Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ДОПУСТИТИ ДО ЗАХИСТУ:
завідуюча кафедри кібербезпеки
та захисту інформації
_____Наталія ЛУКОВА-ЧУЙКО
«14» червня 2022р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

дипломної роботи
бакалавра
(назва освітнього ступеня)

галузь знань 12 Інформаційні технології
(шифр і назва галузі знань)

спеціальність 125 Кібербезпека
(код і назва спеціальності)

освітня програма Кібербезпека
(назва освітньої програми)

на тему: “Розробка безпечного механізму автентифікації з використанням QR-коду”

Виконавець: студент 4 курсу, групи КБ-42

_____ **Дмитро ЖЕБРАК**
(підпис) (ім’я прізвище)

	Прізвище, ініціали	**Підпис**
Керівник	Андрій ФЕСЕНКО	

Нормоконтроль	Сергій ДАКОВ	

Київ 2022

Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ЗАТВЕРДЖЕНО:
завідуюча кафедри кібербезпеки
та захисту інформації
_____Наталія ЛУКОВА-ЧУЙКО
«01» листопада 2021 р.

ЗАВДАННЯ
на виконання дипломної роботи

спеціальності 125 Кібербезпека
(код і назва спеціальності)

освітньої програми Кібербезпека
(назва освітньої програми)

Студентові **КБ-42** **Жебрак Дмитро Віталійович**
(група) (прізвище ім'я по-батькові)

Тема дипломної роботи Розробка безпечного механізму автентифікації з використанням QR-коду

1. ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Тема дипломної роботи затверджена на засіданні кафедри кібербезпеки та захисту інформації протокол №5 від 29.10.2021 р.

2. ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Автентифікація та її види, методи проходження автентифікації, шифрування QR-коду, механізми автентифікації з використанням QR-коду

3. ЗМІСТ РОЗРАХУНКОВО-ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

Проаналізувати структуру компанії ТОВ ТД "Прометей Агро", проаналізувати методи автентифікації, розглянути види QR-кодів, розробити безпечний механізм автентифікації з використанням QR-коду

4. ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Практична цінність полягає у створені безпечного механізму аутентифікації

з використання QR-коду, що передбачає можливість масштабування

5. ДАТА ВИДАЧІ ЗАВДАННЯ

Дата видачі завдання: 29.10.2021 року

Завдання видав _____ (підпис) Андрій ФЕСЕНКО (ініціали, прізвище)

Завдання прийняв до виконання _____ (підпис) Дмитро ЖЕБРАК (ініціали, прізвище)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Найменування етапів робіт	Строки виконання робіт (початок-кінець)	Відмітка про виконання
1	Уточнення постановки задачі	29.10.2021 – 27.01.2022	*виконано*
2	Аналіз літератури	28.01.2022 – 11.02.2022	*виконано*
3	Обгрунтування виробу рішення	12.02.2022 – 24.02.2022	*виконано*
4	Збір даних	25.02.2022 – 24.03.2022	*виконано*
5	Вибір алгоритму аутентифікації	25.03.2022 – 07.04.2022	*виконано*
6	Адаптація алгоритму для вирішення задачі	08.04.2022 – 20.04.2022	*виконано*
7	Проведення аналізу отриманих результатів	21.04.2022 – 05.05.2022	*виконано*
8	Робота над висновками	06.05.2022 – 20.05.2022	*виконано*
9	Оформлення презентації	21.05.2022 – 01.06.2022	*виконано*
10	Оформлення пояснювальної записки	02.06.2022 – 05.06.2022	*виконано*
11	Підготовка до захисту	06.06.2022 – 14.06.2022	*виконано*

Завдання видав _____ (підпис) Андрій ФЕСЕНКО (ініціали, прізвище)

Завдання прийняв до виконання _____ (підпис) Дмитро ЖЕБРАК (ініціали, прізвище)

Термін подання дипломної роботи до ЕК 06 червня 2022 року

РЕФЕРАТ

Пояснювальна записка: 62 с., 24 рис., 3 табл., 1 додатки, 35 джерел.

Об'єкт дослідження: процес автентифікації за допомогою QR-коду.

Мета роботи: дослідити можливість двохфакторної автентифікації за допомогою QR- коду та запропонувати безпечний механізм, який в майбутньому можна буде використовувати у всіх сферах діяльності.

Предмет дослідження: методи та алгоритми генерації та використання QR-коду.

Методи дослідження: структурний аналіз, порівняння, системний підхід, моделювання.

В роботі проведено аналіз видів автентифікації та вивчено саме поняття. Досліджено поняття QR-коду, його створення та методи використання. Проаналізовано можливість використання шифрування при аутентифікації.

Запропоновано створення QR-коду, з симетричним шифруванням, який містить в собі інформацію.

Побудовано логічну схему роботи безпечного механізму автентифікації з використанням QR-коду.

Розроблено безпечний механізм аутентифікації з використанням QR-коду.

Практичне значення роботи полягає у створенні механізму аутентифікації за допомогою QR-коду та можливості шифрування даних при їх використанні.

Результати здійснених у дипломній роботі досліджень можуть бути використані для впровадження в системи авторизації на будь-яких підприємствах, платформах тощо.

Напрямки подальших досліджень спрямовані на удосконалення даного механзму.

Ключові слова: АУТЕНТИФІКАЦІЯ, QR-КОД, ПАРОЛІ, ДВОХФАКТОРНА АУТЕНТИФІКАЦІЯ, ШТРИХ-КОД, КРИПТОГРАФІЧНЕ ШИФРУВАННЯ, АЛГОРИТМИ ШИФРУВАННЯ, БЕЗПЕЧНІ МЕХАНІЗМИ АУТЕНТИФІКАЦІЇ.

ЗМІСТ

ВСТУП

У сучасному інформаційному суспільстві рівень розвитку ІКТ зростає з кожним днем. В останні роки їх інтенсивне використання та глобальне поширення, а також необмежений доступ населення до Інтернету призвели до експоненціального зростання кількості інформації. У зв'язку з цим виникає потреба подати інформацію в компактному, легкому у використанні, зручному та візуально приємному для користувача вигляді. Це допоможе користувачам швидко й легко знайти одразу те, що вони шукають серед великого об'єму інформації, витрачаючи при цьому мінімум часу та зусиль.

Сучасне життя вже неможливо уявити без різноманітних пристроїв, таких як мобільні телефони чи планшети, які стали неодмінним атрибутом кожного. Тому потрібна нова форма інформації, щоб задовольнити масові запити сучасного інформаційного суспільства. Так виникає QR-код або «швидка відповідь» (що перекладається як швидка відповідь). Такими матричними кодами є 2D штрих-коди, які розшифровуються як у горизонтальному, так і у вертикальному розмірах, що дозволяють «кодувати» велику кількість інформації.

Це один з найпопулярніших інструментів мобільної комерції. Спочатку QR-коди використовувалися лише в промисловості, сьогодні вони активно використовуються в споживчих середовищах (реклама, інтернет-магазини), фінансах та економіці (банківські термінали), авіа- та залізничному транспорті (інформація про квитки), освітніх програмах, культурних (музеї)) та медицині.

Таким чином, QR-коди дозволяють будь-якому зацікавленому, наприклад, відразу отримати доступ до сайтів, присвячених компаніям, продуктам, історичним об'єктам і отримати вичерпні дані.

У зв'язку з цим можна сказати, що створення програмних додатків, які будуть шифрувати інформацію та генерувати QR-коди, сьогодні є актуальною темою. Наприклад, програма, розроблена в ході досліджень, дозволить звичайним користувачам сканувати зображення та автоматично потрапляти на веб-сторінки

замість того, щоб вручну писати довгі посилання в пошуковій стрічці на потрібні сайти, зменшуючи кількість дій користувачів, полегшуючи їх роботу та мінімізуючи витрати часу. Потрібні спеціальні знання або навички користувача.

Об'єкт дослідження: процес автентифікації за допомогою QR-коду.

Предмет дослідження: методи та алгоритми генерації та використання QR-коду..

Мета: дослідити можливість двохфакторної автентифікації за допомогою QR-коду та запропонувати безпечний механізм, який в майбутньому можна буде використовувати у всіх сферах діяльності.

Для реалізації мети були поставлені такі завдання:

1. Проаналізувати принцип роботи компанії ТОВ ТД ПРОМЕТЕЙ АГРО;
2. Провести огляд додатків для проходження двохфакторної автентифікації.
3. Дослідити процес генерування QR-коду;
4. Розробити безпечний механізм автентифікації за допомогою QR-коду .

РОЗДІЛ 1
ХАРАКТЕРИСТИКА ТОВ ТД ПРОМЕТЕЙ АГРО

1.1 Принципи роботи ТОВ ТД Прометей агро

ТОВ «ТД «Прометей агро» з моменту державної реєстрації є юридичною особою, має право управління та управління майном, яке розповсюджує, має печатку, печатку та власне найменування. ТОВ «Прометей Агро» має окремий баланс та власний банківський рахунок. Відповідно до чинного законодавства України відносини з підприємствами, установами та організаціями та громадянами в усіх сферах фінансово-господарської діяльності встановлюються на договірних засадах. Основним документом, що закріплює організацію, здійснює фінансово-господарську діяльність та оформляє ліквідацію підприємства, є статут.

Джерелом формування майна є статутний фонд, до складу якого входять основні засоби та оборотні кошти та інші цінності, що належать у праві власності ТОВ «ТД «Прометей Агро».

Генеральному директору підпорядковані фінансовий директор та маркетолог. Фінансовий директор здійснює підготовку фінансової звітності; аналіз проведених угод компанії; планування фінансової політики; управління фінансовими потоками. Маркетолог ТОВ «ТД «Прометей агро» формує та реалізує маркетингову стратегію; аналізує вплив тих чи інших процесів чи проектів; проводить рекламні кампанії від розробки до втілення.

Головний бухгалтер відповідає за стан та ведення бухгалтерського обліку на підприємстві.

Комерційний директор очолює відділ закупівель і збуту, в його безпосередньому підпорядкуванні перебуває менеджер по закупівлях, менеджер по роботі з покупцями та менеджер з логістики.

Менеджера відповідають за збут товару.

ТОВ «Прометей Агро» самостійно планує свою діяльність та визначає перспективи (стратегію) розвитку відповідно до попиту на продану продукцію та необхідності постійного збільшення власного прибутку та забезпечує логістику через систему прямих договорів (контрактів) з постачальниками.

Підприємство здійснює контроль за якісними показниками своєї фінансово-господарської діяльності за допомогою економічного аналізу.

ТОВ «Прометей Агро» веде оперативну бухгалтерську та щорічні статистику та фіксує визначні моменти своєї діяльності. Результати діяльності підприємства відображаються в річному балансі, звіті про фінансові результати та річному звіті.

При цьому товариство користується наступним слоганом: «З любов'ю до землі – з турботою про Вас!». ТОВ «ТД «Прометей агро» активно приймає участь у різноманітних виставках, має власний сайт, представлено в соціальних мережах.

На сьогоднішній день компанія користується всім доступною CRM-системою під назвою TRELLO.

Trello – це безкоштовна багато-платформна система для управління про різними проектами.

Вона використовує так звану парадигму для керування проектами – канбан. Віртуалізація цієї програми наступна: проекти зображені у вигляді дошок, що містять колони (списки.) Списки місять картки, в основному це картки клієнті. В картках є можливість ставити задачі. Картки можуть переходити з одного списку в інший, за допомогою перетягування. Карткам можна присвоювати відповідальних користувачів. Створення команд з користувачів та дошок. Система має підтримку тегів, у вигляді кольорових міток, які налаштовує сам користувач. Картки можуть містити коментарі, час, дату, переліки задач.

Інтерфейс працює за допомогою формату drag-and-drap, усі оновляються у фоні. Є деякі недоліки такі, як: система не працює в офлайні, відсутня можливість редагувати коментарі, немає безпечної аутентифікації кожен раз перед початок роботи.

В даній системі знаходяться всі дані про клієнтів всієї України, та кожен менеджер вносить інформацію про що він домовлявся та спілкувався з кожним із

них. Дана складова компанії є критичною, адже стоїть питання кадрів, та доброчестності, типу коли людина звільняється вона не має забирати з собою базу і це треба контролювати. Тому в даній дипломній роботі ми будемо розробляти безпечний механізм (модуль) аутентифікації за допомогою QR-коду. Кожен раз, як робітник повинен починати роботу, йому буде необхідно пройти аутентифікацію (ідентифікацію) за допомогою QR-коду, щоб отримати доступ до інформації. Це дасть можливість контролювати базу і роботу співробітника.

1.2 Аутентифікація користувача

Що стосується взаємодіючих сторін, аутентифікація означає, що одна зі сторін перевіряє, що взаємодіюча сторона є тим, за кого вона видає себе. Аутентифікацію сторін також зазвичай називають аутентифікацією.

