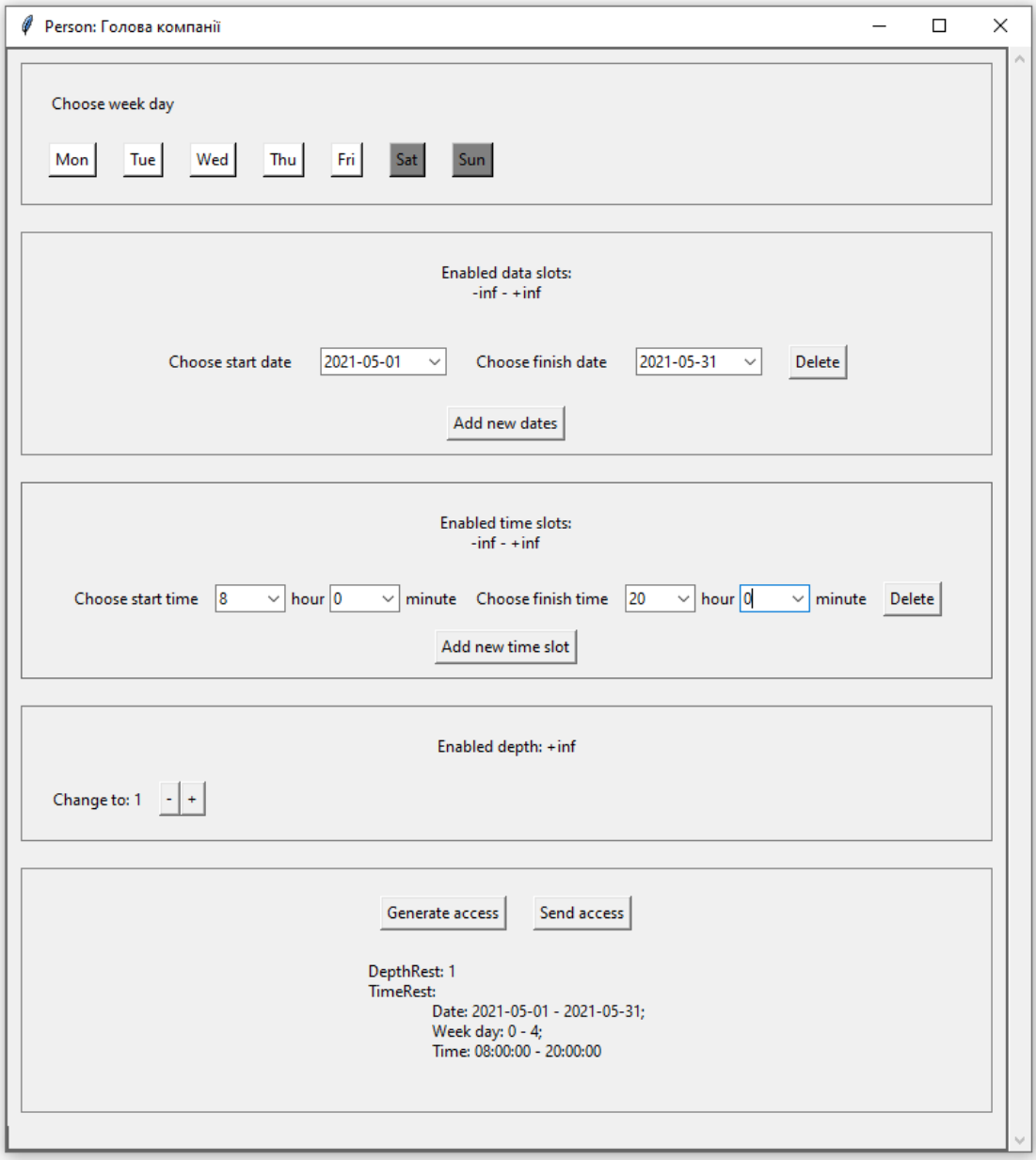



Рисунок 20 – Вікно налаштування обмежень

Блок обмеження часу складається з:

1) Вибору днів тижня.
2) Вибору проміжків дат, де буде діяти дозвіл.
3) Вибору проміжків часу доби, коли буде діяти дозвіл.

У блоках записано поточні обмеження користувача, який намагається надати доступ.

На рисунку (20) зображено як користувач Голова компанії намагається дати доступ користувачу Працівник на двері «1». Оскільки Голова компанії є власником цих дверей, то у нього немає обмежень на використання дверей. Тому у кожному блоці зазначено, що поточні обмеження нескінченні.

Також на рисунку (20) можна побачити, що Голова компанії при створенні запрошення для Працівника накладає обмеження:

1) Дозвіл на дні тижня від понеділка до п’ятниці.
2) Дозвіл на дати з 01.05.2021до 31.05.2021.
3) Дозвіл на час з 8:00 до 20:00.
4) Обмеження на глибину 1. Тобто Працівник зможе надати дозвіл ще одній людині, а та у свою чергу уже нікого не зможе запросити.

