МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ СИСТЕМ ЗАХИСТУ ІНФОРМАЦІЇ

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри
_____ С.В. Казмірчук
«____» _____ 2020 р.

На правах рукопису
УДК 004.056.5:510.22(043.3)

ДИПЛОМНА РОБОТА

ЗДОБУВАЧА ВИЩОЇ ОСВІТИ
ОСВІТНЬОГО СТУПЕНЯ «МАГІСТР»

Тема: Удосконалений метод автентифікації користувачів за допомогою Bluetooth технології

Виконавець: П.Д. Панченко

Науковий керівник: к.т.н., доц. А.Б. Петренко

Нормоконтролер: к.т.н., доц. А.Б. Петренко

Київ 2020

ВСТУП

Існування і розвиток інформаційного суспільства на сучасному етапі неможливе без використання інформаційних мереж, глобальних комп'ютерних мереж і мереж зв'язку - радіо, телебачення, фіксованих і мобільних телефонних мереж, Інтернет тощо. Ідентифікація, автентифікація і авторизація - це три взаємопов'язані поняття, які складають основу системи безпеки. Автентифікація - це процес підтвердження особи користувача.

Автентифікація є невід’ємною складовою робочого циклу будь-якого програмного продукту. З теперішнім розвитком інформаційних технологій, процес автентифікації піддається суттєвим змінам майже щодня. Все популярнішим стає метод багатофакторної автентифікації, його використовують скрізь: від програм обміну повідомленнями, до банківських додатків. Атаки, що спрямовані на обхід автентифікації, стали причиною застосування двох (а іноді і більше) факторів для підтвердження особи користувача.Мета роботи – реалізація системи моніторингу мережевого трафіку для захисту локальної інформації на підприємстві

Об'єкт дослідження – процес автентифікації користувачів

Предмет дослідження – сучасні методи автентифікації користувачів

Методи досліджень. Проведені дослідження в даній роботі базуються на сучасних методах аналізу механізмів захисту автентифікації.

Виходячи з мети, завданням даної дипломної роботи є:

аналіз сучасних методів автентифікації користувачів; розробка удосконаленого методу автентифікації за допомогою Bluetooth технологій; впровадження розробленого методу в програмний продукт на ОС Windows, використовуючи мову C#.

Практична значимість роботи полягає у авторській розробці програмного модулю реалізації розробленого методу автентифікаці з використанням

об'єктно-орієнтованої мови програмування C# на базі використанням інтегрованого середовища розробки Visual Studio.

Наукова новизна. Удосконалено метод багатофакторної автентифікації, за рахунок використання Bluetooth пристроїв задля автентифікації, а саме використання унікальних властивостей Bluetooth пристроїв, що дозволило проводити автентифікацію швидко та надійно, без необхідності втручання користувача в цей процес.

Розділ 1. ЗАГАЛЬНІ ВІДОМОСТІ ПРО АВТЕНТИФІКАЦІЮ, ВРАЗЛИВОСТІ ТА ПРОБЛЕМИ БЕЗПЕКИ.

1.1. Система безпеки доступу до ресурсів. Поняття автентифікації.

«Ідентифікація», «автентифікація» і «авторизація» - це три взаємопов'язані поняття, які складають основу системи безпеки. Ідентифікація - це передача ідентичності інформаційній системі (далі – ІС). Перед автентифікацією заявник зазвичай все одно надає ІС посвідчення (наприклад, логін або адресу електронної пошти), а монітор стверджує посвідчення шляхом автентифікації (наприклад, за допомогою пароля).

Автентифікація - це процес підтвердження особи користувача. Потім спостерігач підтверджує ІС особистість користувача. Користувачі ідентифікуються з використанням різних механізмів автентифікації. В системі безпеки процес автентифікації перевіряє інформацію, надану користувачем, за допомогою бази даних [1], [2].

Нарешті, авторизація - це надання користувачеві привілей. Схему доступу до ресурсів показано на Рис. 1.1

Рис. 1.1. Схема доступу до ресурсів

Системи автентифікації дають відповіді на питання: «хто є користувачем?» і «чи дійсно користувач є тим, ким він / вона собою являє?». Отже, автентифікація є один з найбільш багатообіцяючих способів підвищення довіри і безпеки комерційних додатків.

Зв'язок між позивачем та спостерігачем позначений як канал. Канал - це опора спілкування позивача і спостерігача. Його можна розглядати як конфіденційний, достовірний, безпечний або небезпечний. Конфіденційний канал стійкий до перехоплення; автентичний канал стійкий до злому; безпечний канал стійкий до обох; а незахищений канал не стійкий до обох.

Загальні базові кроки для автентифікації:

1. Початковий крок: позивач не автентифікований;

2. Етап підключення: заявник вимагає від ІС використання функції, що вимагає автентифікація. ІС просить спостерігача перевірити справжність заявника.

3. Крок автентифікації: заявник автентифікований і відкривається сеанс. ІС надає користувачеві необхідні функції.

4. Крок відключення: користувач відключається або відключається від спостерігача, і стан повертається до початкового кроку. Цей крок може бути ініційований після певного періоду, або дією користувача.

Для ІС можуть знадобитися різні рівні автентифікації, наприклад, рівень для адміністраторів і рівень для користувачів. У такій системі рівень автентифікації градуюється за шкалою: рівень 0 для нерозпізнаних користувачів з найнижчими правами в системі; рівень N для адміністратора з повними правами; і один або кілька рівнів від 0 до N. Тут схема полягає в тому, що для перемикання на більш високий рівень довіри з боку ІС може знадобитися автентифікація [3].

Процес автентифікації може бути заснований на поєднанні одного або декількох факторів автентифікації. П’ять широко визнаних факторів для автентифікації людини:

1. Те, що знає користувач: пароль, кодова фраза, PIN-код, дівоче прізвище матері;

2. Те, що належить користувачеві: USB-токен, телефон, смарт-карта, програмний токен, cookie-файл навігатора;

3. Те, що кваліфікує користувача: відбиток пальця, фрагмент ДНК, зразок голосу, геометрія руки;

4. Те, що може зробити користувач: підпис, жест;

5. Де знаходиться користувач: поточне місце розташування / позиція, інформація на поточний час;

Протокол автентифікації - це тип протоколу комп'ютерного зв'язку або криптографічного протоколу, спеціально розроблений для передачі даних автентифікації між двома об'єктами. [4]

Завдання протоколу автентифікації - вказати точну серію кроків, необхідних для виконання автентифікації. Він повинен відповідати основним принципам протоколу:

1. Протокол повинен включати дві або більше сторін, і кожен, хто бере участь в протоколі, повинен знати протокол заздалегідь;

2. Всі залучені сторони повинні дотримуватися протоколу;

3. Протокол повинен бути однозначним - кожен крок повинен бути точно визначений;

4. Протокол повинен бути повним, тобто включати вказану дію для кожної можливої ситуації;

Існує безліч різних систем, до яких користувачеві необхідний доступ, і тому протоколи автентифікації, як правило, є відкритими стандартами - автор далі описує найбільш часто використовувані протоколи:

1. Протокол автентифікації пароля (далі – ПАП). ПАП - один з найстаріших протоколів автентифікації. Автентифікація через ПАП ініціалізується тим, що клієнт відправляє пакет з обліковими даними (ім'я користувача і пароль) на початку з'єднання, при цьому клієнт повторює запит автентифікації до тих пір, поки не буде отримано підтвердження. [5] Це вкрай небезпечно, тому що облікові дані пересилаються «у відкритому вигляді» і неодноразово, що робить його вразливим навіть для найпростіших атак.

2. Протокол автентифікації запит-рукостискання. Процес автентифікації в цьому протоколі завжди ініціалізується сервером / хостом і може виконуватися в будь-який час протягом сеансу, навіть повторно. Сервер відправляє випадковий рядок (зазвичай довжиною 128 Б). Клієнт використовує пароль і рядок, отримані в якості параметрів для геш-функції MD5, а потім відправляє результат разом з ім'ям користувача у вигляді звичайного тексту. Сервер використовує ім'я користувача для застосування тієї ж функції і порівнює обчислений і отриманий геш. Автентифікація пройшла успішно або невдало.

3. Розширений протокол автентифікації (далі – РПА). Автентифікація РПА ініціюється сервером (автентифікатором). Обмін автентифікацією РПА відбувається наступним чином: автентифікатор відправляє запит на автентифікацію однорангового вузла; вузол відправляє пакет відповіді у відповідь на дійсний запит; автентифікатор відправляє додатковий пакет запиту, і вузол відповідає; діалог триває до тих пір, поки автентифікатор не зможе автентифікувати однорангового вузла (неприйнятні відповіді на один або кілька запитів), і в цьому випадку реалізація автентифікатора повинна передати повідомлення про помилку РПА; в якості альтернативи, діалог автентифікації може тривати до тих пір, поки автентифікатор не з'ясує, що автентифікація пройшла успішно, і в цьому випадку автентифікатор повинен передати успіх РПА [6].

4. TACACS, XTACACS і TACACS +. Використовує автентифікацію на основі IP без будь-якого шифрування (імена користувачів і паролі передавалися у вигляді простого тексту). У більш пізньої версії XTACACS (Extended TACACS) додані авторизація та облік. Пізніше обидва цих протоколу були замінені на TACACS +. TACACS + розділяє компоненти, тому вони можуть бути відокремлені і оброблятися на окремих серверах (він навіть може використовувати інший протокол, наприклад, для авторизації). Він використовує TCP (протокол управління передачею) для транспортування і шифрує весь пакет. TACACS + є власністю Cisco.

5. Служба віддаленої автентифікації користувачів з телефонним підключенням (RADIUS). Протокол, що часто використовується інтернет-провайдером. Облікові дані в основному засновані на комбінації імені користувача та пароля, для транспорту використовується протокол NAS і UDP. [7]

6. DIAMETR. Походить від RADIUS і включає в себе безліч поліпшень, таких як використання більш надійного транспортного протоколу TCP або SCTP і більш високий рівень безпеки завдяки TLS. [8]

7. Kerberos. Централізована мережева система автентифікації, доступна як безкоштовна реалізація від Массачусетського технологічного інституту, а також у багатьох комерційних продуктах. Це метод перевірки автентичності за замовчуванням в Windows 2000 і пізніших версіях. Сам процес автентифікації набагато складніше, ніж в попередніх протоколах - Kerberos використовує криптографію з симетричним ключем, вимагає довіреної третьої сторони і може використовувати криптографію з відкритим ключем на певних етапах автентифікації, якщо це необхідно. [9]

1.2. Класифікації автентифікації

1.2.1. З точки зору виду метода

Метою автентифікації є твердження особистості, але набір методів автентифікації дуже великий і може варіюватися по-різному. Нижче наведено список деяких поширених методів автентифікації:

1. Автентифікація за допомогою паролю;
2. Автентифікація за допомогою смарт-карти;
3. Біометрична автентифікація;
4. Автентифікація за допомогою цифрового сертифікату.

Кожен з методів автентифікації має своє застосування і недоліки. Токени можуть бути вкрадені, системи розпізнавання осіб можуть бути зламані, це стосується надійності методу автентифікації. Отже, метою автентифікації є перевірка ідентичності об'єкта із заданим рівнем довіри. Якщо метод перевірки автентичності не може бути повністю надійним, надана перевірка також не може бути надійною.

1.2.2. З точки зору кількості методів

Існує чотири типи автентифікації з точки зору кількості методів:

1. Однофакторна автентифікація;
2. Двофакторна автентифікація;
3. Трьохфакторна автентифікація;
4. Чотирьохфаторна автентифікація.

Однофакторна автентифікація використовує один метод автентифікації. Багатофакторна автентифікація - це багаторівнева система безпеки, яка перевіряє особистість користувачів для входу в систему або інших транзакцій [10]. Двофакторна, трьохфакторна та чотирьохфакторна автентифікації є підмножинами багатофакторної автентифікації.

Двофакторна автентифікація являє собою двоетапний процес автентифікації. Він поєднує в собі ім'я користувача і пароль або PIN-код з фізичним або мобільним токеном для додаткової безпеки. Таке поєднання чинників автентифікації ускладнює доступ потенційного зловмисника.