Основним засобом ідентифікації є протокол ідентифікації, який дозволяє ідентифікувати кожну сторону, яка бере участь у взаємодії. Існують односторонні та взаємні угоди. Протокол — це розподілений алгоритм, який визначає порядок дій для кожної сторони. Під час виконання протоколу ідентифікації кожна сторона не передає жодної інформації про свій ключ, а натомість зберігає її та використовує для формування відповідних повідомлень для запитів, отриманих під час виконання протоколу.

1.2.1 Поняття аутентифікація

Перевірка достовірності інформаційного вмісту по суті є перевіркою його незмінності (з самого початку). Підтвердження достовірності даних означає дійсність того, що вихідні дані були вірні.

Насправді, кожне повідомлення, успішно розшифроване одержувачем, може бути створене лише відправником, оскільки тільки він знає їхній загальний секретний ключ.

Є випадки коли використання ключа є неможливим, тому для аутентифікації

джерела даних потрібен механізм цифрового підпису.

Автентифікація – це процес порівняння вхідних даних з тими, що є в пам’яті об’єкта. Наприклад, порівняння вхідного логіну і паролю, з тим, що є в базі. [2].

Методи автентифікації поділяються за типом ресурсу, структурою та нюансами організації мережі, віддаленістю об’єктів, а також техніками, які використовуються в процесі ідентифікації. Залежно від рівня конфіденційності існує кілька рівнів аутентифікації:

- Порушення або втрата даних призведе до значної шкоди – вимагає посилення аутентифікації;
- Доступ до конфіденційних систем даних передбачає використання методів взаємної аутентифікації та багатофакторної аутентифікації.

Усі методи аутентифікації можна відсортувати в порядку зростання.

Основна аутентифікація. При цьому типі аутентифікації логін та пароль користувача є частиною веб-запиту. Конфіденційну інформацію може легко ідентифікувати будь-який перехоплювач пакетів. Цей спосіб не рекомендується навіть у тих випадках, коли конфіденційна інформація не несе інформацію про користувачів або інтернет-ресурси. Це пов’язано з тим, що більшість людей використовують той самий пароль в Інтернеті для доступу до всіх служб, якими вони користуються. За даними Sophos, компанії, що займається інформаційною безпекою, 41% користувачів Інтернету використовують одну й ту саму інформацію для входу на різних платформах, будь то сторінка банку чи форум, присвячений їхньому улюбленому хобі.

Дайджест аутентифікація. Тип аутентифікації, який означає, що пароль користувача передається в хешованому стані. На перший погляд рівень захисту в цьому випадку здається дещо відрізняється від базового тесту. Насправді це не так: кожен пароль супроводжується довільним рядком (хешом), який генерується окремо для кожного нового веб-запиту. Постійне оновлення хешу не дозволяє зловмиснику розшифрувати пакет - кожне нове з'єднання створює інше значення шифру. Більшість інтернет-браузерів (Mozilla, Google Chrome, Opera) працюють на основі цього методу аутентифікації.

Використовуйте цифрові сертифікати для аутентифікації. Цей метод передбачає використання протоколу запиту та відповіді. Сторінка аутентифікації спрямовує певний набір символів («адресу») до користувача. Відповідь – це запит сервера, підписаний закритим ключем. Аутентифікація за відкритим ключем використовується як механізм безпеки в таких протоколах:

- SSL;
- Kerberos;
- RADIUS.

Аутентифікація за допомогою файлів cookie. Файл cookie – це невеликий набір даних, що надсилається Інтернет-сервером і зберігається на ПК користувача. Щоразу, коли браузер намагається підключитися до ресурсу, він надсилає файл cookie як частину HTTP-запиту.

Як засіб аутентифікації дані файлів куки використовуються системах безпеці наступного: чатів, форумів і різних онлайн-ігор. Файли cookie мають низький рівень захисту – їх неважко вкрасти, якщо сеанс не відфільтрований належним чином. Тому до IP-адреси, на яку входить користувач, будуть застосовані додаткові прив’язки.

Децентралізована аутентифікація.

За принципом децентралізованої автентифікації є декілька основних протоколів, які перелічені нижче:

• OpenID. Цей протокол дозволяє створити єдиний пароль для кількох Інтернет-ресурсів. Недоліком є вразливість до фішингових атак і атак типу «людина посередині». Зараз OpenID включає всім відомі компанії, такі як пошукові системи, платіжні системи тощо.

• OpenAuth. Застосовується до алгоритмів, подібних до OpenID. Є можливість використовувати служби AOL та будь-які інші сервіси, створені на їх основі. При цьому користувачам не потрібно додавати нові аккаунти на кожній платформі. Інформація про сеанс не зберігається в файлі cookie, сам файл cookie аутентифікації відокремлений певним доменом.

• OAuth. Дозволяє одному веб-ресурсу отримувати доступ до даних

користувача на іншому веб-ресурсі. Для служб Twitter і Apple.

1.2.2 Двофакторна аутентифікація

Одним із видів аутентифікації є двофакторна аутентифікація. Цей метод передбачає перевірку даних користувача двома факторами.

Якщо, наводити приклад перевірки, можемо говорити про сервіси від Google і Microsoft. Це працює наступним чином, людина повинна ввійти з нового пристрою, та необхідно ввести код: шестизначний або восьмизначний на додаток до імені для входу та пароля. Отримати його можна такими способів:

• надсилати текстові повідомлення на мобільні телефони;

• голосові дзвінки;

• одноразова реєстрація коду;

• аутентифікатор для мобільного або ПК.

Однією з переваг даної аутентифікації є доступність та безпека. Цей метод звісно має деякі недоліки. Наприклад, у людини проблеми з мережею, в такому випадку вони можуть стати перешкодою для отримання кодів підтвердження, а текстові повідомлення можуть бути перехоплені зловмисниками. Також є деяка затримка в отриманні текстових повідомлень – це пов’язано з процедурою аутентифікації.

Facebook, Web Money, Yandex, Microsoft, Google та інші використовують двофакторну аутентифікацію. Усі вони використовують власні процедури аутентифікації, кожна з яких пов’язана певними стандартами.

Висновки до першого розділу

Тому першочерговим напрямком розвитку поняття двофакторної аутентифікації залишається ризик несанкціонованого доступу до даних. Автентифікацію за двома факторами є можливість використовувати для авторизації в операційні системи Windows і Linux, хмарні служби та корпоративні середовища.

Двофакторна аутентифікація усуває більшість загроз, а вибір розширеної аутентифікації сам по собі виправданий і є зручним спосіб.

РОЗДІЛ 2
ПОНЯТТЯ QR ТА ДВОФАКТОРНОЇ АУТЕНТИФІКАЦІЇ

2.1 Методи двофакторної аутентифікації (в цілому)

У сучасному світі використовується також більше 5 мільярдів мобільних пристроїв, і застосування телефону в якості засобу аутентифікації допомагає швидко вирішити завдання з посиленого захисту, скорочення додаткових витрат і затримок доставки. Проблема витоку інформації актуальна в усьому світі і застосування двофакторної аутентифікації для захисту інформації послужить додатковим бар'єром для зловмисників. Методи двофакторної аутентифікації розглядаються як механізми посилення стійкості аутентифікаторів.

Двофакторний захист досить надійний бар'єр, серйозно ускладнює доступ до чужих даних і в якійсь мірі нівелює недоліки класичної парольного захисту [15].

У дисертації до зв'язків класичного використання логін / пароль застосовується додатковий бар'єр захисту, іменований другим фактором, володіння яким необхідно підтвердити, з метою прийняття прав доступу до облікових та інших даних [16].

Автентифікація за двома факторами використовується для здійснення доступу до інформаційних систем, акаунтів в соціальних мережах, до пошти та інших сервісів.

2.1.1 Одноразові паролі

Розглянемо найбільш поширені типи одноразоваъ аутентифікації користувачів інформаційних систем і відзначимо їх переваги і недоліки [17]:

1. SMS-код. Після успішної авторизації Користувача, на його номер телефону надходить SMS з кодом, який діє певний проміжок часу і вводиться в аккаунт системи. Повторний вхід можливий, при відправці нового SMS з кодом.

Перевагою такого методу аутентифікації, є доступність, так як йде прив'язка до телефонного номера користувача, і сесія для нової генерації пароля повторюється.

Недоліком є відсутність стільникового зв'язку, а також підміна номера.

Відмова від використання SMS для обробки другого фактора полягає в небезпеці даного методу. Стільникові мережі використовують «SignalSystem 7» для взаємодії між собою. Але в цій системі виявлені серйозні уразливості, що дозволяють перехоплювати вхідні дзвінки та SMS абонентів [18].

2. Перевірка входу за допомогою мобільних додатків. Авторизація здійснюється на смартфоні з встановленим спеціальним додатком. Смартфон зберігає ключ і забезпечує вхід в інформаційну систему.

Переваги: немає необхідності вводити пароль, не потрібні стільниковий зв'язок для отримання SMS повідомлень та інтернет.

Недоліки: якщо пройде перехоплення приватного ключа, ймовірна фальсифікація особистого номера.

3. Апаратні (фізичні) токени. Вважається найбільш міцним і надійним методом двофакторної аутентифікації-USB ключ. Він має свій процесор, який виробляє генерацію одноразового пароля для ідентифікації людини при приєднанні до комп'ютера. Підбір ключа залежить від певної послуги. Перевага: цілком незалежний прилад не вимагає смартфона. Недоліки: прилад береться окремо; не всі пристрої мають можливість використовувати цей спосіб; на один аккаунт, один токен.

4. Додатки – аутентификаторы. Пароль на пристрої генерується за допомогою спеціальної програми, розробленої. Під час опції користувач отримує головний ключ, який використовує криптографічний алгоритм для створення одноразового пароля на певний період часу, зазвичай до 1 хвилини. Плюси: для відкриття сеансу потрібен Інтернет. Недолік: можливе перехоплення первинного ключа, а потім порушник може згенерувати наступні паролі.

5. Біометрична аутентифікація. Аутентифікація за допомогою унікальних біометричних даних користувача, таких як відбитки пальців, структури сітківки,

риси обличчя, голос тощо. Під час реєстрації в біометричному аутентифікаторі використовується спеціальний зчитувач для зчитування та запису зразків відповідних біометричних даних користувача. Потім програмний алгоритм обробляє отримані зразки, а система зберігає їх у базі даних як шаблони даних.

2.1.2 Друковані коди

Двофакторна аутентифікація дозволяє забезпечувати більш високий ступінь захисту в порівнянні з однофакторною аутентифікацією, при якій користувач пропонує тільки один фактор, зазвичай пароль, який використовується для контролю доступу до чутливих систем і даних.

Пропонована система двофакторної аутентифікації складається з двох етапів. На першому етапі аутентифікації користувач вводить свої дані для авторизації. При успішному проходженні цього етапу необхідно пройти другий етап, на якому розглядається генерація одноразового пароля на основі програми аутентифікації і смартфона [39].

Розглянемо основні протоколи, які використовуються в дисертаційному дослідженні.

OAuth-сервер від DnC (OAuthSD) – це сервер аутентифікації, який реалізує протокол OAuth і систему OpenID Connect [40].

OpenID-відкрита децентралізована система, що дозволяє використовувати єдиний аккаунт Користувача для аутентифікації web додатків не пов'язаних один з одним. Завдяки централізації аутентифікації додатків і користувачів сервер OpenID Connect дозволяє повністю контролювати доступ до конфіденційної інформації [40, p.2].

OAuth-протокол авторизації, що надає конкретному Сервісу або програмі повноваження на допуск до користувацького ресурсу на іншому сервісі або програмі. Протокол OAuth надає змогу без вказівки даних для авторизації використовувати сторонню програму, а також дозволяє видавати набір прав при налаштуванні. Він базується на застосуванні веб – технологій, запитів і редиректив

HTTP, це дає можливість використання на різних платформах, якщо є глобальна мережа Інтернет і браузер. Протокол OAuth має загальну структуру роботи:

- доступ для авторизації;

- застосування до ресурсів захисту.

Підсумком авторизації вважається "токен доступу" - конкретний ключ, що дозволяє надати доступ до захищених ресурсів. Протокол HTTPS дає можливість звернутися до них, вказавши його в заголовках або в якості одного з параметрів отриманого токена доступу.

2.1.3 Додатки для проходження двохфакторної аутентифікації

Зараз майже всі найважливіші ресурси та сервіси використовують 2FA. На першому етапі аутентифікації користувач вводить свої дані, якщо ви успішно пройшли цей етап, необхідно пройти другий етап. Другим кроком є автентифікація OTP за допомогою SMS або електронної пошти, або за допомогою програмного генератора паролів, встановленого на вашому мобільному пристрої.