## 5.4. Вікно дверей

Кожним дверям у симуляції відповідає окреме вікно (рис. 21). Зліва зверху знаходиться назва дверей. Нижче записаний відкритий ключ дверей. Ще нижче знаходиться поле для записів про користувачів, які намагаються зайти. При зверненні користувача перевіряється його повідомлення з доступом. Після перевірки записується одне з двох значень Access або Denied якщо перевірка пройдена, або ні відповідно. Після цього записується ім’я користувача, який

намагався увійти. Варто зазначити, що двері не мають інформації про назви користувачів, а знають тільки відкриті ключі. Імена записуються для зручнішої демонстрації. Врешті записується відкритий ключ людини, яка намагалася зайти.

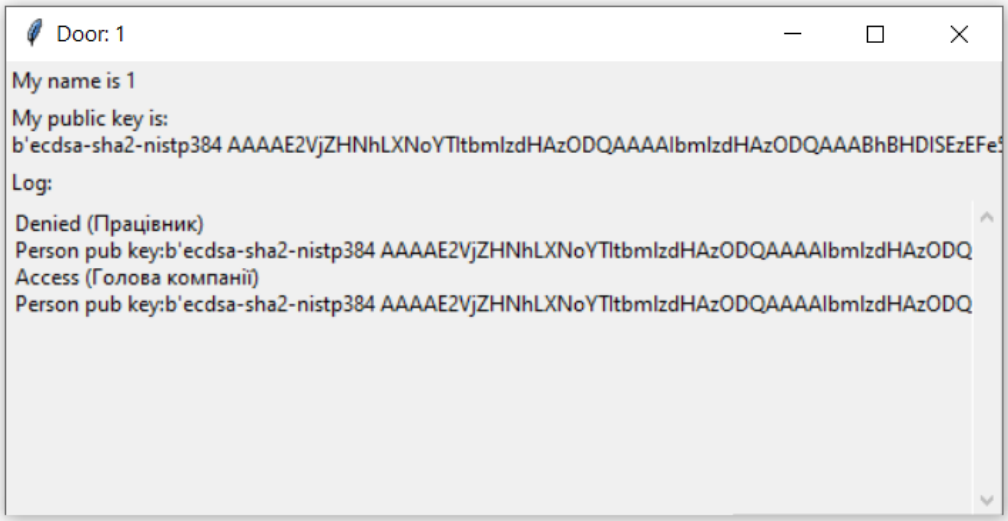



Рисунок 21 – Вікно дверей («1»)

На рисунку (21) зображено вікно дверей з назвою «1». У записах видно, що спершу намагався зайти користувач Працівник, але його запрошення виявилося не дійсним. Пізніше намагався зайти Голова компанії. Його запрошення пройшло перевірку і двері впустили цього користувача.

# ВИСНОВОК

У ході роботи було спроектовано та детально описано принцип роботи децентралізованої системи контролю фізичного доступу, а також різні її модифікації. Було проаналізовано переваги та недоліки кожної з модифікацій та зроблено порівняльну характеристику з традиційними системами контролю доступу. В результаті чого можна зробити висновок, що існують ситуації у яких перевагу має використання запропонованих систем над традиційними.

Було описано декілька сценаріїв використання децентралізованої системи контролю доступу. Проведено розбір конкретних ситуацій та як їх можна вирішити завдяки децентралізованій системі. Для кожної ситуації проаналізовано переваги запропонованої системи над традиційною.

Розроблено програмне забезпечення, яке демонструє принцип роботи децентралізованої системи контролю доступу. У системі є можливість змоделювати будь-який описаний в роботі сценарій. Під час моделювання можна побачити передачу пакетів інформації між учасниками системи, що дозволяє повністю зрозуміти принципи за якими працює запропонована система.

Кількість об’єктів на яких необхідна система контролю доступу буде зростати, оскільки необхідність у офісних центрах та багатоквартирних будинках буде завжди. Тому доцільним є вдосконалення та створення нових альтернативних систем контролю доступу, які б змогли вирішувати проблеми, які не покривають традиційні системи.

Наступим кроком для розвитку децентралізованих систем контролю доступу є створення мобільних додатків та контролерів, які б могли працювати за описаними правилами. Також це необхідно для перевірки дієздатності системи у реальних ситуаціях.

# ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. // Система контролю і управління доступом: [сайт]. URL: https:// uk.wikipedia.org/wiki/Система_контролю_і_управління_доступом (дата звернення: 20.04.2021).

2. Семерак ДО, "СТРУКТУРА БІТКОЇН МЕРЕЖІ," КНУ ім. Тараса Шевченка, Київ, Курсова робота 2020.

3. // Access control topologies serial controllers: [сайт]. URL: https:// en.wikipedia.org/wiki/Access_control#/media/ File:Access_control_topologies_serial_controllers.png (дата звернення: 20.04.2021).