Трьохфакторна автентифікація поєднує в собі фактори «Я знаю», «Я маю», «Я є». Як і при двофакторній автентифікації, «Я знаю» та «Я маю», зазвичай включає імена користувачів, паролі та одноразовий токен. Однак з трьохфакторною автентифікацією є додатковий фактор – «Я є», - який вико-

ристовує біометричні дані, такі як відбитки пальців, для перевірки особи користувача.

Чотирьохфакторна автентифікація - це ще одна форма багаторівневої безпеки, яка включає в себе знання, володіння, приналежність і місце розташування. Як і у випадку з троьхфакторною, знання, володіння і приналежність складаються з паролів і PIN-кодів, автентифікації токена і біометрії. Для додаткового рівня безпеки чотирьохфакторна автентифікація також використовує перевірку логіна користувача для автентифікації користувача.

Завдяки використанню декількох рівнів автентифікації, навіть якщо один елемент пошкоджений або відключений, обліковий запис користувача залишиться в безпеці.

Традиційні імена користувачів і паролі легко зламати. Фактично, вони дуже вразливі для кіберзлочинів, таких як груба сила і захоплення акаунтів.

З іншого боку, багатофакторна автентифікація останнім часом вважається одним з найбільш ефективних способів підвищення безпеки. Багаторівневість гарантує, що користувачі, які потребують доступу, є тими, ким вони себе називають. Навіть якщо кіберзлочинці вкрадуть один обліковий запис, вони будуть змушені перевіряти особистість іншим способом.

1.2.3. З точки зору рівня безпеки

Найбільш часті типи автентифікації, що використовуються для автентифікації онлайн-користувачів, різняться за рівнем безпеки, що забезпечується об'єднанням чинників з однієї або декількох з трьох категорій факторів автентифікації:

1. «Сильна» автентифікація;
2. Неперервна автентифікація;
3. Електронна автентифікація.

«Сильна» автентифікація - автентифікація, при якій користувач ототожнюється з реальною особою, організацією або довірителем. Дуже часто «сильну» автентифікацію ототожнюють з двофакторною, але це невірно.

«Сильна» автентифікація не завжди може бути багатофакторною: відповідь на кілька питань також буде «сильною» автентифікацію. Наприклад, при дзвінку в банк з метою ідентифікації часто запитують номер карти, номер паспорта, місце народження і так далі. Все це потрапляє під один фактор «я знаю» і значить не є багатофакторною автентифікацією, але тим не менш є «сильною» автентифікацією. На практиці в таких випадках рекомендують застосовувати багатофакторну автентифікацію.

31 січня 2013 року Європейський Центральний Банк випустив ряд обов'язкових рекомендацій з безпеки інтернет-платежів. Основна рекомендація полягає в тому, що ініціювання інтернет-платежів, а також доступ до конфіденційних платіжних даних повинні бути захищені сильною клієнтською автентифікацією, щоб гарантувати, що платіж ініціює законний користувач, а не шахрай.

Це став перший в світі нормативно-правовий документ, де офіційно фігурує і визначається це поняття. 1 лютого 2015 року на території ЄС увійшло в силу обов'язкове дотримання даних рекомендацій.

Безперервна автентифікація - це метод перевірки, спрямований на забезпечення підтвердження особи та захисту кібербезпеки на постійній основі. Постійно вимірюючи ймовірність того, що окремі користувачі є тими, ким вони себе називають, безперервна автентифікація перевіряє користувача не тільки один раз, а й безперервно протягом усього сеансу. Безперервна автентифікація, орієнтована на забезпечення інтелектуальної і безпечної перевірки особистості без переривання робочого процесу, реалізується з використанням машинного навчання і безлічі чинників, включаючи поведінкові моделі і біометрії.

Хоча безперервна перевірка справжності є відносно новим явищем, цей тип перевірки привертає увагу, оскільки компанії шукають нові способи запобігання несанкціонованому доступу до критично важливих бізнес-даних. Традиційні форми перевірки, такі як однофакторна автентифікація, яка за-

безпечує захист при вході в систему, і двофакторна автентифікація, яка додає другий рівень безпеки при вході в систему, не пропонують безперервної перевірки ідентифікації користувача. Потреба в нових стратегіях управління ідентифікацією та доступом, таких як безперервна автентифікація, зростає в результаті стрімкого розвитку цифрових технологій і ескалації кіберзлочинності.

Рішення управління ідентифікацією та доступом з функцією безперервної автентифікації постійно збирає інформацію про дії користувача і шаблонах звичайної поведінки і вчиться розрізняти нормальну і ненормальну поведінку користувача на основі зібраних даних. На основі аналізу поведінки користувача може бути надано доступ до системи або може бути запрошена додаткова перевірка особи користувача.

Варіації і невідповідності в поведінці і взаємодії користувача з системою можуть бути виміряні, або фізіологічні характеристики користувача можуть бути визначені безперервно під час сеансу. Крім того, якщо користувач поводиться погано або знаходиться під загрозою, доступ може бути відкликаний, а сеанс завершиться негайно. Можливі методи виявлення змін включають в себе натискання клавіш, відео, відбитки пальців, дотик (сила натискання пальця) або риси обличчя, такі як положення очей, розмір зіниці і частота моргання.

Додаток з функцією безперервної автентифікації може постійно обчислювати «оцінку автентифікації», щоб визначити, наскільки точно власник облікового запису також використовує пристрій. Залежно від оцінки користувачеві може бути запропоновано ввести додаткову інформацію, таку як пароль, карту або відбиток пальця.

Електронна автентифікація - це процес встановлення впевненості в ідентифікаційних даних користувача, представлених в електронній формі в інформаційній системі. Цифрова автентифікація або електронна автентифікація можуть використовуватися як синоніми, коли мова йде про процес автенти-

фікації, який підтверджує або посвідчує особу людини і працює. При використанні разом з електронним підписом він може служити доказом того, чи були підроблені отримані дані після підписання їх початковим відправником. Електронна автентифікація може знизити ризик шахрайства та крадіжки особистих даних за рахунок підтвердження того, що людина є тим, ким вона є, при виконанні транзакцій в Інтернеті.

Існують різні методи електронної автентифікації, які можуть використовуватися для автентифікації користувача, від пароля до більш високих рівнів безпеки, які використовують багатофакторну автентифікацію. В залежності від використовуваного рівня безпеки користувачеві може знадобитися підтвердити свою особистість за допомогою токенів безпеки, контрольних питань або наявності сертифіката стороннього центру сертифікації, що підтверджує його особу.

Існує чотири типи схем електронної автентифікації: локальна автентифікація, централізована автентифікація, глобальна централізована автентифікація, глобальна автентифікація і веб-додаток (портал).

При використанні локальної схеми автентифікації програма зберігає дані, які стосуються до облікового запису користувача. Ця інформація зазвичай не передається іншим програмам. Відповідальність за підтримання і запам'ятовування типів і кількості облікових даних, пов'язаних зі службою, до якої він повинен отримати доступ, лежить на користувачі. Це схема високого ризику через можливість того, що область зберігання паролів може бути скомпрометована.

Використання центральної схеми автентифікації дозволяє кожному користувачеві використовувати одні і ті ж облікові дані для доступу до різних служб. Кожна програма індивідуальна і має мати інтерфейси та можливість взаємодії з центральною системою для успішної автентифікації користувача. Це дозволяє користувачеві отримати доступ до важливої інформації і мати

доступ до закритих ключів, які дозволять йому або їй підписувати документи електронним способом.

Використання третьої сторони через глобальну централізовану схему автентифікації дозволяє користувачеві отримати прямий доступ до служб автентифікації. Потім це дозволяє користувачеві отримати доступ до потрібних їм службам.

Найбезпечніша схема - це глобальна централізована автентифікація і веб-додаток (портал). Вони ідеально підходять для використання в електронному уряді, оскільки надає широкий спектр послуг. Вони використовують єдині механізми автентифікації, що включає мінімум два фактори, щоб дозволити доступ до необхідних службам і можливість підписувати документи.

1.3. Стандарт та правове регулювання автентифікації

FIPS 113 — Computer Data Authentication [11] визначає стандарт, який використовується федеральними організаціями, які вимагають криптографічної автентифікації цілісності комп'ютерних даних. Крім того, він може використовуватися будь-якою організацією щоразу, коли потрібно криптографічна автентифікація. Криптографічна автентифікація даних під час передачі між електронними компонентами або при зберіганні необхідна для підтримки цілісності інформації, представленої даними. Стандарт визначає алгоритм криптографічної автентифікації (Data Authentication Algorithm, далі - DAA) для використання в системах та мережах ADP. Алгоритм автентифікації використовує криптографічний алгоритм стандарту шифрування даних DES.

DAA являє собою основу, на якій можуть бути побудовані розширені протоколи автентифікації. У сфері хмарних обчислень, Інтернету речей і великих даних DAA відіграють життєво важливу роль для захисту переданих, збережених і обмінюваних даних і інформації. Ці базові та розширені алгоритми діляться на чотири основні типи. Паролі та хеш-функції є найпростішими схемами автентифікації. Геш-функції використовуються в більш складних конструкціях алгоритмів автентифікації, таких як MAC і цифрові підписи.

Вибір підкреслених математичних стандартних блоків, таких як еліптичні криві, сприяло поліпшенню алгоритмів автентифікації; в даний час є алгоритми з високою продуктивністю обчислень і більш короткими криптографічними ключами. Крім криптографічної області автентифікації, біометричні і стеганографічні схеми використовуються як інші засоби алгоритмів автентифікації, де характеристики користувача є ключами для автентифікації його в системі [12].

В українському законодавстві визначення терміну «автентифікація» описано в першому розділі Закону України «Про електронні довірчі послуги» [13].

НД ТЗІ 2.5-004-99 «Критерії оцінки захищеності інформації в комп’ютерних системах від несанкціонованого доступу» [14] установлює критерії оцінки захищеності інформації, оброблюваної в комп'ютерних системах, від несанкціонованого доступу. В критерії спостереженості описані такі послуги як автентифікація, автентифікація при обміні, автентифікація відправника та автентифікація отримувача.

1.4. Проблеми безпеки автентифікації

Атаки на процес автентифікації поділяються на три основних групи [15]:

1. Атаки на користувацький інтерфейс;
2. Атаки на базу даних шаблона;
3. Атаки на системні модулі та взаємозв’язки між модулями.

Що стосується атак на користувацький інтерфейс, автентифікація відіграє важливу роль в безпеці веб-додатків. Коли користувач надає своє ім'я для входу і пароль для автентифікації і підтвердження своєї особи, додаток призначає системі певні привілеї користувача на основі ідентифікатора, встановленого наданими обліковими даними.

HTTP може включати кілька різних типів протоколів автентифікації:

1. Базовий - ім'я користувача / пароль у відкритому вигляді, кодування Base-64;

2. Дайджест - як базовий, але паролі зашифровані;

3. На основі форми - налаштовується форма, що використовується для введення імені користувача / пароля (або інших облікових даних) і обробляється з використанням налаштованої логіки в бекенді.

4. NTLM - власний протокол автентифікації Microsoft, реалізований в заголовках HTTP-запиту / відповіді.

5. Negotiate - новий протокол від Microsoft, який дозволяє клієнту і серверу динамічно погоджувати будь-який тип автентифікації, вказаний вище. Також додає Kerberos для клієнтів, які використовують Microsoft IE v5 +.

6. Сертифікати на стороні клієнта. Хоча SSL / TLS використовується не часто, він надає можливість перевірки автентичності цифрового сертифікату, представленого веб-клієнтом, що, по суті, робить його токеном автентифікації.

7. Microsoft Passport - служба єдиного входу (SSI), керована корпорацією Microsoft, яка дозволяє веб-сайтам (так званим «Passport Partners») автентифікувати користувачів на основі їх членства в службі Passport. Механізм використовує ключ, яким користуються Microsoft і партнерським сайтом, для створення файлу cookie, який однозначно ідентифікує користувача.

Ці протоколи автентифікації працюють прямо через HTTP (або SSL / TSL) з обліковими даними, вбудованими прямо в трафік запиту / відповіді.