У зв'язку з цим розглянемо кілька найбільш популярних аутентифікаторів і їх можливості. Незважаючи на те, що базова функція у всіх додатків одна і та ж – створення одноразових кодів по одному і тому ж алгоритму, деякі аутентифікатори володіють додатковими функціями або особливостями інтерфейсу.

1. Google Authenticator. Підтримувані платформи: Android, iOS . Google Authenticator є додатком, створеним для авторизації на основі другого фактора. Google Authenticator не має налаштувань, що полегшує роботу цього додатка [15].

2. Duo Mobile. Підтримувані платформи: Android, iOS. Duo Mobile має ряд переваг: таких як простота у використанні, мінімалістичний і не вимагає додаткових налаштувань [16]. Для відображення коду необхідний токен.

3. Microsoft Authenticator. Підтримувані платформи: Android, iOS. Цей додаток має можливість виробляти настройки токена, так що при запуску токен може бути прихований. Microsoft Authenticator-простий у використанні і є високофункціональним [17].

4. FreeOTP. Підтримувані платформи: Android, iOS. Програмне забезпечення з відкритим кодом [18]. Розмір програми дорівнює 750 Кбайт для платформи iOS, є мінімальним з усіх. Додаток Google Authenticator має розмір рівний 14 Мбайт, Authy дорівнює 44 Мбайта. У додатку FreeOTP є можливість проводити настройку токена в ручну, що У інших відсутня.

5. Authy. Працює на платформах Android, iOS, Windows, macOS, Chrome [69]. Ця програма дозволяє використовувати хмарні сервіси для зберігання токенів, що є перевагою. Хмарні сервіси дозволяють з будь-яких пристроїв надати доступ до токену. Обов'язковим є те, що додаток використовує аккаунт, прив'язаний до телефонного номера і без цього робота неможлива.

6. Яндекс.Ключ. Підтримувані платформи: Android, iOS. «Яндекс. Ключ» також як і додаток Google Authenticatorне має необхідності проводити реєстрацію при вході [10]. Додаток має можливості такі як використання хмари Яндекс для зберігання резервних токенов.

2.2 Використання QR коду для двухфактороні аутентифікації

Звичайно, ви можете користуватися відбитками пальців або FaceID, USB-токени, одноразові паролі, теги NFC і все, що вам потрібно, щоб змінити частини сервера або змінити обладнання клієнта. Зчитуючи QR-код безпосередньо в поле введення сервера, я використовую той самий логін і пароль. Камери та сучасні браузери на більшості сучасних ноутбуків, планшетів, ПК, які використовуються для цього.

2.2.1 Історія створення QR

QR Code — код швидкої відповіді.

Японія вже давно відома різноманітними цікавими рішеннями, які необхідні щоб облегшити бізнес-процеси та завдання максимально компактними та ефективними. На початку 1990-х років перед інженерами Denso, великої інженерної

фірми, постало непросте завдання: створити єдиний штрих-код (або щось подібне) для маркування деталей і компонентів для сканування. У той час компанія прийняла понад 10 сховищ різного призначення, і працівники заводу скаржилися, що використання сховищ вимагає інтенсивної концентрації, а самі сховища містять мало корисної інформації. Проте не можна назвати це питання приватним: компанії по всій країні розробили власні версії штрих-кодів і намагалися зберігати чим більший об’єм інформації [3]. Нові коди мали закривати наступні позиції:

- об’єм даних, які мають міститься в коді, повинен значно збільшуватися;
- процес читання повинен бути максимально ретельним і швидким;
- коди мали бути стійкі до масляних плям, бруду та інших пошкоджень;
- зчитувач мав бути досить простим, та не дорогим.

Згідно з легендою, натхнення прийшло до Масахіро під час обіду за грою в шахи. Що ж, дуже схоже на правду: QR-код дійсно виглядає як ігрове поле з камінчиками на ньому, а ігрова ситуація — це ті ж закодовані дані.

Від тоді інформацію кодували не тільки по горизонталі, але і по вертикалі. Їх об’єм зріс до 7000 символів, включаючи не тільки латиницю, а й ієрогліфи. Сам код мав особливість прочитатись під будь-яким кутом завдяки квадратам у трьох кутах коду, які діють як детектори положення.

За словами Масахіро « "квадрати", які створюють код, теж обрані не випадково: виявилося, що квадратні візерунки майже не зустрічаються в ділових документах і взагалі на етикетках. Таким чином, можливість помилки читання, викликаної некоректними даними, зводиться до нуля. З метою додаткової впевненості Масахіро запропонував використовувати певну величину відхилень між інформативною частиною патерну і його межами.

Через те, що код був максимально легким, він сподобалися людям за межами Японії. У середині 2000-х про японський винахід почув весь світ. Саме позначення QR-код є офіційно товарним знаком, який належить компанії-винахіднику.

Про версії QR-кодів можна говорити досить довго: є як «маленькі» версії (21x21px), так і більші — 177x177px. Ось основні кодування даних, прийняті в усьому світі:

- цифрове кодування
- буквено-цифрове кодування
- байтове кодування
- кодування "кандзі".

У 2000-х роках QR-коди стали одним з методів оплати, проте то було експериментом, але спочатку на цю процедуру витрачалося багато часу — сканеру знадобилося 20 секунд, щоб зчитувати інформацію. У 2003 році китайська компанія Inspiry створила набагато швидкий зчитувач QR-кодів. Пізніше він також випустив перший портативний сканер, і технологія стала дуже популярною.

Вибух технологій стався під час масової появи мобільної техніки. Виробники гаджетів навчили камери розпізнавати QR-коди, і ця технологія почала поширюватися по всьому світу.

WeChat став потужним інструментом маркетингу та монетизації для бізнесу: кожен бренд у соціальній мережі має свою загальнодоступну сторінку та окремий QR-код. Мешканці швидко звикли до технології в середовищі Messenger – QR-коди тепер використовуються по всій країні.

Технологія також активно розвивалася західними соціальними мережами. Наприклад, Instagram запустив сервіс Nametag-card, схожий на QR-код, що працює за тим же принципом. Власник будь-якого профілю міг створити таку карту для швидкого переходу на сторінку.

2.2.2 Різновиди QR

Примітка. Наведені нижче QR-коди призначені тільки для демонстраційних цілей.

Ігри з кольорами. Сучасні смартфони та зчитувачі набагато менше залежать від чіткості зображення коду, ніж їхні попередники. Це дозволяє фарбувати QR-коди в різні кольори та інтегрувати їх у дизайн продуктів (як показано на рисунку 2.1).

Рисунок 2.1. Кольоровий QR

Ігри та форми. «Кругові» QR-коди, коди із зображеннями та логотипами миттєво – теж не новина. Методи гіперкодування використовуються, щоб дозволити пристроям формально аналізувати нестандартні коди (як зображено на рисунку 2.2).

Рисунок. 2.2. «Круглий» QR

QR-коди бувають у різних «дизайнерських» версіях: це не просто квадрат із зображенням у полі коду, це витвір мистецтва (як показано на рисунку 2.3, 2.4).

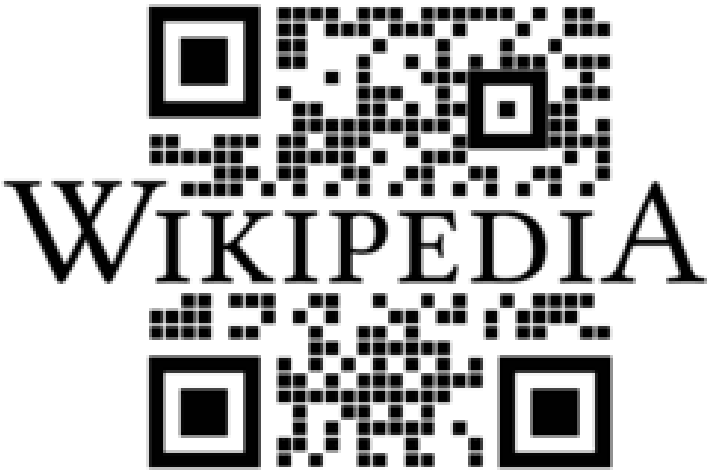

Рисунок. 2.3. «Дизайнерський» QR

Рисунок. 2.4. Нестандартно розташовані QR

Крім того, ви можете регулярно знаходити анімовані QR-коди, але це більше диво моди, ніж справедливе використання технологій.

Непряме використання QR-кодів. Листівки, футболки, прикраси з пам’ятними кодами — тут навіть є QR-коди (як показано на рисунку 2.5).

У публічних виставках і вуличних бібліотеках використовуються QR-коди: вони часто шифрують основну інформацію про художників, письменників і музикантів, а за короткими посиланнями можна завантажити копії робіт.

Роздрібна торгівля не є винятком: QR-коди шифрують коди купонів і номери карток програми лояльності. Однак у QR-кодів є сильні конкуренти, наприклад, рішення NFC (як показано на рисунку 2.6).

Рисунок. 2.5. Підвіска у вигляді QR-коду

Рисунок 2.6. Картка-меню в QR-коді

QR-коди все ще користуються популярністю серед рекламодавців: QR-коди шифрують адреси веб-сайтів і посилання для завантаження, намагаючись перевести користувачів з офлайну в режим онлайн.

2.2.3 Порівняння QR з штрих-кодом

QR-коди є прямими наступниками штрих-кодів. Але другий заснований на технології азбуки Морзе і використовується для автоматизації різних товарів і обладнання. Протягом десятиліть штрих-коди були єдиним звичайним варіантом маркування. Звичні смуги і цифри з який складався штрих-код приїлися користувачам та необхідно було щось нове. Однак функціональність штрих-кодів обмежена.

Основна відмінність між QR-кодом і традиційним штрих-кодом полягає в тому, що сканер розпізнає його як 2D-зображення. Щоб нормалізувати зображення під час читання і зменшити ймовірність помилок, код мав містити кілька квадратів в одному з кутів, а також багато менших точок синхронізації, розкиданих по всій області коду. Це створило ціле поле експериментів, яке не закінчилося й донині.

Штрих-коди можуть містити від 20 до 30 символів, що іноді замало. Фахівці собі мали мету розширити функціональність штрих-коду, але класичними методами це не вдалося. Двовимірні (матричні) коди виходять на арену, де QR-коди по праву є робочою конячкою.

Традиційні штрих-коди, які ми часто скануємо в супермаркетах, щоб дізнатися ціни на продукти, обмежені 20 горизонтальними буквено-цифровими символами. Вони засновані на стандарті Універсального коду продукції США (UPC), створеному в 1973 році.

Недолік: якщо штрих-код пошкоджений, інформація недоступна. А я не можу шифрувати ієрогліфи.

При порівнянні з звичайними штрих-кодами, QR-коди мають багато позитивних якостей:

- Подвійний обсяг закодованої інформації;
- інформація не копіюється зрозумілими людиною символами;
- Є кілька варіантів на вибір.

2.2.4 Процес генерування QR

Щоб згенерувати QR-коди, потрібно зрозуміти, як вони працюють. Принцип використання QR-коду полягає в тому, що об'єкт друкується або малюється кодом, який потім можна прочитати та розшифрувати за допомогою пристрою з робочою камерою та встановленим програмним забезпеченням для розшифрування QR-коду. (як зображено на рисунку 2.7).

Рисунок. 2.7. Приклад QR-коду

Інформація, що міститься в QR-коді, має дві орієнтації: горизонтальну і вертикальну. Завдяки такому розміщенню даних у QR-коді він має можливість мати в собі більше інформації, ніж його попередник, а саме: цифри, літери, значки, символи тощо. Можна включити в QR-код, наведений у таблиці 2.1:

Таблиця 2.1

Типи даних і максимальний можливий обсяг в QR-кодах

Тип даних	Максимальний обсяг (символ)	Можливі символи
Числові дані	7089	9,8,7,6,5,4,3,2,1,0
Символьні дані	4296	A–Z, $, %, *, +, -, ., /, space,:
Бінарна інформація, байт	2953	JIS X 0201
Ієрогліфи	1817	JIS-X/0208

Ще одним великим плюсом QR-коду є його можливість відновлювати дані, тобто якщо вони пошкоджені, він все рівно зчитається у повному обсязі. Система корекції помилок Ріда-Соломона робить це можливим.

Максимальна кількість кодових слів, які можна відновити, становить 30%. Відповідно до специфікації, QR-код має 4 рівня виправлення помилок: L-7%, M-15%, Q-25%, H-30%. Це працює за таким принципом, в QR-код зашифрується набагато меньше даних, якщо ступінь виправлення помилок буде висока.

Сам QR-код містить мітки і пікселі, які є закодованою інформацією, що

зберігається в QR-коді.

Будь-який QR-код повинен містити такі типи етикеток (як показано на малюнку 8):

1. Позиціонування.

2. Номер версії.

3. Синхронізація.

4. Формат.

5. Вирівняти.

6. Рівень виправлення помилок. Дозволяє визначити рівень захисту від шуму, який використовується на етапі кодування, щоб вибрати правильний метод виявлення можливих помилок у вашому коді (як показано на рисунку 2.8).

Рисунок 2.8 Мітки та дані на QR-коді

Усі існуючі програми для зчитування та декодування QR-кодів реалізують простий алгоритм виявлення QR-кодів на зображеннях, отриманих з камер. Потім виконується стандартна процедура декодування інформації з QR-коду. Однак цей алгоритм ідентифікації вимагає дуже точного позиціонування конкретної області на пристрої, який потрібно ідентифікувати, а також певного позиціонування QR-коду в просторі. Коли вибрана область на пристрої точно збігається з QR-кодом, знайдіть

три теги локатора, згадані вище.

Ці мітки розташовані строго в певних позиціях у виділеній області зображення. Недоліком цього методу є те, що QR-код можна розташувати в будь-якій області зображення, користувач повинен спочатку зосередитися на бажаному QR-коді і переконатися, що область, де фіксується QR-код, збігається з самим QR-кодом, і треба розшифрувати.

Процедура двофакторної аутентифікації дуже проста. Це те, що потрібно зробити користувачеві – встановити додаток на смартфон, отримати доступ до налаштувань безпеки сервісу, забезпечити використання такої програми у варіанті двофакторної аутентифікації, потім вибрати відповідну опцію, відсканувати QR-код, що відображається на екрані Сервіси, які використовують програми двофакторної аутентифікації.

Після цього програма періодично запускається (наприклад, кожні 30 секунд) для створення нового одноразового коду. Паролі генеруються на основі ключа, відомого тільки вам і серверу, плюс поточний час, округлений до 30 секунд. Оскільки обидва компоненти однакові як для клієнта, так і для сервісу, коди генеруються синхронно. Цей алгоритм називається OATH TOTP (одноразовий пароль на основі часу) і використовується в більшості випадків.

QR-код-це монохромне зображення, на якому деякі пристрої (наприклад, смартфон зі спеціальним додатком) розпізнають текст. Цей текст може бути не просто простою пропозицією, але, хоча і не включеним в офіційну специфікацію, посиланням, номером телефону або візиткою. Такі коди часто використовуються для кодування посилання і її друку на плакаті або Візитці.

Процес генерації QR-коду розділений на кілька прозорих кроків:

1. Кодування даних;
2. Додавання та заповнення інформації про послугу;
3. Поділ інформації на блоки;
4. Генерація байтів корекції;
5. З'єднувальні блоки;
6. Розміщення інформації на QR-коді.

Кодування даних. QR-код підтримує певну кількість способів шифрування даних в залежності від використовуваних символів: Числові, буквено-цифрові кандзі (китайсько-японські символи) і побайтовое кодування. Цифрове кодування логічно, що включає лише використання цифр від 0 до 9, буквено-цифрова — великих літер латинського алфавіту, цифр і символів $%*+-. / : І простір. Для початку потрібно створити порожню бітову послідовність, яка в подальшому буде заповнюватися.

Цифрове кодування. Цей тип кодування вимагає 10 біт по 3 символи. Послідовність інформації ділиться на групи по 3 біти, і кожна група перекладається в 10-розрядне двійкове число і додається до бітової послідовності. Якщо загальна кількість символів не кратна 3, то отримане двозначне число кодується 7 бітами, якщо в кінці залишилося 2 символи, і 4 бітами, якщо це 1 символ. Наприклад, є рядок «12345678», який потрібно закодувати. Ми розділили його на числа: 123, 456 і 78, потім перетворили їх у двійкову форму: 0001111011, 0111001000 і 1001110 відповідно, і об’єднали їх у потік: 000111101101010.

Таблиця 2.2

Значення символів в буквено-цифровому кодуванні.

Значення	Символ	Значення	Символ	Значення	Символ	Значення	Символ
0	0	12	C	24	O	36	Пробіл
1	1	13	D	25	P	37	$
2	2	14	E	26	Q	38	%
3	3	15	F	27	R	39	*
4	4	16	G	28	S	40	+
5	5	17	H	29	T	41	-
6	6	18	I	30	U	42	.
7	7	19	J	31	V	43	/
8	8	20	K	32	W	44	:
9	9	21	L	33	X		
10	A	22	M	34	Y		
11	B	23	N	35	Z		

Алфавітно-цифрове кодування. У цьому випадку для 2 символів потрібно 11 біт інформації. Вхідний потік символів розбивається на 2 групи, кожен символ у групі кодується відповідно до наступної таблиці, значення першого символу в групі множиться на 45 і додається значення другого символу. Отримане число перетворюється в 11-розрядне двійкове число і додається до бітової послідовності.

Якщо в останній групі є 1 символ, його значення негайно кодується як 6-значне число і додається до бітової послідовності.

Байтове кодування. Це загальний метод кодування, яким можна закодувати будь-який символ. Недоліком методу є відносно мала щільність інформації. При цьому текст кодується в будь-якому кодуванні (рекомендується в UTF-8) і результуючий набір електронних даних приймається без змін.

Додайте інформацію про послугу. На цьому етапі потрібно обрати рівень корекції: чим вищий рівень, тим вища можливість пошкодження зображення і тим менше інформації при цьому ж розмірі. Є 4 рівня модифікаторів: L (макс. 7% пошкодження), M (15%), Q (25%) і H (30%). Найбільш часто використовуваним рівнем є M.

Ще однією особливістю QR-коду є його версія (чим більше, тим більший розмір). Є всього 40 версій. Номер версії залежить від обсягу інформації та ступеня виправлення.

Додавання службових полів. До цього моменту необхідно визначити рівень корекції та версію. Метод кодування - це 4-бітове поле з наступними значеннями: 0001 для числового кодування, 0010 для буквено-цифрового і 0100 для байтового кодування. Обсяг даних-це кількість закодованих символів, а побайтно - кількість байтів, а не бітів в прийнятій послідовності), зображено у вигляді двійкового числа заданої довжини (визначається по таблиці 2.2)

Таблиця 2.3

Довжина поля кількості даних.

	Версія 1-9	Версія 10-26	Версія 27-40
Цифрове	10 біт	12 біт	14 біт
Буквенно-цифрове	9 біт	11 біт	13 біт
Побайтове	8 біт	16 біт	16 біт

Наприклад, цей рядок має довжину 100 байт, кодується побайтно, рівень зміщення дорівнює M. Довжина бітової послідовності цього рядка становить 800

біт. Використовуючи таблицю 2, ви можете визначити, яка шоста версія є кращою. Довжина поля вказує на кількість інформації в нашому випадку-8 біт (Таблиця 2.3)

Поле методу кодування-0100, поле кількості даних-01100100 (100 в двійковому форматі). В результаті виходить бітова послідовність 010001100100-вихідна послідовність.

Якщо довжина отриманого бітового рядка виявиться більше допустимої для обраної версії, збільште версію на одиницю і знову додайте службові поля.

Специфікація дозволяє використовувати ЗМІШАНЕ кодування. Це означає, що кілька груп даних можуть бути закодовані по-різному і об'єднані в послідовність. Робиться це наступним чином: і так далі. Заповнення. На даному етапі є рядок бітів даних, кількість бітів якої свідомо не кратно 8. Нам потрібно заповнити її нулями, щоб її довжина стала кратна 8. Тепер нашу рядок бітів можна розбиті на групи по 8 біт і представлені у вигляді послідовності байтів. Якщо кількість бітів в поточній послідовності байтів менше, ніж потрібно для обраної версії, її необхідно заповнити.

Приклад. Існує послідовність: послідовність бітів довжиною, кратною 8 10101011101; доповніть її нулями, щоб зробити її кратною 8: послідовність бітів довжиною, кратною 8 10101011101 00000; тепер припустимо її довжину становить 104 біти. Для вибраної версії вам потрібно 128 біт, потім до заповнення потрібно додати 24 біти «заповнення» (3 байти): послідовність бітів довжиною, кратною 8 10101011101 00000 11101100 00010001 11101100.

Поділ інформації на блоки. Набір електронних даних (далі-дані), отриманий на попередньому етапі, розбивається на кількість блоків, пов'язаних з версією і рівнем виправлення. Якщо кількість блоків дорівнює одному, то цей крок можна пропустити.

Визначення кількості байтів в кожному блоці. Для цього необхідно розділити загальну кількість байтів (кількість байтів можна знайти в даних або розділити число з таблиці 2 на вісім) на кількість блоків даних. Якщо це число не є цілим числом, ви повинні визначити залишок від ділення. Цей залишок визначає, скільки всього блоків буде заповнено (тих блоків, в яких кількість байтів на один більше,

ніж в інших). Всупереч тому, що очікувалося, доповнені блоки повинні бути не першими блоками, а останніми.

Заповнення блоків. Блок повністю заповнений байтами даних. Коли поточний блок заповнений, черга переходить до наступного. Байтів даних повинно вистачити рівно на всі блоки, не більше і не менше.

Створення коригувальних байтів. До кожного блоку даних застосовується наступний алгоритм. Цей алгоритм створений на алгоритмі Ріда-Соломона. Перше, що необхідно зробити, це визначити, скільки байтів патча потрібно створити. Так званий генераторний поліном визначається кількістю коригуючих байтів. Він називається многочленом, тому що вихідний метод використовує многочлен з тими ж коефіцієнтами.

Перед написанням циклу необхідно обрати масив, довжина якого дорівнює максимальній кількості байтів у поточному блоці та кількості байтів корекції, і заповнити його початок, кінець – нулями байтами в поточному блоці.

Цикли в даному списку, повторюються стільки разів, скільки байтів інформації міститься в поточному блоці.

1. Обираємо перший елемент масиву, зберігаємо його значення в змінній a і видаляємо з масиву.

2. Якщо A дорівнює нулю, пропускаємо наступні кроки (до кінця списку) і перейдіть до наступної ітерації циклу.

3. Знайдіть у таблиці 8 число, що відповідає числу a, і підставте його у змінну.

4. Тоді для перших n елементів, де N-кількість байтів корекції, i-лічильник циклів:

- До i-го значення многочлена, який генерується, треба додати значення б і записати цю суму в змінну в (сам многочлен не змінювати).
- Якщо більше значення ніж 254, треба використовувати її залишок при розподілі на 255.
- Знаходимо значення в таблиці 7 і виконайте побітове додавання за модулем 2 (XOR, оператор ^ у багатьох мовах програмування) з i-им значенням підготовленого масиву та запишіть отримане значення в i-е значення A

підготовленого масиву комірок.

Перші n байт масиву, підготовленого після цього циклу, є коригуючими байтами. Для кожного блоку даних буде отримано відповідний блок коригуючих байтів.

З'єднувальні блоки. Є багато блоків і така ж кількість коригуючих блоків байтів, які повинні бути об'єднані в один потік. Цей процес виконується наступним чином: з кожного блоку даних береться по одному інформаційному байту за раз, тоді коли черга доходить до кінцевого блоку, з нього береться один байт і черга рухається на початок. Це триває до тих пір, поки байти в кожному блоці не будуть вичерпані. Якщо в поточному блоці більше немає байтів, він буде проігнорований (це відбувається, коли звичайні блоки вже порожні, а в повних залишився ще один байт). Ви повинні зробити те ж саме з блоками корекції байтів. Вони завантажуються в тому ж порядку, що і відповідні блоки даних.

2.2.5 Перспективи аутентифікації за допомогою QR

QR-коду є зручним та легким методом аутентифікації, тому що:

- Є багато додатків, які зберігають пароль в системи. Це нормально, але ці програми показують пароль у в відкритому вигляді, і його потрібно вводити вручну. Ви можете відобразити збережений пароль перетворивши його у QR-код.

- Зручно, що є можливість мати QR-код на папері, пластику і прикріпити у вигляді брелка.

- QR-код може бути згенерований при реєстрації де-небудь і збережений в якості резервного методу входу в систему.

- можна згенерувати досить складний одноразовий пароль і згодувати його в QR.

2.3 Використання криптографічного шифрування в QR

Як варіант є застосування ЦВЗ до QR-коду шляхом зміни яскравості пікселів

матриці QR-коду. При цьому, з метою підвищення стійкості повідомлення ЦВЗ до порушень, пропонується виправити значення всіх трьох компонентів зображення QR-коду: R – червоного, G – зеленого, B – синього. Далі потрібно визначити, які пікселі будуть піддаватися корекції, і за яким правилом буде виконуватися ця корекція.