Цей вид атаки не є технологічною дірою в безпеці операційної системи або серверного програмного забезпечення. Це скоріше залежить від того, наскільки надійно і складно зберігаються паролі, а також від того, наскільки легко зловмисникові дістатися до сервера.

Коли зловмисник вторгається в систему, доводячи додатку, що він є відомим і чинним користувачем, зловмисник отримує доступ до тих привілеїв, які адміністратор призначив цьому користувачу.

Це означає, що якщо зловмисникові вдасться увійти в систему як звичайний користувач, він може мати обмежений доступ тільки для перегляду деякої важливої інформації. З іншого боку, якщо йому вдасться увійти як адміністратору з глобальним доступом до системи, він буде мати майже повний контроль над додатком разом з його вмістом (з обмеженнями самого вебдодатку).

Зазвичай зловмисник спочатку намагається отримати доступ до екрану запрошення / входу в систему, де додаток запитує логін і пароль. Наступним кроком буде введення правильного збігу логіна і пароля, яке додаток розпізнає як правильне і яке має високі привілеї в системі. [16]

В таблиці 1.1 наведено поширені атаки, спрямовані на процес автентифікації.

Таблиця 1.1

Типи атак на процес автентифікації

Атака	Опис
Метод грубої сили	Метод злому пароля шляхом вичерпної обробки всіх можливостей, щоб знайти пароль. При атаці методом грубої сили зловмисник: 1. Захоплює деяку форму гешування паролів; 2. Атакує файл геша в автономному режимі за допомогою зломщика грубої сили.
Словник	Спроба вгадати пароль, переглядаючи список слів зі словника. Часто символи і букви верхнього і нижнього регістра замінюються всередині словникової роботи. Це контрастує з атакою грубої сили, в якій перевіряються всі можливості. Атака по словнику працює, тому що користувачі часто вибирають паролі, що легко вгадати. Надійна політика паролів - кращий захист

	від словникових атак.
Аналіз паролів (сніффінг)	Спроба перехоплення паролів, що проходять по комп'ютерній мережі. Зазвичай для захоплення пакетів в мережі використовуються програми. Потім зловмисник аналізує пакети, щоб визначити, які з них містять паролі. Шифрування забезпечує кращий захист від атак сніффінга. Такі технології, як SSL, SSH і IPSEC, забезпечують рівень захисту, що виходить за рамки традиційної мережевої компонування і конструктивних контрзаходів.
Спуфінг	Атака використовується, щоб приховати справжнє джерело пакетів або перенаправити трафік в інше місце. Найбільш поширеною формою підміни типового IP-пакета є зміна адреси джерела. Таким чином ховається правильна адреса вихідного пристрою. Спуфінгові атаки: 1. Використовує в пакетах змінені адреси джерела і / або призначення; 2. Може включати спуфінг сайту, який обманом змушує користувачів розкрити інформацію.
Атака посередника	Використовуються для перехоплення інформації, що передається між двома партнерами по зв'язку. Атаки типу посередника: 1. Зловмисник логічно знаходиться між клієнтом і сервером. Клієнта обманом змушують пройти автентифікацію для зловмисника; 2. Обидві сторони на кінцевих точках вважають, що вони обмінюються даними безпосередньо з іншим,

	в той час як зловмисник перехоплює і / або змінює дані в дорозі. Потім зловмисник автентифікується на сервері, використовуючи перехоплені облікові дані; 3. Зазвичай використовується для крадіжки кредитних карт, облікових даних онлайн-банку, а також конфіденційної особистої та ділової інформації.
Повторне відтворення	Зловмисник перехоплює і записує повідомлення. Захоплений трафік використовується в інший час, щоб спробувати відтворити автентифікацію.
Злом	Атака, при якій зловмисник краде відкритий і активний сеанс зв'язку у легітимного користувача (розширення атаки типу посередника): 1. Зловмисник захоплює сеанс і відключає вихідний пристрій; 2. Стан сеансу TCP / IP регулюється так, щоб зловмисник міг вставляти альтернативні пакети в потік зв'язку. [17]

BlackICE - один з найбезпезпечніших мережевих екранів для Windows. Розроблений ISS (Internet Security Systems), компанією по забезпеченню безпеки в Інтернеті, яка тестує Windows, а також інші небезпечні додатки. Перевіряються як вхідні, так і вихідні мережеві потоки, і, якщо є підозри, що щось не так, адміністратор отримує попередження і з'єднання може бути закрито. Функції, які викликаються атаками автентифікації, представлені в таблиці 1.2.

Таблиця 1.2

Функції автентифікації

Функція	Опис
HTTP_Auth_ContainsBinary	Шукає HTTP-автентифікацію,

	яка містить двійкові дані.
HTTP_Auth_TooLong	Виявляє рядок авторизації HTTP, довжина якої перевищує задане системою значення максимальної довжини авторизації HTTP.
HTTP_Authentication	Виявляє базову автентифікацію HTTP на веб-сервері і реєструє імена користувачів і паролі.
HTTP_Authentication_Format_String	Виявляє атаку рядка формату базової автентифікації HTTP в іменах користувачів і паролях.
HTTP_IIS_Hit_Highlighting_Auth_Bypass	Здійснює пошук спроб обійти обмеження безпеки за допомогою вразливості в функціях виділення збігів сервера Microsoft IIS.
HTTP_Login_Known_User	Виявляє ім'я входу і зіставляє його з користувацькими іменами входу для відомих імен входу.
HTTPS_ClearText_Session	Виявляє дійсний HTTP-запит і відповідь на порт 443, який не зашифрований. [18]

Заходи протидії атакам на автентифікацію:

1. Реалізація політики надійних паролів;
2. Збереження історії паролів, щоб запобігти їх повторне використання;
3. Використання багатофакторної автентифікації;

4. Використання строгої системи порядкової нумерації;

5. Використання тимчасових міток на кадрах, щоб відбити атаку відтворення;

6. Аудит на предмет надмірної кількості невдалих спроб входу в систему;

7. Контролювання мережі або системи на предмет наявності інструментів для перехоплення і крадіжки паролів;

8. Впровадження блокування облікового запису при використанні декількох неправильних паролів.

Пропозиції щодо посилення паролів:

1. Паролі повинні містити кілька типів символів: прописні, рядкові, числа і символи;

2. Мінімальна довжина пароля - вісім або більше символів;

3. Не використовувати частину імені користувача або адреси електронної пошти;

4. Уникання словника, сленгу і скорочень;

5. Зміна пароля кожні 30 днів.

Що стосується атак на бази даних шаблонів, біометричні дані, що зберігаються в базі даних шаблонів, можуть бути змінені або вилучені. Зловмисники можуть вносити зміни в базу даних для отримання доступу або контролю над захищеними ресурсами. Вони також можуть перешкодити авторизованим користувачам отримати свої права доступу. Незаконне отримання біометричного шаблону відоме як витік інформації. Витік може викликати серйозні проблеми, оскільки він не тільки забезпечує доступ не уповноваженій особі, а й порушує вимоги конфіденційності даних біометричної системи. Після крадіжки або підробки біометричних даних їх неможливо відновити або замінити, як в інших системах автентифікації.

Для захисту біометричного шаблону були запропоновані різні методи, такі як скасована біометрія [19] і дрібна біометрія [20]. Захист за допомогою

скасованої біометрії включає навмисне повторюване спотворення прийнятого біометричного сигналу на основі певного перетворення. З іншого боку, метод дробової біометрії маскує частину біометричних даних перед відправкою.

Атаки на системні модулі пов'язані зі зміною внутрішніх компонентів. Вони можуть виконуватися в модулях попередньої обробки, вилучення ознак, зіставлення і прийняття рішень. Одна з них полягає в тому, що шкідливе програмне забезпечення прикидається одним з модулів і відправляє вихідні дані, що належать противнику, в наступні модулі, такі як атака троянського коня [21].

З іншого боку, атаки на взаємозв'язок між модулями загрожують конфіденційності і цілісності даних каналу зв'язку, такі як атаки посередника і повторне відтворення (див. таблицю 1.1). Атака сходження на пагорб (hill-climbing attack) є порушенням безпеки, яке впливає на шлях від датчика до добувача ознак і від добувача ознак до порівнювача. Він націлений на отримання балів, необхідних для підтвердження особи, з наступною зміною існуючого біометричного зразка або набору характеристик [22].

Перехід від одного фальшиво згенерованого виведення до іншого контролюється шляхом підвищення оцінки відповідності, яка виражає силу зв'язку між наданими та збереженими біометричними даними. Щоб домогтися успіху, ця загроза повинна мати можливість безпосередньо надавати системі необроблені дані або характеристики біометричних зразків. Він також повинен отримати відповідну оцінку. Це вважалося найнебезпечнішою загрозою. Основна проблема з атакою по горбах - це величезна втрата, яку вона може завдати. Вона не тільки проходить через систему і впливає на цілісність, але також ставить під загрозу особистість користувача в будь-якій системі автентифікації, яка досліджує ту ж біометричну характеристику.

Надійна біометрична система являє собою рішення для відображення цих атак. Вона утримує модулі разом в одному місці або логічно зв'язані за

допомогою взаємної автентифікації, безпечного виконання коду або спеціалізованого обладнання з захистом від несанкціонованого доступу.

1.5. Висновки до розділу

Ідентифікація, автентифікація і авторизація - це три взаємопов'язані поняття, які складають основу системи безпеки. Автентифікація - це процес підтвердження особи користувача. Системи автентифікації дають відповіді на питання: «хто є користувачем?» і «чи дійсно користувач є тим, ким він / вона собою являє?». Для ІС можуть знадобитися різні рівні автентифікації, наприклад, рівень для адміністраторів і рівень для користувачів.

Класифікація автентифікації відбувається з точки зору метода (автентифікація за допомогою паролю, смарт-карти, цифрового сертифікату та біометрична автентифікація), з точки зору кількості цих методів (однофакторна, двуфакторна, трьохфакторна, чотирьохфакторна) та з точки зору рівня безпеки («сильна», безперервна, електронна).

Якщо зловмисникові вдасться увійти в систему як звичайний користувач, він може мати обмежений доступ тільки для перегляду деякої важливої інформації. З іншого боку, якщо йому вдасться увійти як адміністратору з глобальним доступом до системи, він буде мати майже повний контроль над додатком разом з його вмістом (з обмеженнями самого веб-додатку).

Атаки на процес автентифікації поділяються на три основних типи: атаки на користувацький інтерфейс, атаки на базу даних шаблона, атаки на системні модулі та взаємозв’язки між модулями. Що стосується атак на бази даних шаблонів, біометричні дані, що зберігаються в базі даних шаблонів, можуть бути змінені або вилучені. Найпоширенішими атаками є метод грубої сили, словник, сніффінг, спуфінг, атака посередника, повторне відтворення, злом. Атаки на системні модулі пов'язані зі зміною внутрішніх компонентів. Вони можуть виконуватися в модулях попередньої обробки, вилучення ознак, зіставлення і прийняття рішень.

Не один з методів не є цілком безпечним, тому зараз найбільш актуаль-

ним є метод двофакторної/багатофакторної автентифікації, тому що якщо один з видів автентифікації виявиться скомпрометованим, то інший залишається захищеним.

Розділ 2. ДОСЛІДЖЕННЯ МЕТОДІВ АВТЕНТИФІКАЦІЇ. АВТЕНТИФІКАЦІЯ ЗА ДОПОМОГОЮ BLUETOOTH ТЕХНОЛОГІЇ.

2.1. Загальні відомості про методи автентифікації

За останні декілька років навіть найбільші компанії переконались в тому, що не захищені від порушень безпеки. Системи великих компаній, таких як LinkedIn, Target, Home Depot і Sony Pictures, були зламані, що дозволило розкрити конфіденційну інформацію їх власників, співробітників і клієнтів. У зв'язку з тим, що були розкриті мільйони паролів, адрес електронної пошти та багато чого іншого, на тих, хто займається безпекою підприємства, чиниться тиск, щоб посилити свій захист.