При генерації QR-коду кожному біту інформації відповідає лише один білий або чорний піксель на зображенні, але під час друку зображення QR-код масштабується так, щоб QR-код можна було прочитати камерою телефону. Для того, щоб мати можливість застосувати ЦВЗ, пропонується зіставити 9 пікселів одного кольору з кожним пікселем зображення перед друком (як показано на рисунку 2.9).

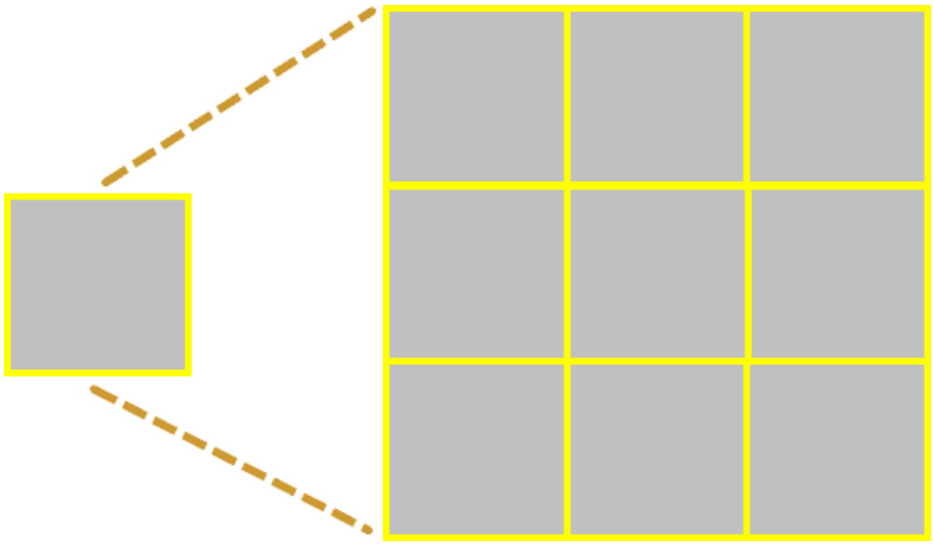

Рисунок. 2.9 Формування квадрату QR-коду для одного пікселя

Після такого методу зміни QR-коду ми застосовуємо ЦВЗ, змінюючи середній піксель кожного квадрата 3 × 3.

Примітка 1. Заміна середнього пікселя при зчитуванні ЦВЗ має певні переваги. По-перше, невелике налаштування яскравості лише одного з дев’яти пікселів не завадить зчитувати повідомлення QR-коду безпосередньо стандартними засобами. По-друге, наявність корекції середньої яскравості пікселя навіть після грубої експозиції можна визначити, аналізуючи вихідні значення сусідніх пікселів,

оскільки вони спочатку мають однаковий колір.

Примітка 2. Це правило визначення пікселів, які підлягають корекції за своїми значеннями, на перший погляд дає пропускну здатність алгоритму 1/9 біт на піксель.

Отже, було отримане правило, за яким визначаються пікселі, що підлягають корекції своїх значень.

2.3.1 Шифрування та його види

Якщо поєднати деякі математичні алгоритми з різними ключами, а саме криптографічними, виникне процес, який називається шифрування. Розглянемо два типи шифрування, які є основними – симетричне та асиметричне, а також 5 найбільш часто використовуваних алгоритмів шифрування.

Симетричне шифрування

Користуючись цим методом, використовується єдиний криптографічний ключ для двох процесів: шифрування та дешифрування. Так, як один ключ використовується для двох операцій – це облегшує роботу методу. Розберемося з процесом симетричного шифрування на простому прикладі:

Двоє близьких друзів Антон і Аліса живуть у Києві. Алісі необхідно поїхати у інше місто. В них є лише один спосіб спілкуватися друг з другом – пошта. Але є одна проблема: вони піклуються про те, що хтось зможе прочитати їхні листи.

Щоб захистити свої літери, вони вирішили зашифрувати повідомлення, щоб кожна літера була замінена буквою на сім позицій внизу алфавіту. Замість того, щоб писати «Apple», вони напишуть «hwwsl» (A -> H, P -> W, L -> S, E -> L). Щоб розшифрувати повідомлення, ви повинні замінити кожну літеру сімома позиціями в алфавітному порядку назад. Така техніка має назву «шифр Цезаря», адже нею користувався полководець Юлій Цезар.

Найбільшою перевагою симетричного шифрування є простота процесу, оскільки один ключ використовується як для шифрування, так і для дешифрування. Якщо необхідно зашифрувати великий фрагмент даних, симетричне шифрування — чудовий варіант. В результаті симетричні алгоритми шифрування:

- Набагато швидше, ніж їхні аналоги з асиметричним шифруванням (про яке ми поговоримо незабаром);
- Вимагають меншої обчислювальної потужності;
- Не знижує швидкість Інтернету.
- алгоритми симетричного типу шифрування

«Шифр Цезаря» створений на спеціальній логіці шифрування даних, яка вирішує проблему, що ви можете легко розшифрувати інформацію. Сучасні методи шифрування засновані на дуже складних математичних функціях, які майже неможливо зламати.

Існують сотні симетричних алгоритмів! Найпоширенішими з них є AES, RC4, DES, 3DES, RC5, RC6 тощо. Розглянемо три найпопулярніші.

DES-алгоритм симетричного шифрування

DES (стандарт шифрування даних), являється найстарішим методом симетричного шифрування.

DES перетворює 64-розрядні блоки даних доступного тексту в зашифрований текст, роділяючи їх на два окремих 32-розрядних блоки, використовуючи процес шифрування до кожного окремо. Блоки містять у собі 16 циклів різних процесів, таких як розширення, перестановка, тощо, через які дані будуть проходити в зашифрованому вигляді.

2.3.2 Алгоритми типів шифрування

3DES, також відомий як TDEA, є алгоритмом потрійного шифрування.

Усі алгоритми шифрування в кінцевому підсумку підпорядковані силі часу, і 3DES не є винятком. Уразливість в алгоритмі Sweet32 3DES виявили Cartikean Bhavargan і Gaetan Leurent. Цей висновок спонукав індустрію безпеки розглянути алгоритми старіння, про які Національний інститут стандартів і технологій (NIST) офіційно оголосив у проекті управління 2019 року.

Згідно з проектом, усі нові проекти після 2023 року мають виключити використання 3DES. Також варто зазначити, що останній стандарт протоколу SSL/TLS, TLS 1.3, також припинив 3DES.

Симетричний алгоритм шифрування AES

AES - є одним із найпоширеніших алгоритмів шифрування. Він був розроблений як заміна DES і затвердився, як новий стандартом шифрування після схвалення NIST у 2001 році. AES — це набір блочних шифрів з різною довжиною ключа та різними розмірами блоків. AES працює з методами заміни та перестановки. Спочатку перетворіть незашифровані дані в блоки, а потім застосуйте ключове шифрування. Процес шифрування включає багато підпроцесів, таких як переміщення рядків, перемішування карток і додавання ключів. Залежно від довжини ключа виконується 10, 12 або 14 таких перетворень (раундів). Важливо зазначити, що фінальний раунд, на відміну від попередніх, не включає змішаний підпроцес.

Перевага використання алгоритму шифрування AES

Підводячи маленький підсумок, то aes - є безпечні, швидкі та гнучкі. Алгоритм AES набагато швидше, ніж DES. Варіанти з різною довжиною ключів є найбільшою перевагою: чим довші ключі, тим важче їх зламати.

Типи шифрування: симетричне та асиметричне:

Асиметричне шифрування має кілька ключів, які використовуються для шифрування та дешифрування даних, які повністю пов’язані один з одним.

Симетричне шифрування чудово підходить для обміну інформацією між нашими Алісою та Антоном.

Щоб обійти це, Антон використовує шифрування з незашифрованим ключем, де він дає відкритий ключ кожному, хто надсилає йому повідомлення, а приватний ключ носить із собою. Він наказує вам зашифрувати дані відкритим ключем, щоб їх можна було розшифрувати лише за допомогою закритого ключа. Це створює ризик скомпрометованих ключів, оскільки дані можна розшифрувати лише за допомогою приватного ключа, який належить Антону.

Першою (і найбільш очевидною) перевагою цього шифрування є безпека, яку воно забезпечує. У цьому підході відкритий відкритий ключ використовується для шифрування даних, тоді як розшифрування даних виконується за допомогою закритого ключа, який повинен надійно зберігатися.

Другою важливою особливістю асиметричного шифрування є аутентифікація. Як ми бачили, зашифровані з відкритим ключем дані можна розшифрувати лише за допомогою пов'язаного з ним приватного ключа. Таким чином, він гарантує, що тільки той об'єкт, який має отримати дані, зможе їх побачити та розшифрувати. Коротше кажучи, це підтверджує, що ви розмовляєте з реальною людиною чи організацією або ділитесь інформацією з ними.

Розглянемо два основних типи асиметричних алгоритмів шифрування.

1. Алгоритм RSA Алгоритм асиметричного шифрування

На сьогоднішній день це найпоширеніший алгоритм асиметричного шифрування. Його ефективність полягає в підході «розбив це один раз». По суті, два різних випадкових простих числа заданого розміру (наприклад, 1024 біта кожне) вибираються та перемножуються, щоб створити ще одне гігантське число. Завдання полягає в тому, щоб визначити вихідне просте число гіганта множника. Головоломка виявилася практично неможливою для сучасних суперкомп'ютерів, не кажучи вже про людей.

У 2010 році дослідження групи волонтерів показало, що для зламу 768-бітного ключа RSA знадобиться понад 1500 років обчислень (поширених на сотні комп'ютерів), що набагато менше, ніж стандартний 2048-бітний.

Перевага використання алгоритму шифрування RSA

Найбільшою перевагою RSA є його масштабованість, ключі можуть бути різної довжини шифрування: 768 біт, 1024 біт, 2048 біт, 4096 біт тощо.

RSA заснована на простій математиці, тому його легко інтегрувати в інфраструктуру відкритих ключів (PKI). Легкість і безпека роблять RSA – алгоритмом, який найбільше використовується для асиметричного шифрування в різних програмах, включаючи сертифікати SSL/TLS, шифрування та шифрування електронної пошти.

2. Алгоритм шифрування Алгоритм ECC

У 1985 році два математики, Ніл Кобліц і Віктор Міллер, запропонували використовувати еліптичні криві в криптографії.

Під час шифрування ECC поняття, як еліптична крива несе за собою набір точок, які задовольняють математичне рівняння (y 2 = x 3 + ax + b).

Еліптична крива

Цих два алгоритма дотримуються принципу незворотності. Простіше кажучи, в ECC існує число, яке зображує точку на кривій, воно множиться на інше число, щоб отримати іншу точку на кривій, що є логічним. Тепер, щоб вирішити це питання, ви повинні знайти нову точку на кривій. Математика ECC побудована таким чином, що навіть якщо ви знаєте початкову точку, знайти нові точки майже неможливо.

Переваги використання цього алгоритму:

Хоча ECC використовує меншу довжину ключа в порівнянні з RSA, він забезпечує більшу безпеку (від сучасних методів злому).

Ярлики вимагають меншого навантаження на мережу та обчислювальної потужності, що чудово підходить для пристроїв, які не можуть швидко обробляти, та мабть ліміти для зберігання. Використання алгоритму ECC в сертифікаті SSL/TLS значно скорочує час, необхідний для шифрування та дешифрування, допомагаючи швидше завантажувати веб-сайти. Алгоритми ECC використовуються в криптографічних програмах, цифрових підписах, генераторах псевдовипадкових даних тощо.

Однак проблема широкого використання ECC полягає в тому, що багато серверних програм і панелей керування ще не додали підтримку сертифікатів ECC/SSL/TLS. Ми сподіваємося, що це зміниться найближчим часом, і RSA залишиться найбільш широко використовуваним алгоритмом асиметричного шифрування на даний момент.

Висновки до другого розділу

Отже, QR-коди сьогодні є майже скрізь, оскільки багато сфер діяльності і користувачів впевнелись в ефективності та легкості QR-кодів. Чорні та білі квадрати, які легко розпізнати, допомагають людям вирішувати різноманітні проблеми в багатьох напрямках життя.

Навколо цієї теми розкривається тенденція до зростання використання QR-кодів у освіті та запропоновано практичні шляхи використання QR-кодів у навчальному процесі та прикладній діяльності навчальних закладів.

РОЗДІЛ 3
РОЗРОБКА БЕЗПЕЧНОГО МЕХАНІЗМУ АУТЕНТИФІКАЦІЇ З ВИКОРИСТАННЯМ QR-КОДУ

3.1. Програмна реалізація механізму аутентифікації з використанням QR-коду

Для виконання поставленої задачі, було прийнято рішення розробити модуль аутентифікації з використанням QR-коду. Насамперед, необхідно було вирішити, які функції буде містити в собі проект. Для цього побудовано функціональну модель. Головна перевага таких моделей є те, що завдяки схемам або блоксхемам є можливість змоделювати додаток або в даному випадку – модуль, оптимізувати функції та створити часові рамки процесу розробки [17]. Функціональне модулювання використовує графічну мову, щоб описати процеси та логічні дії, тому будь-яка функціональна модель буде зображена , як сукупність упорядкованих та взаємологічних діаграм. Для опису використовується дві методології:

- діаграма потоків даних;
- метод функціонального моделювання.

Функціональна модель модуля, зображена нижче, де представлено принцип його роботи (як показано на рисунку 3.1).

Розробка поставленого завдання в середовищі програмування Python:

Для механізму аутентифікації з використанням QR-коду були використанні наступні модулі Python: (як показано на рисунку 3.2).

- Json;
- Random;
- Time;
- Qr-code;
- Cryptography;
- Flask;

- Jinja2.

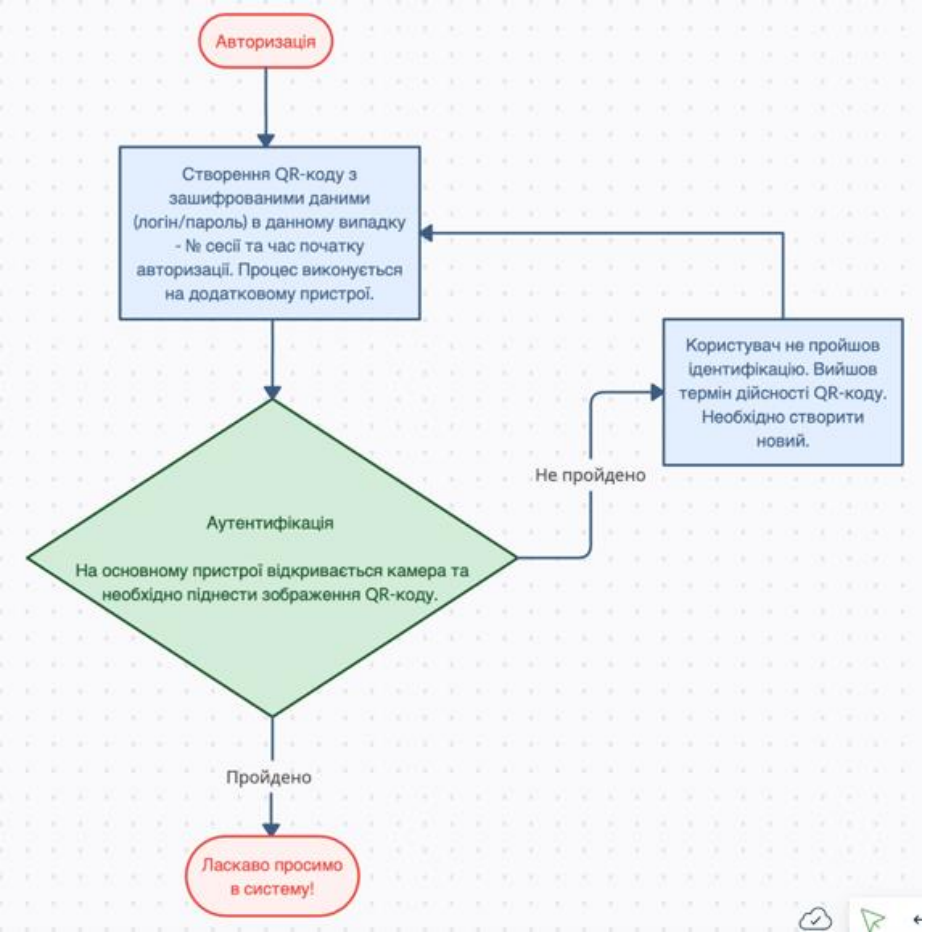

Рисунок. 3.1 Функціональна модель модуля

Json – за допомогою цього формату виконується обмін даними, який працює на JavaScript.

Random – це модуль, щоб генерування випадкові числа, букви, символів, вибираючи елементи випадково з послідовності.

Time – це модуль для роботи і оформлення часу в Python.

Qr-code – це модуль, який надає можливість створити або прокласти шлях до зображень QR-коду, який буде використовуватись у майбутньому.

Cryptograph – це модуль, який криптографія включає в себе як рецепти високого рівня, так і низькорівневі інтерфейси для поширених криптографічних алгоритмів, таких як симетричні шифри, дайджест повідомлень і функції виведення ключів. В роботу використовувався алгоритм Fernet, який гарантує, що

повідомлення, зашифроване за допомогою нього, не можна маніпулювати або прочитати без ключа. Fernet — це реалізація симетричної аутентифікованої криптографії. Fernet також підтримує реалізацію обертання ключів через MultiFernet.

Jinja2 – це шаблонізатор для мови програмування JavaScript.

Рисунок. 3.2 Частина коду 1

При запуску сервера, генерується ключ шифрування криптографічного модуля Fernet, створюємо саму функцію шифрування та дешифрування (як показано на рисунку 3.3):

Рисунок. 3.3 Частина коду 2

Створюємо додаток об’єкта Flask та додаємо дебагер, щоб в подальшому було зручно створювати та модифікувати поставлену задачу (як показано на рисунку 3.4):

Рисунок. 3.4 Частина коду 3

Додаємо словник сесій, де будуть зберігатися дані сесії, тобто кожен раз, як будемо намагатися пройти аутентифікацію (як показано на рисунку 3.5):

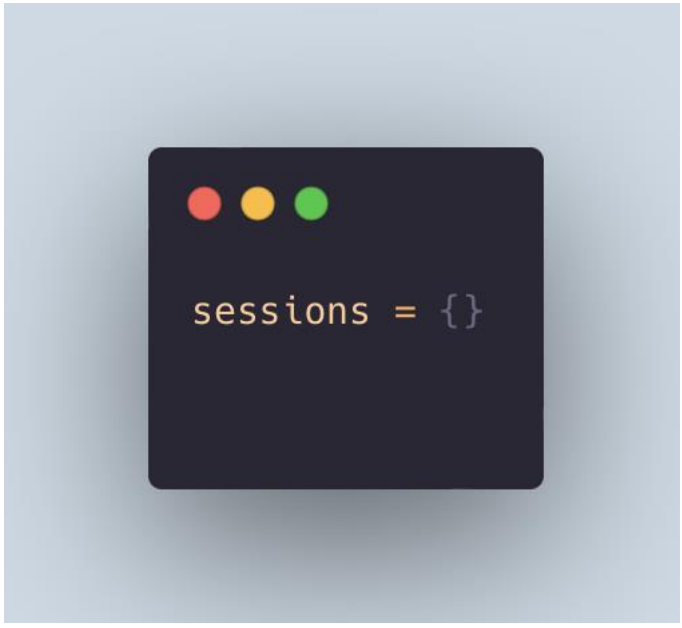

Рисунок. 3.5 Частина коду 4

Далі по структурі створюємо опрацювання запитів в веб-браузері/сервері аутентифікації:

- Запит переходу на головну сторінку (як показано на рисунку 3.6)::

Рисунок. 3.6 Частина коду 5

- Запит переходу на сторінку “Додаток на телефоні” (як показано на рисунку 3.7):

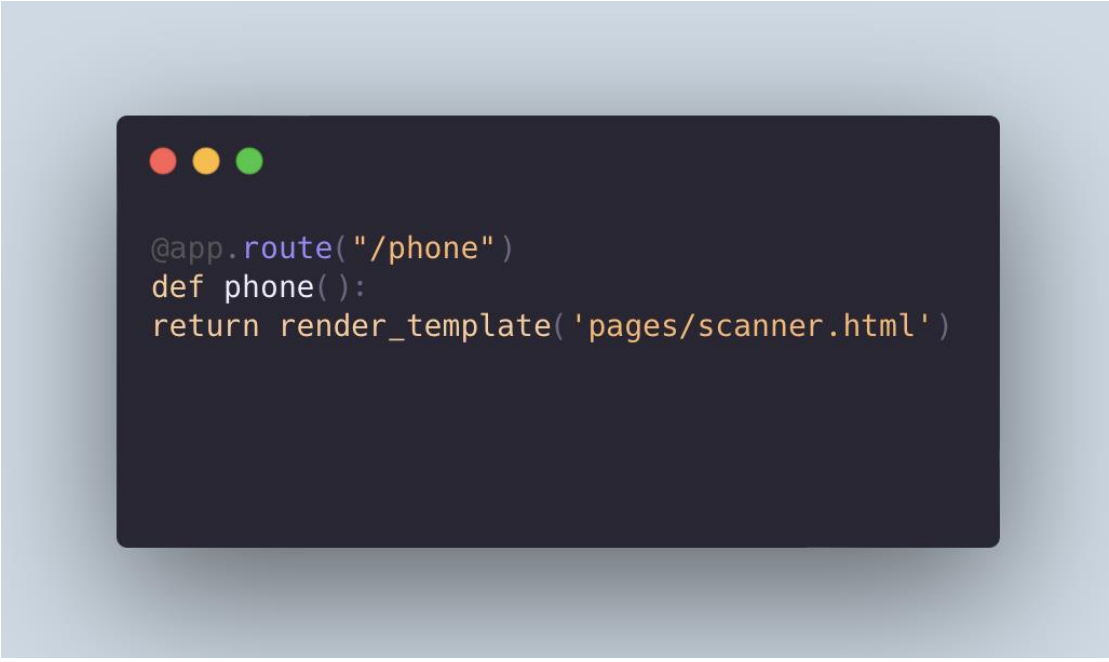

Рисунок. 3.7 Частина коду 6

- Запит на авторизацію, де створюється сесія, яка містить в собі наступні дані – id сесії та час її початку, потім вона додається в словник сесії, про який згадувалось вище. Згодом дані сесії, переводяться в json формат, щоб в результаті отримати звичайний закодований текстовий рядок. Після процесу шифрування рядка, отримується QR-код (як показано на рисунку 3.8):

Рисунок. 3.8 Частина коду 7

- Запит на перевірку авторизації, отримуються дані сесії, виконується перевірка чи є такий id-сесії в словнику. Перевірка часу, якщо пройшло більше ніж 20 секунд, то QR-код стає не дійсним, сесія видаляється та користувач отримує помилку на екрані “Time is out”. Якщо ж користувач ввійшов в таймінг, то після авторизації отримує повідомлення “OK” та дані сесії. Якщо ж id-сесії немає в словнику, отримується помилка “Error session not found” (як показано на рисунку 3.9):

```
@app.route("/check-auth", methods=['POST'])
def check_auth():
try:
    data = request.json
            session_id = int(data['session_id'])
         if session_id in sessions:
        if time() - sessions[session_id]['start_time'] > 20:
            del sessions[session_id]
                    answer = {
                'status': 'ERROR',
                                'error_message': 'time is out!'
            }
        else:
            answer = {
                'status': 'OK',
                                'user': sessions[session_id]['user']
            }
        else:
                answer = {
                      'status': 'ERROR',
                                    'error_message': 'session not found'
                }
        except Exception as e:
answer = {
                'status': 'ERROR',
                    'error_message': repr(e)
}
            return json.dumps(answer)
```

Рисунок. 3.9 Частина коду 8

• Запит на сканування QR-коду. Отримується контент, який шифрувався вище, він дешифрується, результатом є json файл, що був на початку. Json файл перекладається в дані, які бути на початку, в даному випадку – це дані сеccії (номер та час). Перевіряється чи є id-сесії, який зчитався с QR-коду, в словнику. Якщо id – дійсний, то користувач успішно пройшов аутентифікацію. В другому випадку, якщо id не має в словнику, користувач отримує помилку типу “Session not found”. (як показано на рисунку 3.10):

```
@app.route("/send-auth", methods=['POST'])
def send_auth():
try:
data = request.json
session = f.decrypt(data['content'].encode())
session = json.loads(session)
user = data['user']
if session['id'] in sessions:
sessions[session['id']]['user'] = user
                    answer = {
                            'status': 'OK',
                        'session': session
                    }
            else:
                    answer = {
                        'status': 'ERROR',
                     'error_message': 'session not found'
                }
        except Exception as e:
            answer = {
                    'status': 'ERROR',
                    'error_message': repr(e)
                }
            return json.dumps(answer)
```

Рисунок. 3.10 Частина коду 9

Під час дослідження було поставлене завдання розробити безпечний механізм аутентифікації з використанням Qr-коду. Модуль розроблявся на об’єктно-орієнтованій мові програмування Python. Проект підтримує числове і буквено-числове кодування за допомогою алгоритму Fernet. При такому кодуванні є можливість закодувати будь-що, наприклад букви, цифри, номера, адресі тощо.