Існує таке поняття, як фактор автентифікації - різні відмінні риси суб'єкта, його характеристики. Зараз використання тієї чи іншої характеристики в системі залежить від необхідної надійності, захищеності і вартості впровадження. Виділяють 3 фактора автентифікації:

1. Я знаю;

2. Я маю;

3. Я є.

Оскільки важко наздогнати тому, як швидко кіберзлочинці можуть поліпшити свої знання про системи, мережеві адміністратори зіткнулися з безліччю проблем і були змушені почати впроваджувати більш складні способи автентифікації користувачів. Нижче автор описує загальні методи автентифікації, що використовуються для безпеки, щоб обійти сучасні кіберзлочини:

1. Автентифікація за допомогою паролю;

Цей тип автентифікації вимагає від користувача згадати те, що він знає. Цей метод складається з двох частин. По-перше, користувач вводить ім'я, а

по-друге, пароль. Пароль - це секретна комбінація слів і цифр, відома користувачеві.

Коли користувач створює пароль, копія цих облікових даних зберігається системою або веб-сайтом в захищеній базі даних паролів, з якої сервер може порівнювати будь-які подальші спроби входу в систему. Оскільки всі ці паролі зберігаються в централізованому місці, важливо, щоб системи автентифікації на основі паролів забезпечували першокласну безпеку цих баз даних.

Одна з переваг полягає в тому, що довший пароль дуже складно зламати. При використанні паролів обов'язково використовувати надійні паролі. Надійний секретний ключ складається з великих і малих літер, цифр і унікальних символів. Тепер адміністратори безпеки рекомендують паролі з 12 символів. [23] Для злому пароля з 12 символів, потужності 94 і ентропії 78,7 біт на суперкомп'ютерах буде потрібно 55 днів. А при використанні ПК на злом потрібно 3018 років.

Зазвичай паролі зберігаються в зашифрованому вигляді, так що навіть якщо хакер зможе отримати доступ до бази даних, інформація, яку він бачить, буде для нього марна. Це називається обробкою і гешуванням паролів.

Гешування паролів переводить їх у випадковий набір символів. Після гешування даних декодувати інформацію без ключа стає вкрай складно. Коли користувачі входять в систему, застосовується те ж гешування, а потім порівнюється інформація у файлі.

Метод «соління» [24] додає додаткове значення в кінці паролів, щоб було важче визначити, які фактичні облікові дані. Наприклад, пароль «admin» зміниться на «admin + salt» при додаванні солі. В якості додаткової міри безпеки сіль, що додається до кожного паролю, повинна бути випадковою і унікальною.

Як правило, «соління» відбувається після гешування пароля, а це означає, що два процеси часто працюють разом, щоб забезпечити додаткові рівні

безпеки. За допомогою «соління» і гешування паролів додаються нові засоби захисту в базу даних, що ускладнює розшифрування даних і отримання доступу до інформації користувачів.

Однак важливо розуміти, що в міру вдосконалення технології на основі паролів інструменти кіберзлочинців часто не відстають. Фактично, деякі хакери з'ясували, як проводити атаки методом перебору паролів навіть після того, як вони були оброблені і гешовані.

2. Автентифікація за допомогою смарт-карти;

Автентифікація за допомогою смарт-карти - це фактор «Я маю». Смарт-карта - це карта розміром з кредитну, в яку вбудований сертифікат, який використовується для ідентифікації власника. Користувач може вставити карту в пристрій читання смарт-карт для автентифікації людини. Смарт-карти зазвичай використовуються з PIN-кодом, що забезпечує багатофакторну автентифікацію. Іншими словами, користувач повинен мати щось (смарт-карту) і щось знати (PIN-код).

Одна з переваг смарт-карт полягає в тому, що вони доступні в двох варіантах. По-перше, в комплекті йде карта пам'яті зі зберіганням даних з двофакторною автентификацією. По-друге, вона оснащена процесором, що робить двофакторну автентифікацію більш надійною. Смарт-карта з мікропроцесором зберігає сертифікат відкритого і закритого ключів. Смарт-карта блокується, якщо PIN-код введено невірно після кількох спроб. Смарт-карта запобігає словниковій атаці. Вона портативна і може легко переноситися користувачами. Крім того, інформацію, що зберігається на смарт-карті, не можна легко видалити, змінити або відновити. Навіть якщо смарт-карта потрапить в руки зловмисників, малоймовірно, що хтось зможе створити дублікат і порушити безпеку [25].

3. Біометрична автентифікація;

Біометричні методи забезпечують фактор «Я є». Біометрична автентифікація користувача - це метод, який ідентифікує користувача і / або переві-

ряє його особистість на основі вимірювання їх унікальних фізіологічних рис або поведінкових характеристик. Фізіологічна біометрія - це відбиток пальця, розпізнавання осіб, сканування райдужної оболонки, геометрія руки, сканування сітківки. Поведінкова біометрія - це розпізнавання голосу, ходи, сканування натискання клавіш і сканування підпису. Відбитки пальців і рук - найбільш широко використовуваний сьогодні біометричний метод. Багато ноутбуків оснащені сканерами відбитків пальців, а також доступні зчитувачі відбитків пальців на USB-накопичувачах.

Біометрична автентифікація широко використовується і має велику надійність: позбавляє користувача від складного завдання з відновлення паролів; біометричні дані унікальні і прості; дуже складно відтворити біометричні характеристики; біометричні характеристики не можуть бути втрачені; сканування відбитків пальців невелике і недороге; може використовуватися по телефонних лініях; сканування очей - це точність в ідентифікації користувачів.

Найпоширеніший біометричний метод - сканер відбитків пальців, статистику використання якого на 2014 та 2020 роки можна побачити на Рис. 2.1 та 2.2 відповідно. Сканер відбитків пальців ідентифікує зображення на пальці користувача, а потім зіставляє його з даними в базі даних. У кожного користувача різні характеристики малюнка пальців. Коли шаблон за відбитками пальців зберігається в базі даних, система перетворює його в двійковий файл.

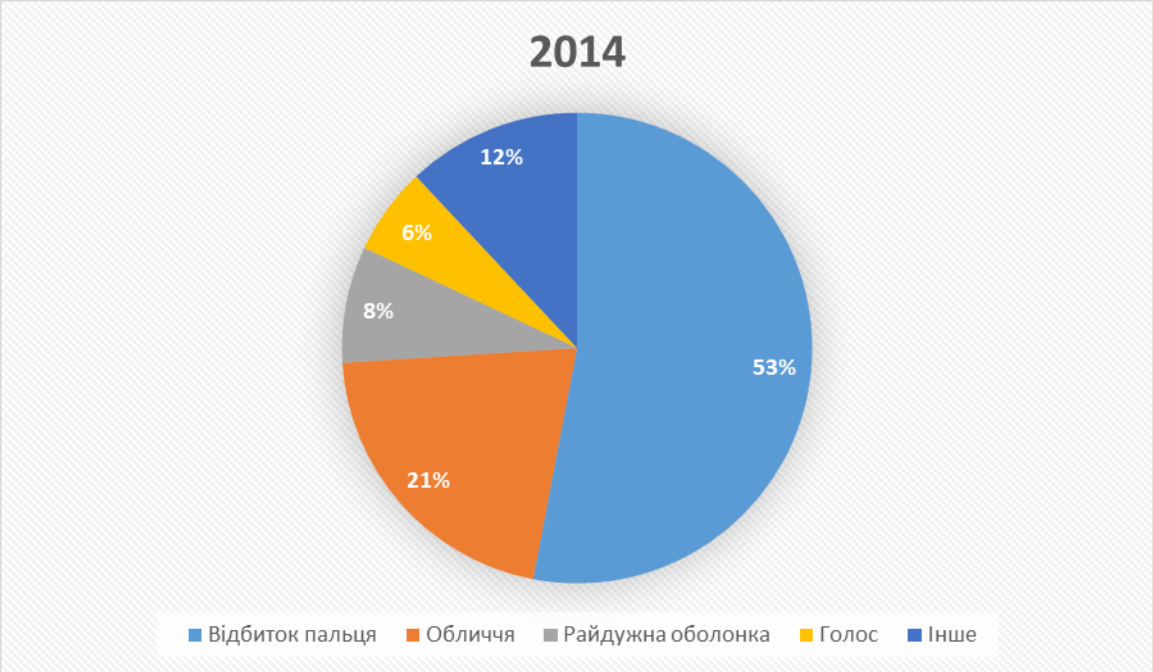

Рис. 2.1. Статистика використання методів біометричної автентифікації на 2014 рік

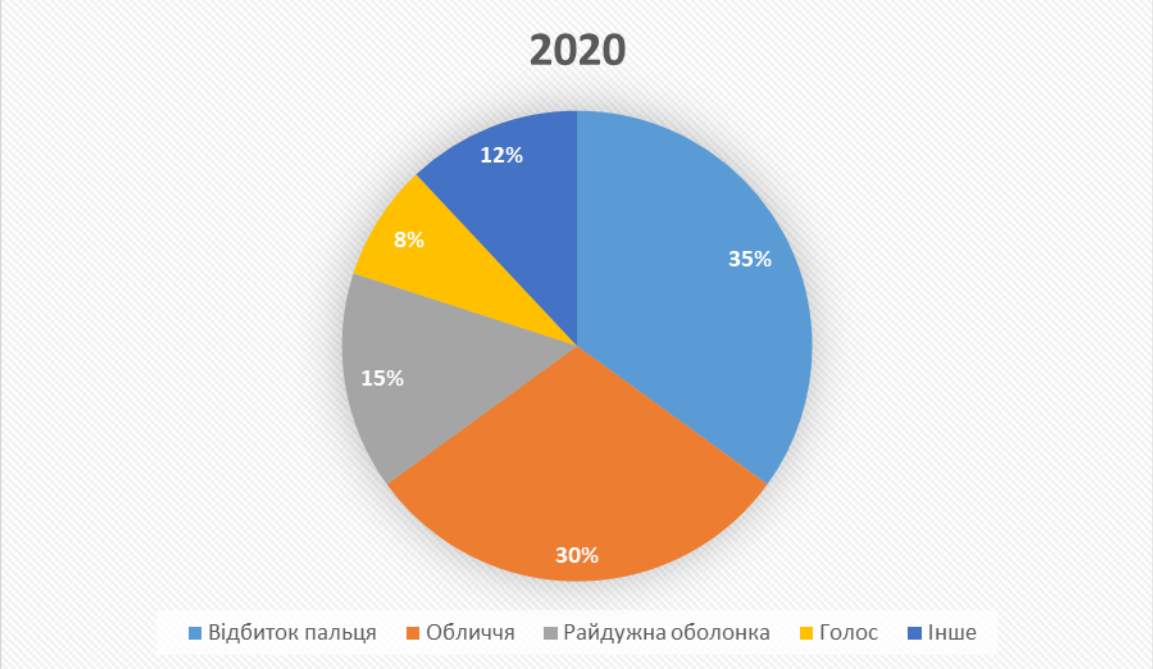

Рис. 2.2. Статистика використання методів біометричної автентифікації на 2020 рік

Основна перевага методів біометричної автентифікації перед іншими методами автентифікації користувача полягає в тому, що ці методи використ-

товують реальні фізіологічні або поведінкові характеристики людини для автентифікації користувачів. Ці біометричні характеристики є більш-менш постійними і незмінними. Також непросто, хоча в деяких випадках і принципово неможливо, змінити відбиток пальця, райдужну оболонку ока або інші біометричні характеристики.

Користувачі не можуть передавати свої біометричні характеристики іншим користувачам так само легко, як свої карти або паролі.

Біометричні об'єкти не можна вкрасти як жетони, ключі, карти або інші об'єкти, що використовуються для традиційної автентифікації користувачів, проте біометричні характеристики можуть бути вкрадені з комп'ютерних систем і мереж.

Біометричні характеристики не є секретом, і тому наявність відбитка пальця або малюнка райдужної оболонки очі не порушує безпеку так же, як доступність пароля користувача. Навіть використання мертвих або штучних біометричних характеристик не повинно пропустити зловмисника. [26]

Більшість біометричних методів засновано на тому, що не можна втратити або забути. Ця перевага як для користувачів, так і для системних адміністраторів, оскільки можна уникнути проблем і витрат, пов'язаних із загубленими, перевипущеними або тимчасово випущеними токенами / картами / паролями, що дозволяє заощадити деякі витрати на управління системою.