Загалом модуль працює наступним чином, необхідно мати два пристрої, пристрій на якому необхідно пройти аутентифікацію, та той за допомогою якого

створюється QR-код, це може бути ноутбук і телефон, або два телефона. На пристрої, де маємо пройти аутентифікацію, в даному випадку - це ноутбук відкриваємо головну сторінку, яка виглядає наступним чином (як показано на рисунку 3.11):

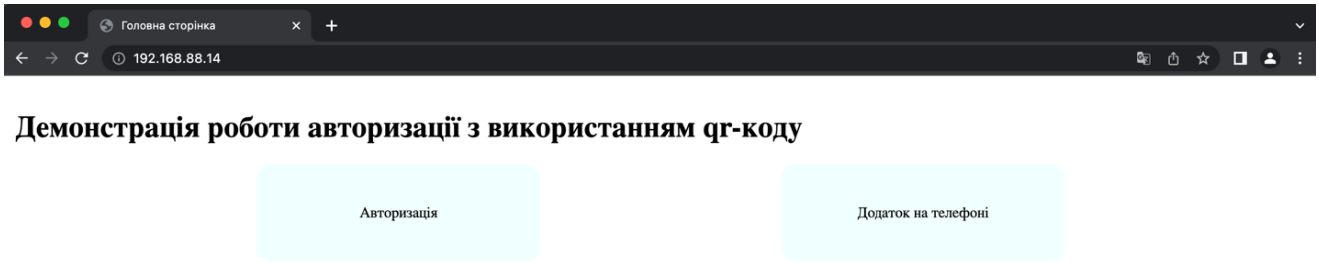

Рисунок. 3.11 Головна сторінка

Далі, відкриваємо цю саму сторінку на додатковому пристрої, щоб створити QR-код та натискаємо “авторизація” (як показано на рисунку 3.12):

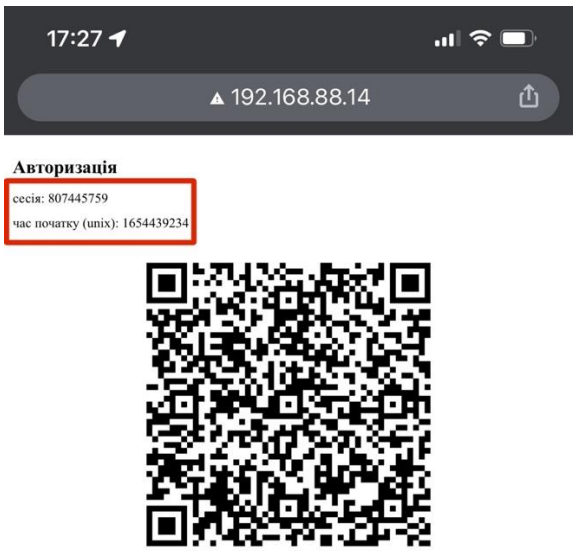

Рисунок. 3.12 Сторінка авторизації

Бачимо, що згенерувався QR-код, в якому знаходяться дані, які в червоному

прямокутнику, в даному випадку – це номер (id) сесії та час її початку, замість цього може бути будь-що, наприклад логін пароль.

Примітка: Якщо користувач не пройшов аутентифікацію протягом 20 секунд, QR-код стає не дійсним, та він отримує таку помилку (показано на рисунку 3.13).

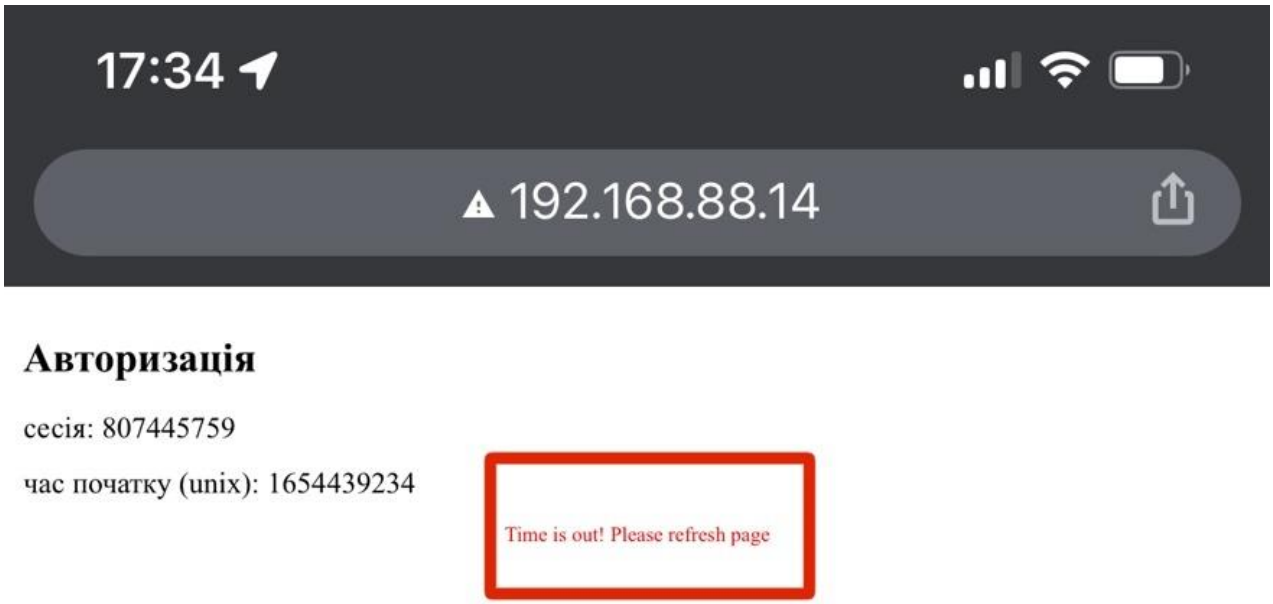

Рисунок. 3.13 Помилка QR-коду

На основному пристрої натискаємо “Додаток на телефоні”, відкривається вікно, де вмикається камера, щоб просканувати QR-код, та є можливість вибрати користувача який має пройти аутентифікацію (як показано на рисунку 3.14):

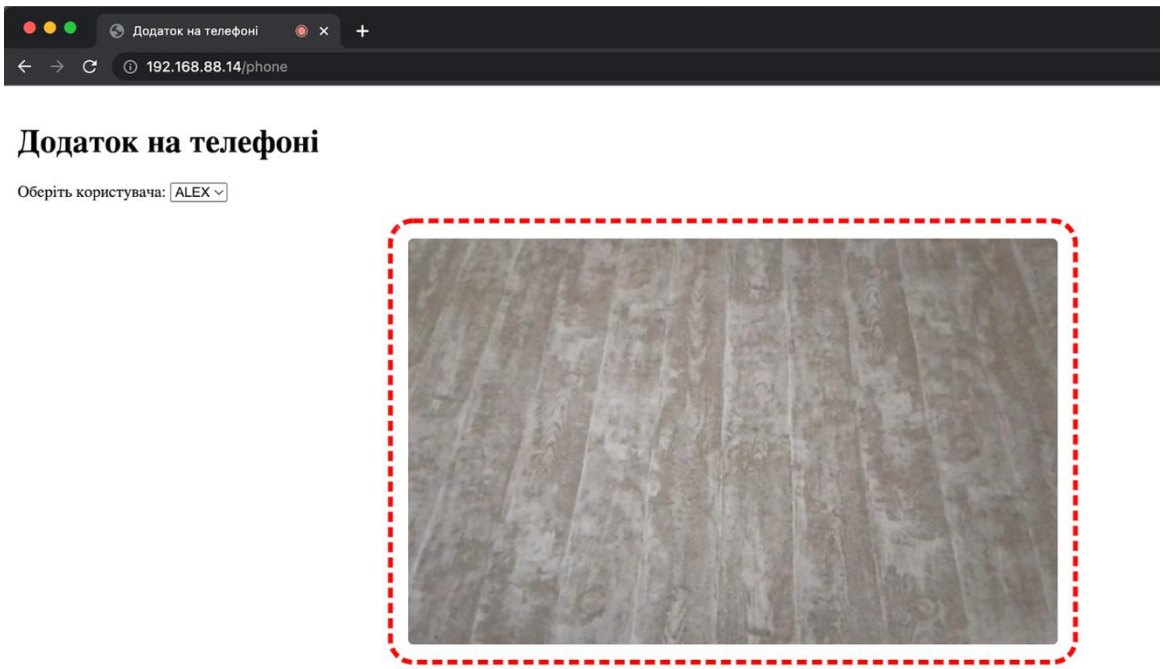

Рисунок. 3.14 Сторінка аутентифікації

Далі підносимо пристрій, на якому згенерувався QR-код до камери (як показано на рисунку 3.15):

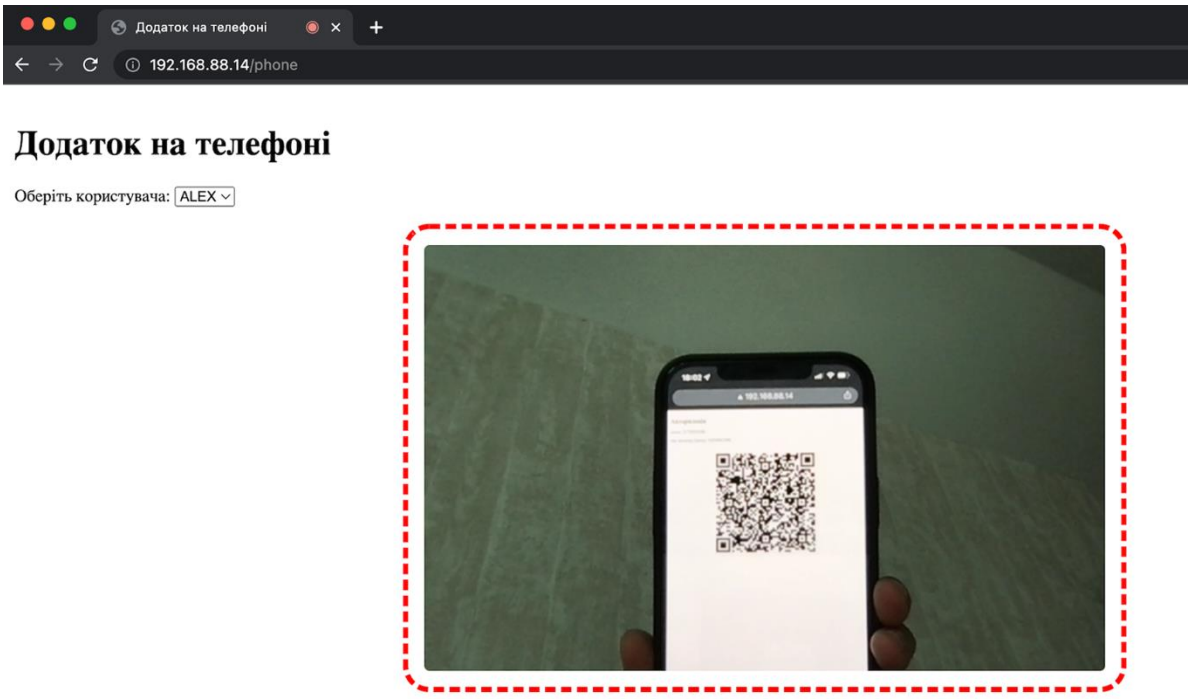

Рисунок. 3.15 Процес авторизації

Після того, якщо QR-код правильний, користувач проходить авторизацію отримує доступ, в даному випадку – це дані сесії та імя користувача у відкритому вигляді. В червоному прямокутнику наявно показано зашифрований вигляд даних, які містяться в QR-коді – це набір букв та цифр (як показано на рисунку 3.16):

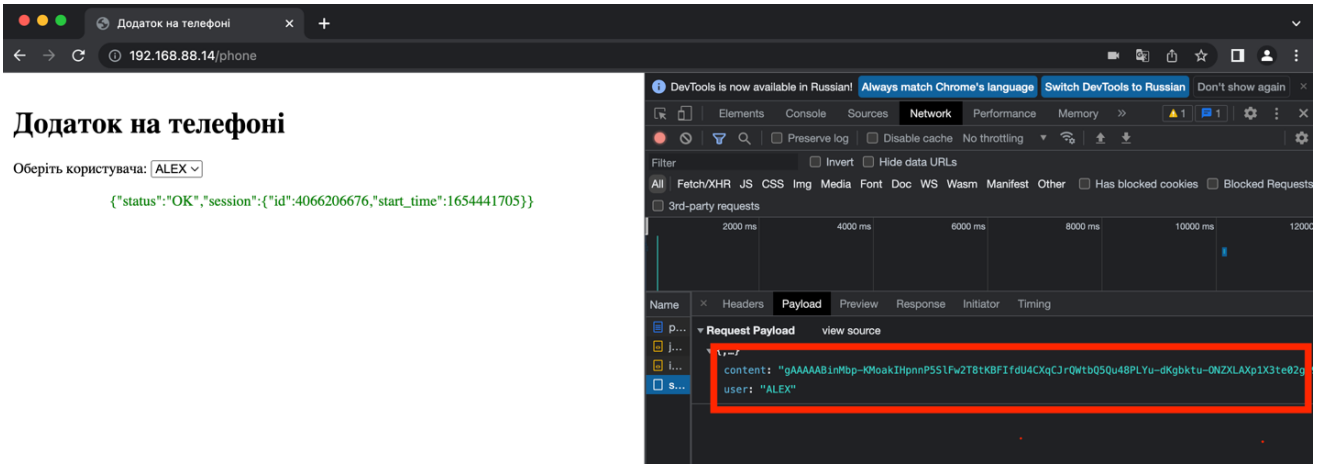

Рисунок. 3.16 Успішна авторизація

Така логічна послідовність дій для авторизації працює і в зворотному порядку та на будь-яких пристроях.

3.2. Перспективи та можливості використання механізму

Таким чином, вивчивши теоретичну частину про аутентифікацію, опанувавши принципи роботи QR-кодів та створивши програмний продукт для аутентифікації через QR-код, можна виділити наступні можливості використання механізму:

- Можна використовувати у навчальному процесі, де кожний студент або учень буде отримувати персональний доступ до свого онлайн кабінету, де є інформація, нехай, про його успішність.
- Можливе використання механізму на будь-який сайтах в мережі Інтернет, де необхідно створювати особистий кабінет, тобто реєструватись.
- Окремо виділяємо застосування модуля аутентифікації у приватних компаніях, які мають особисті бази даних, щоб робітники проходили

аутентифікацію кожного разу, як приступали до роботи, адже дані у базах – конфіденційна інформація.

Висновки до третього розділу

На основі поставленої задачі було створено модуль, що дозволяє проходити аутентифікацію за допомогою QR-коду з використанням криптографічного шифрування даних, що безпезпечує їх конфіденційність та цілісність.

Програмне середовище для реалізації проекту вибиралося в силу популярності та зручності роботи в ньому. В якості мови програмування, на якій був розроблений механізм аутентифікації, було обрано Python, адже це мова має безліч бібліотек з якими легко та просто працювати.

ВИСНОВКИ

У проведеному дослідженні була проаналізована наукова та методична література в області аутентифікації за допомогою QR-коду, шифрування та самих QR-кодів. З проведеного аналізу літератури були зроблені висновки про принципи роботи, про структуру QR-коду, про його матриці і способи його кодування та про методи шифрування інформації. Матриця QR-коду, його структура, а так само всі її складові частини були детально вивчені. Таким чином, обробивши теоретичну інформацію, зрозумівши принципи роботи QR-коду та створивши програмний продукт для аутентифікації за допомогою QR-коду було запропоновано наступні можливості та перспективи використання механізму: можна використовувати у навчальному процесі, де кожний студент або учень буде отримувати персональний доступ до свого онлайн кабінету, де є інформація, нехай, про його успішність.

Можливе використання механізму на будь-який сайтах в мережі Інтернет, де необхідно створювати особистий кабінет, тобто реєструватись.

Окремо виділяємо застосування модуля аутентифікації у приватних компаніях, які мають особисті бази даних, щоб робітники проходили аутентифікацію кожного разу, як приступали до роботи, адже дані у базах – конфіденційна інформація.

Дана робота та механізм аутентифікації написані з урахуванням того, що будуть в майбутньому удосконалюватись, адже цей модуль аутентифікації використовується у компанії ТОВ ТД “Прометей Агро”. В планах на магістерську роботу розробити програму для баз даних компанії та використати цей механізм для забезпечення конфіденційності інформації.

Удосконалення плануються наступні:

- візуалізація;
- розмежування доступу;
- нові опції.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Pidru4niki − [Електронний ресурс] Режим доступу: https://pidru4niki.com (дата звернення 19.05.2021р.) − Назва з екрану.

2. Msdn.microsoft − [Електронний ресурс] Режим доступу: https://msdn.microsoft.com (дата звернення 19.05.2021р.) − Назва з екрану.

3. PHP QR Code − [Електронний ресурс] Режим доступу: http://phpqrcode.sourceforge.net (дата звернення 19.05.2021р.) − Назва з екрану.

4. Qrcc − [Електронний ресурс] Режим доступу: http://qrcc.ru/generator.php (дата звернення 19.05.2021р.) − Назва з екрану.

5. Qrcode − [Електронний ресурс] Режим доступу: http://qrcode.com.ua (дата звернення 19.05.2021р.) − Назва з екрану.

6. Qrcode.kaywa − [Електронний ресурс] Режим доступу: http://qrcode.kaywa.com (дата звернення 20.05.2021р.) − Назва з екрану.

7. Qrcoder − [Електронний ресурс] Режим доступу: http://qrcoder.ru (дата звернення 20.05.2021р.) − Назва з екрану.

8. Qr-coder − [Електронний ресурс] Режим доступу: http://qr-coder.ru (дата звернення 20.05.2021р.) − Назва з екрану.

9. Qreambee − [Електронний ресурс] Режим доступу: http://qreambee.ru (дата звернення 21.05.2021р.) − Назва з екрану.

10. QREncode − [Електронний ресурс] Режим доступу: https://packages.altlinux.org/ru/Sisyphus/srpms/qrencode (дата звернення 21.05.2021р.) − Назва з екрану.

11. QR-код. Добавление служебной информации − [Електронний ресурс] Режим доступу: https://ru.wikipedia.org/wiki/QR-код (дата звернення 22.05.2021р.) − Назва з екрану.

12. QR-код. Этап размещения информации на поле кода − [Електронний ресурс] Режим доступу: https://ru.wikipedia.org/wiki/QR-код(дата звернення

22.05.2021р.) − Назва з екрану.

13. QR-код. Кодирование − [Електронний ресурс] Режим доступу: https://ru.wikipedia.org/wiki/QR-код (дата звернення 22.05.2021р.) − Назва з екрану.

14. Messaging made easy − [Електронний ресурс] Режим доступу: http://platform.twit88.com (дата звернення 22.05.2021р.) − Назва з екрану.

15. ZXing (Zebra Crossing) − [Електронний ресурс] Режим доступу: https://github.com/zxing/zxing/wiki/Getting-Started-Developing (дата звернення 22.05.2021р.) − Назва з екрану.

16. ГОСТ Р ИСО/МЭК 18004-2015 Спецификация символики штрихового кода QR-code − Москва: Стандартинформ, 2015. − 113 с.

17. МЕТОДОЛОГИЯ ФУНКЦИОНАЛЬНОГО МОДЕЛИРОВАНИЯ IDEF0 / Москва: ГОССТАНДАРТ РОССИИ, 2000. 75 с. – Електрон. аналог друк. вид.: режим доступу: https://nsu.ru/smk/files/idef.pdf (дата звернення 22.05.2021р.) − Назва з екрану.

18. О.М. Котов Язык C#. Краткое описание и введение в технологии программирования. − Екатеринбург: Издательство Уральского университета, 2014. − 208 с.

19. Таблица Unicode − [Електронний ресурс] Режим доступу: https://geektimes.ru/post/256932/ (дата звернення 22.05.2021р.) – Назва з екрану.

20. Gonzoblog − [Електронний ресурс] Режим доступу: http://gonzoblog.ru/post/2009/02/02/Dvuhmernie_shtrihkodi.aspx (дата звернення 23.05.2021р.) – Назва з екрану.

21. Web.archive − [Електронний ресурс] Режим доступу: https://web.archive.org/web/20091201190328/http://rubymag.ru/articles/primerispolzovaniya-qr-code (дата звернення 23.05.2021р.) – Назва з екрану.

22. Apple − [Електронний ресурс] Режим доступу: https://support.apple.com/uk-ua/guide/iphone/iphe8bda8762/13.0/ios/13.0 (дата звернення 23.05.2021р.) – Назва з екрану.

23. Леонід Бугаїв. Мобільний маркетинг. Як зарядити свій бізнес у мобільному світі, 2012 – 167 с.

24. Зчитування QR-кодів − [Електронний ресурс] Режим доступу: http://www.mobile-barcodes.com/qr-code-software/ (дата звернення 23.05.2021р.) – Назва з екрану.

25. Computer Bild No12 2011, Обзор «Код QR» − [Електронний ресурс] Режим доступу: http://www.mobile-barcodes.com/qr-code-software/ (дата звернення 23.05.2021р.) – Назва з екрану.

26. Daring Librarian «Чертова дюжина идей использования QR-кода» − [Електронний ресурс] Режим доступу: http://solbiblfil2.ucoz.ru/index/chertova_djuzhina_idej_ispolzovanija_qr_koda_ot_daring_librarian/ 0-159 / (дата звернення 24.05.2021р.) – Назва з екрану.

27. А. Баданов «TagMyDoc» / Интерактивности – WEB сервисы для образования» − [Електронний ресурс] Режим доступу: https://sites.google.com/site/badanovweb2/home/tagmydoc (дата звернення 23.05.2021р.) – Назва з екрану.

28. Читаємо QR-код − [Електронний ресурс] Режим доступу: https://habr.com/ru/post/127197/ (дата звернення 23.05.2021р.) – Назва з екрану.

29. Шифр − [Електронний ресурс] Режим доступу: http://cyclop.com.ua/content/view/1450/1/1/5/#7616/ (дата звернення 23.05.2021р.) – Назва з екрану.

30. Reinholm, James H. Classification of Cryptographic Keys (Functions & Properties). Cryptomathic. − [Електронний ресурс] Режим доступу: https://www.cryptomathic.com/news-events/blog/classification-or-cryptographic-keys (дата звернення 23.05.2021р.) – Назва з екрану.

31. Management. National Institute of Standards and Technology (NIST) − [Електронний ресурс] Режим доступу: https://www.cryptomathic.com/news-events/blog/classification-or-cryptographic-keys (дата звернення 23.05.2021р.) – Назва з екрану.

32. Barker, Elaine. NIST Special Publication 800-57 Part 1 Revision 4: Recommendation for Key − [Електронний ресурс] Режим доступу: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r4.pdf (дата

звернення 23.05.2021р.) – Назва з екрану.

33. Остапов, Євсеєв, Король; розділ: 10.1 Основні положення керування ключами. Життєвий цикл криптографічного ключа

34. Остапов, Євсеєв, Король; розділ: 10.4 Безпека керування ключами

35. M. Rabbani, R. Joshi, “An overview of the JPEG2000 still image compression standard” , Eastman Kodak Company, Rochester, NY 14650, USA, Signal Processing: Image Communication, 2012, c.38-44

ДОДАТОК А

Лістинг модуля

```
import json
import random
from time import time

import qrcode
from cryptography.fernet import Fernet
from flask import Flask, render_template, request

key = Fernet.generate_key()
f = Fernet(key)

app = Flask(__name__)
app.debug = True

sessions = {}



@app.route("/")
def index():
    return render_template('pages/index.html')



@app.route("/phone")
def phone():
    return render_template('pages/scanner.html')
```

```
@app.route("/auth")
def auth():
    session_id = int(''.join(str(random.randint(0, 9)) for x in range(10)))
    session = {
        'id': session_id,
        'start_time': int(time())
    }
    sessions[session_id] = session
    img = qrcode.make(f.encrypt(json.dumps(session).encode()))
    img.save(f'./static/qr/{session_id}.png')
    return render_template('pages/auth.html', session=session)


@app.route("/check-auth", methods=['POST'])
def check_auth():
    try:
        data = request.json
        session_id = int(data['session_id'])
        if session_id in sessions:
            if time() - sessions[session_id]['start_time'] > 20:
                del sessions[session_id]
                answer = {
                    'status': 'ERROR',
                    'error_message': 'time is out!'
                }
            else:
                answer = {
                    'status': 'OK',
                    'user': sessions[session_id]['user']
```

```
            }
        else:
          answer = {
           'status': 'ERROR',
           'error_message': 'session not found'
        }
    except Exception as e:
        answer = {
            'status': 'ERROR',
            'error_message': repr(e)
        }
    return json.dumps(answer)


@app.route("/send-auth", methods=['POST'])
def send_auth():
    try:
        data = request.json
        session = f.decrypt(data['content'].encode())
        session = json.loads(session)
        user = data['user']
        if session['id'] in sessions:
            sessions[session['id']]['user'] = user
            answer = {
                'status': 'OK',
                'session': session
            }
        else:
          answer = {
           'status': 'ERROR',
```

```
            'error_message': 'session not found'
        }
    except Exception as e:
        answer = {
            'status': 'ERROR',
            'error_message': repr(e)
        }
    return json.dumps(answer)



if __name__ == "__main__":
    app.run('192.168.88.14', 80)
```