Ще однією перевагою систем біометричної автентифікації може бути їх швидкість. Автентифікація звичного користувача за допомогою системи ідентифікації на основі райдужної оболонки ока може зайняти 2 (або 3) секунди, а пошук зв'язки ключів, пошук потрібного ключа і його використання може зайняти близько 5 (або 10) секунд.

Але немає нічого ідеального, і методи біометричної автентифікації теж мають свої недоліки. По-перше, продуктивність біометричних систем не ідеальна. Біометричні системи все ще потребують поліпшення з точки зору точності і швидкості. Біометричні системи з рівнем помилкового відхилення

менше 1% (разом з досить низьким рівнем помилкового прийняття) сьогодні все ще рідкісні. Хоча мало біометричних систем є швидкими і точними (з точки зору низького рівня помилкового прийняття), достатніми для забезпечення ідентифікації (автоматичного розпізнавання особистості користувача), більшість сучасних систем підходять тільки для перевірки, оскільки рівень помилкового прийняття занадто високий.

Не всі користувачі можуть використовувати будь-яку біометричну систему. Люди без рук не можуть використовувати відбитки пальців або системи на основі рук. Людям з ослабленим зором важко використовувати методи на основі райдужної оболонки або сітківки. Оскільки не всі користувачі можуть використовувати конкретну біометричну систему, систему автентифікації необхідно розширити.

Біометричні дані не вважаються секретними, і безпеку біометричної системи не може ґрунтуватися на секретності біометричних характеристик користувача. Сервер не може автентифікувати користувача відразу після отримання його правильних біометричних характеристик. Автентифікація користувача може бути успішною тільки в тому випадку, якщо характеристики користувача є свіжими і отримані від автентифікуємого користувача. Це означає, що біометричному пристрою введення потрібно довіряти. Його справжність повинна бути перевірена (якщо пристрій і посилання не є фізично безпечними), і життєздатність користувача теж бути перевірена. Пристрій введення також має перебувати під наглядом людини або захищен від злому. Той факт, що біометричні характеристики не є секретними, викликає деякі проблеми, з якими традиційні системи автентифікації не повинні мати справу. Багато сучасних біометричних систем не знають про цей факт, і тому рівень безпеки, який вони пропонують, обмежений.

Деякі біометричні датчики (особливо ті, які контактують з користувачами) також мають обмежений термін служби. Хоча зчитувач магнітних карт може використовуватися роками (або навіть десятиліттями), оптичний при-

стрій для читання відбитків пальців (при інтенсивному використанні) необхідно регулярно чистити, і навіть в цьому випадку термін служби не повинен перевищувати одного року.

Біометричні системи можуть порушувати конфіденційність користувача. Біометричні характеристики - це конфіденційні дані, які можуть містити багато особистої інформації. ДНК містить, серед іншого, схильність користувача до захворювань. Це може бути дуже цікава інформація для страхової компанії. Запах тіла може дати інформацію щодо діяльності за останній час користувача.

Використання біометричних систем також може означати втрату анонімності. Хоча у одного може бути кілька ідентифікаторів, коли методи автентифікації засновані на тому, що користувач знає або має, біометричні системи іноді можуть пов'язувати всі дії користувача з одним ідентифікатором.

Біометричні системи потенційно можуть бути досить неприємними для деяких користувачів. Ці користувачі знаходять деякі біометричні системи нав'язливими. Навіть якщо ніяка біометрична система не є по-справжньому небезпечною, користувачі іноді бояться чогось, про що вони мало знають. У деяких країнах люди не люблять торкатися до чогось, до чого вже торкалися багато разів, наприклад, до біометричного датчика, що є дуже актуальним в період пандемії у 2020 році, в той час як в деяких країнах люди не люблять фотографуватися або їх особи повністю закриті.

Відсутність стандартів (або незнання стандартів) також може бути серйозною проблемою. Дві схожі біометричні системи від двох різних виробників навряд чи будуть взаємодіяти в даний час.

4. Автентифікація за допомогою цифрового сертифікату.

Цифровий сертифікат - це технологія шифрування, яка працює аналогічно інтернет-версії паспорта. Використовуючи інформацію по відкритому і закритому ключах, цифрові сертифікати по суті гарантують одержувачу повідомлення, що повідомлення виходить від конкретної людини. Цифровий

сертифікат засвідчує особу відправника, щоб забезпечити більш безпечний зв'язок і запобігти шахрайству в Інтернеті. Найбільшою перевагою автентифікації на основі цифрових сертифікатів є конфіденційність. За рахунок шифрування ваших повідомлень - електронної пошти, логінів або транзакцій онлайн-банкінгу - цифрові сертифікати захищають особисті дані і запобігають перегляд інформації сторонніми очима. Системи цифрових сертифікатів також зручні для користувача, зазвичай працюють автоматично і вимагають мінімальних дій або участі як від відправника, так і одержувача.

Завжди існує компроміс між підвищенням безпеки і пов'язаними з цим витратами і тягарем для кінцевих користувачів. Більшість людей не замислюється про це, але використовувати сертифікати дуже легко для кінцевих користувачів. Після установки сертифіката (а в деяких випадках це може відбуватися автоматично) більше нічого не потрібно робити. Крім того, більшість корпоративних рішень вже підтримують автентифікацію на основі сертифікатів. [27]

Також можна легко використовувати існуючі групові політики та дозволи, щоб контролювати, які користувачі і машини можуть отримати доступ до різних програм і мереж. Таким чином, можна гарантувати, що тільки привілейовані користувачі можуть отримати доступ до конфіденційних або важливих операцій.

Ще однією перевагою використання сертифікатів є можливість взаємної автентифікації, що означає, що обидві сторони, що беруть участь в обміні даними, ідентифікують себе, незалежно від того, чи йде цей обмін даними між користувачами, користувачем та машиною або машинами. Наприклад, клієнт повинен підтвердити свою особистість в інтрамережі компанії, а інтрамережа повинна підтвердити свою особистість для клієнта, перш ніж можна буде встановити з'єднання.

Хоча ідея цифрових сертифікатів полягає в тому, щоб заблокувати сторонніх від перехоплення повідомлень, система не є надійною. Наприклад, у

2011 році хакери зламали голландський центр цифрової сертифікації DigiNotar. Оскільки центри сертифікації несуть відповідальність за видачу цифрових сертифікатів, хакери часто націлені на ці органи, щоб маніпулювати інформацією сертифіката. В результаті, коли центр сертифікації буде дискредитовано, хакери можуть створювати веб-сайти або відправляти електронні листи, які виглядають справжніми і проходять сертифікаційні тести, але насправді є шахрайськими.

Центри цифрових сертифікатів постійно оновлюють своє програмне забезпечення, щоб звести до мінімуму подібні загрози безпеки, але загрози безпеки як і раніше викликають занепокоєння. У 2013 році Forbes відзначив, що цифрові сертифікати стали основною метою для хакерів і інших кіберзлочинців, враховуючи, що інформація, яку вони захищають, настільки цінна. Програмне забезпечення вимагає постійної пильності для захисту користувачів від кіберзлочинності.

2.2. Сучасні проблеми «сильної» автентифікації

В силу того, що дуже часто, але не завжди, під процесом «сильної» автентифікації розуміють багатофакторну автентифікацію, то варто розглянути проблеми останньої.

Основною проблемою багатофакторної автентифікації є проблема сумісності між пристроями, а саме стандартизація цих пристроїв (наприклад usb-токенів).

При реалізації системи багатофакторної автентифікації потрібно дотримуватися повної незалежності факторів. Також проблемою багатофакторної автентифікації є спосіб її реалізації.

Наприклад, якщо використовувати в якості одного з факторів фактор «я знаю», а саме класичний «багаторазовий» пароль, то це є серйозною уразливістю при роботі з чужих комп'ютерів, наприклад в інтернет-кафе. Це підштовхнуло провідних виробників ринку автентифікації до створення апаратних генераторів одноразових паролів. Такі пристрої генерують черговий пароль

або за розкладом (наприклад, кожні 30 секунд), або за запитом (при натисканні на кнопку). Кожен такий пароль можна використовувати тільки один раз.

Також проблема створення користувача в системі нерозривно пов'язана з тематикою пароля, що тягне за собою ряд проблем.

1. Бачення процедури автентифікації;

Під процедурою автентифікації мається на увазі метод, за яким ідентифікаційний номер передається користувачеві і яким чином сертифікати передаються користувачам. Процес автентифікації повинен бути детально і якісно проаналізовано на наявність вразливостей. Кращий шлях для вирішення цієї проблеми - видалення людського фактора або зведення його до мінімуму. Наприклад, банківські сервіси передають конфіденційну інформацію з використанням робота (генератора мови по тексту або смс), виключаючи оператора. [28]

2. «Місце» автентифікації;

Дуже часто алгоритм підтвердження автентичності користувача розташований на серверах недоступних службі безпеки організації. Ці сервера, можна назвати місцем автентифікації. Такі сервера мають бути детально обстежені. Сам користувач вводить пароль в веб-формі браузера або мобільного телефону, тому призначений для користувача інтерфейс або пристрій, з якого відбувається автентифікація, також можна назвати місцем автентифікації. В ідеалі, вони також повинні бути захищені. На практиці набагато простіше зламати веб-форму введення логіна і пароля у вікні браузера, ніж алгоритм підтвердження автентичності на сервері з сервісом або базою даних.

3. Захист персональних ідентифікаційних даних;

Дані беруть участь в ідентифікації повинні також бути конфіденційними і до них повинні застосовуватися такі ж засоби захисту, які застосовуються до зберігання паролів.

Наприклад, телефонний номер не може виступати паролем, так як є публічною інформацією. У реальних системах іноді зустрічаються такі уразливості, часто в корпоративних системах.

4. Втомленість користувача.

Обтяжуючи користувача додатковими заходами автентифікації, система ризикує втомити його. Як показало нове дослідження South West News Service [29], мільйони американців втомилися від надмірно складних заходів безпеки в Інтернеті і телефонах. Дослідники, які опитали 2000 дорослих американців, виявили, що 81 відсоток не бачать необхідності в тому, що вони вважають непотрібними процедурами безпеки. 48% втомилися від використання двоетапної автентифікації, а 7 з 10 (71%) незадоволені кодами капчі, оскільки вони, як правило, містять нерозбірливі слова.

Дослідження, проведене на замовлення компанії FICO, що займається розробленням програмного забезпечення, також показало, що понад дві третини (71%) вважають, що в даний час існує дуже багато заходів безпеки. Віце-президент по боротьбі з шахрайством в FICO зазначив, що споживачі раді, що банк захищає їх, але розчаровані тим, що такий захист ускладнює їм відкриття рахунків і здійснення покупок.

Необхідність запам'ятовувати адреси електронної пошти для відновлення паролів викликає роздратування у 58% - і точно так же 6 з 10 (65%) знаходять дратівливим, коли поштові системи випадковим чином виводять їх із системи в якості міри безпеки. Навіть 38% вважають, що введення PIN-коду для мобільного телефону - це деяка проблема.

Втомлений користувач може часто вводити неправильний пароль. Група таких користувачів може створити шум для системи моніторингу атак. Запобіжні заходи повинні бути гнучкими та адекватними, у міру можливостей підлаштовуватися під кожного користувача. Наприклад, в системах з автентифікацією за логіном, паролем і смс-оповіщення з одноразовим паролем, часто відбувається затримка смс через проблеми зі зв'язком, в результаті ко-

ристувач може кілька разів форматувати відправлення смс, що викличе підозру у системи моніторингу атак, яка блокує втомленого користувача. Такі похибки варто враховувати, наприклад, додавши до варіанту «вислати одноразовий пароль повторно» варіант «мені нічого не прийшло». Необхідний баланс між зручністю сервісу і його захищеністю.

Кількість ресурсів, які вимагають автентифікацію за допомогою пароля, безліч, а користувачі, в свою чергу, не в змозі зберігати в своїй пам'яті унікальні паролі для кожного з них і не дивлячись на всі рекомендації нехтують елементарними правилами. Додаткова кількість факторів автентифікації може привести до так званої «втоми користувача», в зв'язку з цим система потребує нового альтернативного методу автентифікації - автентифікація за допомогою Bluetooth-технологій.

2.3. Аналіз безпеки Bluetooth пристроїв

Bluetooth став популярним механізмом бездротового зв'язку, який широко використовується на багатьох електронних пристроях, таких як комп'ютери, клавіатури, миші, принтери та мобільні телефони. Оскільки смартфони використовуються повсюдно, все більша кількість обладнання і програмні додатки інтегруються в них. Наприклад, в смартфони вбудована камера, щоб вони могли робити фотографії, знімати фільми і проводити відеоконференції. Що стосується програмного забезпечення, програми дозволяють користувачам перевіряти електронну пошту, переглядати веб-сторінки і редагувати документи, а також передавати їх по бездротовій мережі. Такі додатки вимагають конфіденційності, цілісності і доступності даних для збереження конфіденційності користувачів. Однак через природи бездротових з'єднань транслюються повідомлення можуть бути легко перехоплені, і, таким чином, безпеку Bluetooth має вирішальне значення для забезпечення дотримання політик безпеки.

З моменту розробки Bluetooth було прийнято багато заходів безпеки, щоб забезпечити безпеку передачі інформації через Bluetooth. Багато при-

строїв використовують технологію Bluetooth, і тому існує безліч додатків, спеціально розроблених для використання Bluetooth в якості середовища передачі даних між пристроями.

Стандарт Bluetooth визначає п'ять основних принципів безпеки:

1. Автентифікація;

Мета Bluetooth - мати можливість перевіряти ідентичність взаємодіючих пристроїв з унікальною адресою Bluetooth. Bluetooth не забезпечує власну автентифікацію користувача.

2. Конфіденційність;

Іншою метою є запобігання розкриття інформації, викликаної підслуховуванням, шляхом забезпечення того, щоб тільки авторизовані пристрої могли отримувати доступ і переглядати передані дані;

3. Авторизація;

Стандарт Bluetooth дозволяє управління ресурсами, гарантуючи, що пристрою дозволено використовувати послугу першим.

4. Цілісність повідомлення;

Повідомлення, що передаються між двома пристроями Bluetooth, не повинні змінюватися при передачі.

5. Сполучення / зв'язування;

Остання мета стандарту Bluetooth - створити один або кілька загальних секретних ключів між двома пристроями і зберегти їх для майбутнього використання при наступних комунікаціях. [30]

В загальному профілі доступу для Bluetooth є три режими, в яких може працювати з'єднання Bluetooth:

1. Режим безпеки 1: Небезпечний;
2. Режим безпеки 2: Примусова безпека на рівні обслуговування;
3. Режим безпеки 3: Примусова безпека на рівні каналу.

На рівні незахищеного режиму процес автентифікації не буде ініційований до передачі інформації. Це означає, що немає ні шифрування, ні ключів, ні випадкових чисел. Для пристроїв, що працюють на цьому рівні, автентифікація необов'язкова.

Другий режим безпеки Bluetooth забезпечує помірну безпеку через певний канал або програму. Це забезпечує гнучкість для додатків, які можуть працювати одночасно на пристрої. Коли додатки або пристрої використовують більше одного режиму безпеки, за винятком небезпечного режиму, то другий режим безпеки є обов'язковим. На цьому рівні безпеки заходи безпеки не застосовуються до тих пір, поки не буде встановлено з'єднання або поки не буде ініційований процес встановлення з'єднання. Підключення цього типу класифікуються як авторизація. Зазвичай пристрої намагаються перевірити інші пристрої, які намагаються отримати доступ до служб, за допомогою персонального ідентифікаційного номера (далі – ПІН) Bluetooth.

Третій рівень безпеки Bluetooth - це посилена безпека канального рівня. Це найвищий рівень безпеки, який забезпечується підключенням Bluetooth. Як і другий режим безпеки, третій режим безпеки також є обов'язковим, якщо є більше одного іншого рівня безпеки, за винятком небезпечного режиму. Коли пристрій хоче з'єднатися з іншим пристроєм з підтримкою Bluetooth, воно відправляє запит на з'єднання по протоколу Link Manager (далі - LMP). Після цього другий пристрій встановить з'єднання з LMP Pairing, LMP Authentication і шифруванням даних, забезпечуючи прийняття всіх заходів безпеки перед відправкою будь-якої інформації. Якщо будь-яка з вимог для безпечного з'єднання не виконана, це призведе до збою автентифікації. Коли обидва пристрої відчують, що всі вимоги безпеки виконані, вони видають команду set up complete. [31]

Для ідентифікації окремих пристроїв кожному Bluetooth-пристрою видається унікальна адреса Bluetooth-пристроя. Адреси пристроїв Bluetooth мають довжину 48 біт і унікальні для кожного. Є кілька різних ключів, вико-

ристовуваних для встановлення і автентифікації з'єднання між пристроями. Приватний ключ автентифікації - це ключ, який використовується в процесі автентифікації. Довжина цього ключа становить 128 біт. Приватний ключ шифрування - це ключ, який використовується для шифрування, довжина якого може складати від 8 до 128 біт. Також, існує випадкове число, що виробляється окремими пристроями Bluetooth і має довжину 128 біт.

При використанні Bluetooth безпечні з'єднання і шифрування інформації покладаються на секретність ключів. Отже, управління ключами стає дуже важливою темою при використанні Bluetooth. Є один ключ, який обробляє всі захищені транзакції між двома або більше пристроями. Це називається ключем зв'язку і являє собою випадкове число довжиною 128 біт. Ключ зв’язку можна класифікувати як напівпостійний або тимчасовий. Ця класифікація визначає час життя ключа. Ключ зв’язку зберігається і може бути повторно використаний для автентифікації інших пристроїв Bluetooth, які використовують його, навіть після завершення транзакції. Ключ зв’язку використовується при автентифікації користувачів, а також для отримання ключа шифрування, який буде використовуватися для поточної транзакції.

Залежно від програми, що використовує Bluetooth, ключ зв'язку може бути різних типів. Існують комбіновані клавіші, ключі одиниць, майстер-ключі і ключі ініціалізації. Ключ пристрою - це ключ, що генерується пристроєм при його установці в мережі. Коли два пристрої в пару, вони будуть використовувати комбінацію клавіш. Цей ключ - це ключ, створений на основі інформації, взятої з кожного пристрою. Головний ключ - це тимчасовий ключ, який використовується пристроєм, коли він хоче транслювати інформацію більш ніж на один пристрій. Нарешті, ключ ініціалізації використовується, коли два пристрої повинні встановити з'єднання, але поки недоступні поодинокі або комбіновані клавіші.

Коли два пристрої намагаються з'єднатися один з одним, обидва пристрої запитують у користувача ПІН. Введений ПІН-код може складатися з 4-8

цифр, що дає ПІН від 1 до 16 октетів. ПІН-код також можна зафіксувати так, щоб його потрібно було вводити тільки на пристрої, який хоче підключитися. Користувач отримує запит на введення ПІН-коду, коли між двома пристроями немає іншого попереднього з'єднання. ПІН-код допоможе в генерації ключа ініціалізації. Ключ ініціалізації генерується з використанням алгоритму E22. Алгоритм E22 приймає випадкове число, що генерується пристроєм, ПІН-код і довжину ПІН-коду в якості вхідних даних, а отриманий результат є ключем ініціалізації. Після узгодження та генерації ключа зв'язку ключ ініціалізації відкидається.

У свою чергу, ключ об'єкта генерується за допомогою алгоритму E21. Цей алгоритм використовує випадкове число і адреса пристрою Bluetooth для виведення ключа пристрою. Після генерації ключа пристрою він буде збережений в незалежній пам'яті. Інші пристрої можуть використовувати цей ключ пристрою як ключ зв'язку між собою. Використовуваний ключ пристрою буде визначено в процесі ініціалізації. Комбінований ключ також генерується з використанням алгоритму E21. Однак після того, як обидва пристрої згенерували ключ, кожне з них поділиться своїми випадковими числами, щоб вони могли згенерувати ключ іншого пристрою і обчислити комбінований ключ, який буде використовуватися між ними.

Майстер-ключ - це тимчасовий ключ, також отриманий з використанням алгоритму E22. Входи - два 128-бітних випадкових числа. Потім на пристрій відправляється третій випадкове число. Алгоритм генерації ключа і поточний ключ зв'язку будуть використовувати це число, щоб допомогти як ведучому, так і веденого пристрою обчислити накладення. Майстер-ключ піддається XOR-операції з накладенням і відправляється відомому пристрою. Це дозволяє відомому пристрою обчислити головний ключ. Нарешті, ключ шифрування генерується алгоритмом E3. E3 приймає випадкове число, ключ зв'язку і 96-бітове число зсуву шифрування для виведення 128-бітного алго-

ритму шифрування. Номер зміщення шифрування генерується в процесі автентифікації.

Bluetooth шифрує інформацію в пакетах корисних даних. Шифрування виконується з використанням потокового шифру E0. Ключ потоку - XOR з корисними даними. Ключовий потік створюється з використанням криптографічного алгоритму, заснованого на чотирьох регістрах зсуву з лінійної зворотним зв'язком. Алгоритм E0 приймає головну адресу пристрою Bluetooth, випадкове число, номер слота і ключ шифрування. Номер слота змінюється з кожним відправленим пакетом, тому механізм шифрування необхідно повторно форматувати перед шифруванням кожного пакета.

Ключ шифрування може відрізнятися по довжині між пристроями Bluetooth, тому перед шифруванням трафіку необхідно узгодити загальну довжину ключа шифрування, перш ніж шифрування може статися. Майстер відправить пропозицію про довжину ключа шифрування, а підлеглий пристрій може прийняти або відхилити. Якщо відоме пристрій відхиляє пропозицію, воно робить нову пропозицію ведучого, і ведучий пристрій вирішує, чи прийме він його. Пропозиції будуть ходити туди і назад, поки не буде визначена загальна довжина ключа шифрування. Якщо не може бути угоди досягнуто, то шифрування буде недоступно.

Доступні різні режими шифрування в залежності від того, який ключ використовує додаток Bluetooth. Коли для ключа зв'язку використовується одиничний ключ або комбінований ключ, дані можуть бути зашифровані тільки при відправці одноадресного трафіку, проте це не обов'язково. При відправці широкомовного трафіка їх не можна зашифрувати. Це призводить до трьох режимів шифрування:

1. Режим шифрування 1: нічого не зашифровано;
2. Режим шифрування 2: шифрується тільки одноадресний трафік;

При використанні майстер-ключа доступний третій режим:

3. Режим шифрування 3: весь трафік (одноадресний і широкомовний) зашифрований.

Автентифікація виконується в системі Bluetooth з використанням симетричних ключів. Вся система заснована на припущенні, що тільки два пристрої, які хочуть зв'язуватися один з одним, мають доступ до ключа. На жаль, схема автентифікації Bluetooth дуже слабка, і зламати ПІН-код насправді дуже легко.

Автентифікація Bluetooth виконується в три основних етапи. Весь процес називається «процесом сполучення». Кожен крок включає в себе передачу даних і генерацію ключів.

Першим кроком автентифікації (див. Рис. 2.3) є генерація ключа ініціалізації. Коли два пристрої хочуть автентифікувати один одного, пристрій, що вважається провідним, відправляє відомому випадкове 128-бітове число. Потім ключ обчислюється з використанням випадкового числа (IN_RAND), адреси пристрою Bluetooth веденого пристрою (BD_ADDR) і введеного користувачем ПІН-коду. За допомогою алгоритму E22 обчислюється ключ ініціалізації, який використовується на наступному етапі автентифікації.

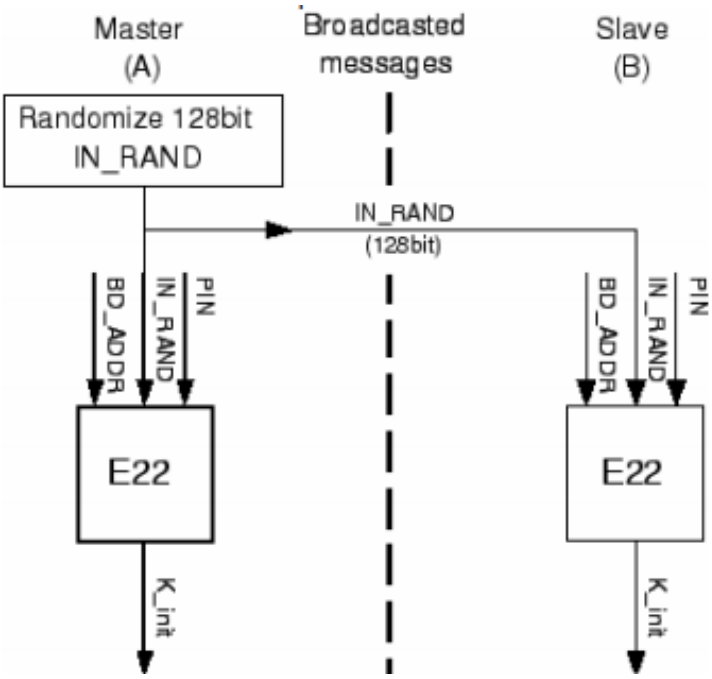

Рис 2.3. Перший крок Bluetooth автентифікації

Другий крок процесу (див. Рис. 2.4) починається з передачі іншого випадкового числа XOR з ключем ініціалізації, переданим від ведучого до веденого (LK_RANDA). Підлеглий пристрій передає аналогічний сигнал назад ведучому (LK_RANDb). Потім створюється одиничний ключ (K_{ab}) з використанням значень, отриманих з цих двох чисел XOR, BD_ADDR і алгоритму E21. Ключ Пристрій підтримує в пам'яті кожного пристрою, щоб використовуватися для майбутньої автентифікації. Нарешті, щоб переконатися, що обидва пристрої мають однаковий ключ пристрою (і, отже, ПІН), ведучий пристрій генерує інше випадкове число (AU_RAND) і відправляє його відомому пристрою. Потім обидва пристрої генерують значення (SRES), використовуючи ключ пристрою, адреса пристрою Bluetooth і випадкове число, використовуючи алгоритм E1. Потім підлеглий пристрій передає це значення ведучому, який перевіряє, чи відповідає це число того, яке було створене провідним. Якщо значення не збігаються, процес автентифікації зупиняється. Після певної кількості неприпустимих значень весь процес автентифікації буде перезапущений. Якщо значення відповідають один одному, то цей крок процесу автентифікації повторюється з перемиканням ролей ведучого і підлеглого. Автентифікація завершена, якщо другі значення SRES відповідають один одному.

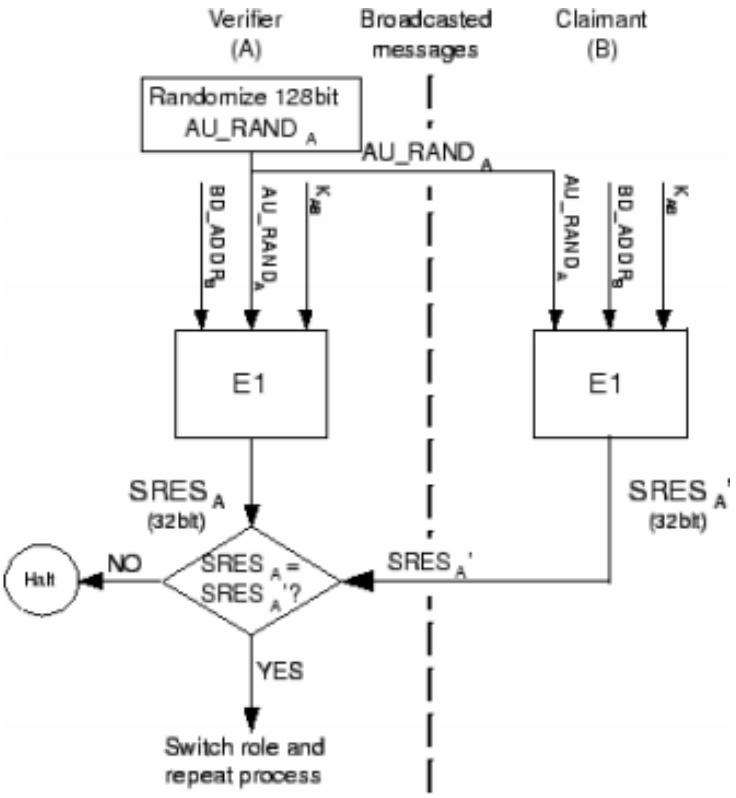

Рис 2.4. Другий крок Bluetooth автентифікації

У міру того, як Bluetooth стає все більш популярним і широко використовується, проблеми безпеки були виявлені і їх число зростає, особливо в смартфонах. Проблеми безпеки та вразливості безпеки Bluetooth викликали безліч загроз і атак, спрямованих на смартфони. Нижче представлений список сучасних проблем безпеки з Bluetooth:

1. Шифрування даних не є обов'язковим;

2. Іноді параметри конфігурації пристрою небезпечні. Наприклад, такі функції, як автентифікація і шифрування, можуть бути відключені або ПІН-код може бути встановлений на «0000». Також може бути складно змінити налаштування безпеки за замовчуванням;

3. Піни використовуються для генерації ключів шифрування і посилань між користувачами. Короткі і тривіальні контакти вважаються слабкими контактами - їх легко вгадати або зламати;

4. У великих мережах виникають труднощі з розподілом і установкою контактів. Крім того, з ПІН-кодом виникають типові проблеми як з па-

ролем, такі як: паролі записуються, змінюються нечасто, забуваються або передаються;

5. Ключ модуля використовується повторно: після того, як ключ пристрою згенерований, один і той же ключ використовується для кожного з'єднання між двома пристроями, що робить з'єднання небезпечними. Таким чином, якщо зловмисник успішно отримує ключ пристрою, він може відслідковувати весь трафік між двома пристроями;

6. В даний час Bluetooth не має спеціального механізму для генерації випадкових чисел. Тому виробникам необхідно мати власний спосіб генерації випадкових чисел. Таким чином, якість таких номерів може відрізнятися у різних виробників;

7. В даний час надається тільки автентифікація пристрою, Bluetooth не забезпечує ніякої форми автентифікації користувача;

8. Сполучення рекомендується проводити в приватному місці. Публічне з'єднання вразливе для атаки злому ПІН-коду;

9. Сквозна (end-to-end) безпека не передбачена. Тільки окремі посилання зашифровані і автентифіковані;

10. Аудит, невідказність та інші послуги не надано.

Були виявлені різні види атак, які націлені на уразливості Bluetooth смартфонів і призводять до втрати конфіденційності, цілісності і доступності ресурсів:

1. Bluesnarfing;

Ця атака дозволяє непомітно підключитися до пристрою без попередження власника пристрою і необхідності отримання дозволу від власника. Потім зловмисники можуть отримати доступ до будь-яких збережених даних і навіть до обмеженої області пам'яті, включаючи телефонну книгу, зображення, налаштування, повідомлення, історію викликів і серійні номери телефонів. Зловмисники зазвичай використовують атаку Bluesnarfing для копіювання вмісту телефону, що може привести до «клонування телефону». Атака

також може використовуватися для телефонних дзвінків і відправлення текстових повідомлень без згоди користувача. Зазвичай це можливо тільки тоді, коли пристрій переключено в «видимий» або «видимий» режим з «невидимого» режиму, коли включений Bluetooth. Однак недавно було виявлено, що інструменти, які дозволяють атакувати навіть в «невидимому» режимі доступні в Інтернеті.

За допомогою програмного забезпечення процедура Bluesnarfing стає простіше. Програмне забезпечення, яке використовується для процедури, повинно бути запущено на портативному комп'ютері на ранніх стадіях розробки, але для цього потрібно, щоб атака проводилася на невеликій відстані від цільового пристрою, щоб ставило атаку під великий ризик бути поміченою. Сьогодні ця атака не викликає підозр, оскільки програмне забезпечення Bluesnarfing, написане на Java, може працювати на будь-якому мобільному телефоні J2MEenabled.

Bluesnarfing працює через механізм обміну об'єктами, що використовує протокол OBEX. OBEX використовується для обміну всілякими об'єктами, такими як файли, зображення, записи календаря та ін. Програма Bluesnarfing намагається підключитися до цільового пристрою Bluetooth через профіль OBEX Push Bluetooth. Але він використовує функцію «pull» замість функції «push» для отримання будь-яких збережених даних на пристрої. Профіль Bluetooth OBEX Push - серйозна уразливість технології Bluetooth, яка була реалізована в більш ранніх мобільних телефонах Bluetooth. OBEX Push може виконуватися без автентифікації за допомогою пристрою. Це тому, що спочатку він був розроблений для відправлення повідомлення не шкідливих даних, наприклад візитки. Інженери мобільних телефонів порахували непотрібним автентифікацію для таких простих обмінів, і тому реалізація обміну візитними картками обійшла механізми безпеки Bluetooth. Тепер доступні оновлення прошивки для вирішення проблеми.

2. Бекдор атака;

Ця атака також пов'язана з незаконним підключенням, яке призведе до розкриття особистих даних, з тією лише різницею, що необхідно встановити довірчі відносини через процес «сполучення» Bluetooth. Потім атакуючий пристрій віддаляється зі списку пар після успішного підключення, щоб гарантувати, що воно більше не буде записано в регістрі парного пристрою. Оскільки власник пристрою навряд чи буде стежити за списком пар весь час або в той момент, коли встановлюється з'єднання, йому важко помітити з'єднання, якому не довіряє. Потім зловмисник може продовжити безперешкодний доступ до телефону з привілеєм довірчих відносин. Додаткові привілеї включають доступ до Інтернету, шлюзу WAP або GPRS, крім даних, що зберігаються в телефоні.

3. Bluejacking;

Bluejacking сильно відрізняється від інших атак. Мета Bluejacking - дозволити зловмиснику відправляти анонімні повідомлення на пристрій замість збору даних за допомогою функції «push» OBEX, як згадувалося раніше, оскільки автентифікація не вимагається. Повідомлення, що містить ім'я пристрою, буде відображатися на цільовому пристрої під час процесу ініціалізації сполучення Bluetooth. Bluejacking часто нешкідливий і не пов'язаний з доступом до даних, їх видаленням або зміною. Атака зазвичай виконується як жарт, щоб викликати реакцію деяких користувачів зміною назви пристрою. Проте, Bluejacking також може бути використаний для зловмисних дій. Цільовий користувач легко може попастися на прийом і дозволити зловмиснику отримати доступ до вашого пристро.

4. Bluebugging;

Ця атака створює послідовне з'єднання з телефоном жертви, яке дозволяє зловмисникові отримати доступ до телефонної команди за допомогою технології Bluetooth. Процедура атаки зазвичай починається з пуша OBEX, як і в випадку з Bluesnarfing і Bluejacking, які не вимагають автентифікації або введення ПІН-коду. Потім через недоліки в реалізації Bluetooth в телефоні

Bluebugger може перервати процес відправки, і телефон Bluebugger залишиться в списку телефону жертви як «довірений» пристрій. Bluebugger може увійти в Ascii Terminal (B) команди для управління телефоном. Ascii Terminal-команди дуже поширені для настройки та управління телекомунікаційними пристроями. Потім зловмисникам буде дозволено здійснювати телефонні дзвінки, відправляти та отримувати повідомлення, отримувати доступ і редагувати телефонну книгу і налаштування телефону, а також підключатися до Інтернету (аналогічно атакам Bluesnarf і бекдор). Крім того, атака Bluebug дозволяє зловмисникові підслуховувати телефонні розмови. Було виявлено, що можна навіть відслідковувати інші телефонні розмови поблизу, якщо цільовий (атакований) пристрій знаходиться в мережі GSM.

5. Хробак Cabir

Мобільний телефон з Bluetooth також вразливий для хробаків і вірусів, як і комп'ютери. Хробак Cabir - це шкідливий програмний засіб, яке класифікується як самовідтворний хробак. Він намагається пов'язати підключений пристрій Bluetooth з іншим цільовим пристроєм. При успішному сполученні він встановлюється на цільовий пристрій і повторює цей процес для інших подібних вразливих пристроїв. Зворотною стороною хробака Cabir є те, що він розряджає акумулятор пристрою щоразу, коли шукає інші Bluetooth-пристрої, що робить його присутність в деякій мірі виявляємою.

6. Атаки відмови в обслуговуванні (далі – DoS).

DoS-атаки також можливі на мобільних телефонах Bluetooth. Вони працюють так само, як і традиційні атаки DoS; зловмисники просто продовжують посилати недійсні запити на Bluetooth-сумісний пристрій і займати Bluetooth-канал пристрою. Ці запити вважаються неприпустимими повідомленнями Bluetooth OBEX. DoS-атака не тільки розряджає акумулятор пристрою, але і зберігає канал Bluetooth пристрої зайнятим, відключаючи зв'язок з іншими пристроями Bluetooth. Ця атака сильно впливає на доступність мереж Bluetooth.

Атаки Bluesnarfing, Бекдор і Bluebugging є результатом помилок реалізації виробниками мобільних телефонів. Була проведена робота по виправленню несправних моделей і забезпечення того, щоб майбутні продукти не постраждали від такої ж уразливості. Більшість цих атак малоймовірно, коли Bluetooth на мобільному телефоні відключені. Інших слабких місць мобільного телефону Bluetooth, таких як Bluejacking і проблеми з хробаками, можна уникнути, якщо використовувати пристрій обережно, тому що ці атаки вимагають від користувача прийняття процесу сполучення. Хоча атак можна уникнути, проблема простого злому ПІН-коду все ще залишається. Підвищення безпеки Bluetooth має бути пріоритетом. Поточний рівень безпеки не краще, ніж WEP-шифрування для бездротової мережі. Оскільки у Bluetooth і бездротової мережі частково збігаються проблеми, можна використати поліпшення, внесені WPA і WPA2 в шифрування WEP, щоб поліпшити безпеку Bluetooth. Основна проблема з безпекою Bluetooth - простота злому ПІН-коду. Отже, необхідно збільшити довжину ПІН. На додаток до цього, не повинно бути можливості отримати ПІН-код в процесі автентифікації. Рішенням цієї проблеми було б створення тимчасових ключів, отриманих з ПІН-коду, так щоб ПІН-код безпосередньо не використовувався.

Автоматична зміна ключів також має бути реалізовано таким чином, щоб навіть якщо ключ був скомпрометований, то не міг використовуватися для розшифрування всього обміну даних між двома пристроями. Це було реалізовано в WPA2 і довело свою ефективність в захисті головного ключа (в даному випадку ПІН-коду) від атак. Довжина ПІН-коду повинна бути збільшена таким чином, щоб до того часу, коли ключ, отриманий з ПІН-коду, може бути скомпрометований, в процесі зміни ключів вже мав бути створений новий ключ для майбутніх комунікацій. Це завадить зловмисникові успішно зібрати будь-які дані. Ще одна проблема, яку необхідно вирішити, - це відмовостійкі налаштування за замовчуванням. Коли два пристрої сполучені, то отримують повний доступ до ресурсів один одного. Таким чином, коли без-

пека скомпрометована, всі пристрої в її мережі також скомпрометовані. Безпека Bluetooth повинна мати контроль доступу, який обмежує потенційні небезпеки в результаті злому.

2.4. Висновки до розділу

Всі існуючі методи автентифікації не тільки не гарантують повну безпеку, а ще і приводять до «втомленості» користувача. Як показали дослідження, люди втомилися від надмірно складних та додаткових заходів безпеки. Користувачі не в змозі тримати в пам’яті паролі для всіх ресурсів, в яких вони заводили обліковий запис, тому нехтують елементарними правилами та рекомендаціями, такими як створювати унікальні паролі для кожного облікового запису та користуватися генераторами складних паролів, але не використовувати шаблони паролів або загальну інформацію про себе. Через це метод автентифікації за допомогою паролю є самим вразливим.

Біометричні методи не підходять всім користувачам, наприклад людям с обмеженими можливостями. Біометричні дані розкривають конфіденційність людини, наприклад інформація ДНК може бути використана страховою компанією у корисливих цілях. Також в Україні немає нормативно-правової бази для регулювання та відстеження правильності вилучення та зберігання біометричних даних. Та особливо актуальним в період пандемії в 2020 році є негігієнічність цього методу, наприклад використання одного біометричного датчику.

В зв’язку з цим постає питання використання нового методу – автентифікація за допомогою Bluetooth-технології.

Розділ 3. УДОСКОНАЛЕННЯ МЕТОДУ АВТЕНТИФІКАЦІЇ ЗА ДОПОМОГОЮ BLUETOOTH ТЕХНОЛОГІЇ

3.1 Вирішення проблем «сильної» автентифікації за допомогою Bluetooth пристроїв

В силу того, що дуже часто, під процесом «сильної» аутентифікації розуміють багатофакторну аутентифікацію, варто розглянути багатофакторну автентифікацію як приклад найпопулянішої «сильної» автентифікації.

Основною проблемою багатофакторна аутентифікації є проблема сумісності між пристроями, а саме стандартизація цих пристроїв (наприклад usb-токенів).

При реалізації системи багатофакторної аутентифікації потрібно дотримуватися повну незалежність факторів. Також проблемою багатофакторної аутентифікації є спосіб її реалізації.

Наприклад, якщо використовувати в якості одного з факторів фактор «я знаю», а саме класичний «багаторазовий» пароль, то це є серйозною уразливістю при роботі з чужих комп'ютерів, наприклад в інтернет-кафе. Це підштовхнуло провідних виробників ринку аутентифікації до створення апаратних генераторів одноразових паролів. Такі пристрої генерують черговий пароль або за розкладом (наприклад, кожні 30 секунд), або за запитом (при натисканні на кнопку).

Кожен такий пароль можна використовувати тільки один раз.

Також проблема створення користувача в системі нерозривно пов'язана з тематикою пароля, що тягне за собою ряд проблем.

Під процедурою аутентифікації мається на увазі метод, за яким ідентифікаційний номер передається користувачеві і яким чином сертифікати пере-

даються користувачам. Процес аутентифікації повинен бути детально і якісно проаналізовано на наявність вразливостей. Кращий шлях для вирішення цієї проблеми - видалення людського фактора або зведення його до мінімуму. Наприклад, банківські сервіси передають конфіденційну інформацію з використанням

робота (генератора мови по тексту або смс), виключаючи оператора.

Дуже часто алгоритм підтвердження автентичності користувача розташований на серверах недоступних службі безпеки організації. Ці сервера, можна

назвати місцем аутентифікації. Такі сервера мають бути детально обстежені.

Сам користувач вводить пароль в web-формі браузера або мобільного телефону, тому призначений для користувача інтерфейс або пристрій, з якого відбувається аутентифікація, також можна назвати місцем аутентифікації. В ідеалі, вони також повинні бути захищені. На практиці набагато простіше зламати web-форму введення логіна і пароля у вікні браузера, ніж алгоритм підтвердження автентичності на сервері з сервісом або базою даних.

Обтяжуючи користувача додатковими заходами аутентифікації, система ризикує втомити його. Втомлений користувач може часто вводити неправильний пароль. Група таких користувачів може створити шум для системи моніторингу атак. Запобіжні заходи повинні бути гнучкими та адекватними, у міру можливостей підлаштовуватися під кожного користувача.

Наприклад, в системах з аутентифікацією за логіном, паролю і СМС-оповіщення з одноразовим паролем, часто відбувається затримка СМС через проблеми зі зв'язком, в результаті користувач може кілька разів форматувати відправлення СМС, що викличе підозру в системі моніторингу атак, яка блокує втомленого користувача.

Такі похибки варто враховувати, наприклад, додавши до варіанту «вислати одноразовий пароль повторно» варіант «мені нічого не прийшло». Необхідний баланс між зручністю сервісу і його захищеністю.

Для вирішення описаних вище проблем пропонується використовувати Bluetooth пристрої в якості апаратного ключа для автентифікації.

Для впровадження даного методу, потрібно визначити які саме параметри Bluetooth пристрою будуть використані в ролі ключа, для гарантування надійності даної перевірки. Цей параметр, або набір параметрів, повинен бути унікальним для кожного пристрою, адже це є першою вимогою для використання пристрою задля автентифікації.

Кожен Bluetooth пристрій має унікальну адресу: Bluetooth Device Address (BD_ADDR). Адреса пристрою Bluetooth (або BD_ADDR) - це унікальний 48-розрядний ідентифікатор, присвоєний кожному пристрою Bluetooth виробником. Адреса Bluetooth зазвичай відображається як 6 байтів, записаних шістнадцятковою системою та розділених двокрапками (приклад - 00: 11: 22: 33: FF: EE). Верхня половина адреси Bluetooth (найбільш значущі 24 біти) називається так званим організаційно-унікальним ідентифікатором (OUI). З його допомогою можна визначити виробника пристрою. Префікси OUI призначаються Інститутом інженерів електротехніки та електроніки (IEEE). Додатково до ідентифікації, адреса пристрою Bluetooth використовується для визначення схеми стрибків частоти в радіозв'язку між пристроями Bluetooth.

Адреса Bluetooth складається з трьох частин: NAP, UAP та LAP (Рис. 3.1).

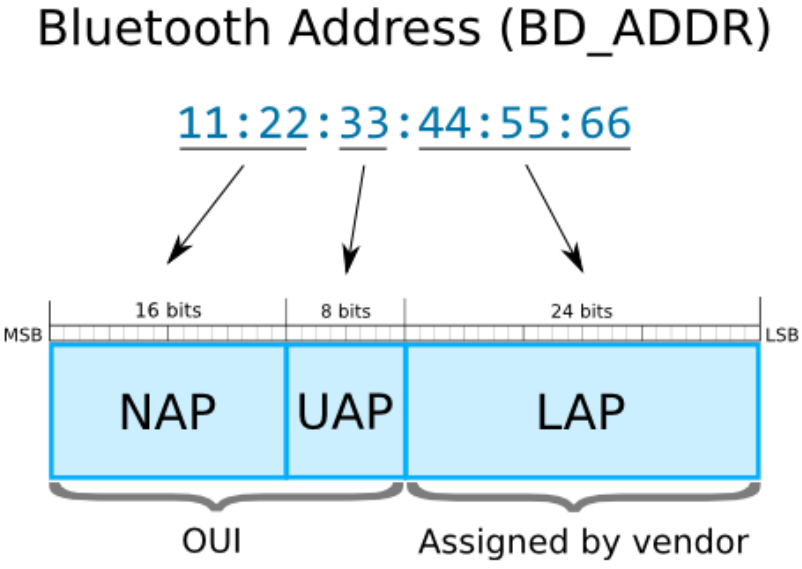

Рис.3.1 Структура адрес Bluetooth (NAP, UAP, LAP, OUI)

NAP: Незначна частина адреси (2 байти). Містить перші 16 бітів OUI. Значення NAP використовується в кадрах синхронізації перескоку частоти.

UAP: Частина верхньої адреси (1 байт). Містить 8 біт OUI, що залишилися. Значення UAP використовується в різних алгоритмах специфікації Bluetooth.

LAP: Частина нижньої адреси (3 байти). Цю частину адреси Bluetooth виділяє постачальник пристрою. Значення LAP однозначно ідентифікує пристрій Bluetooth як частину коду доступу у кожному переданому кадрі.

LAP та UAP складають значущу адресну частину (SAP) адреси Bluetooth.[33]

Опираючись на це можна зробити висновок, що адреса пристрою Bluetooth ідеально підходить для використання задля автентифікації користувача.

Проблемою, що може виникнути при даному підході, є діапазон дії Bluetooth сигналу. В сучасних версіях діапазон сигналу Bluetooth може сягати 100 метрів, що не задовольняє вимоги безпеки автентифікації. Цю проблему можна вирішити програмно, обмеживши «безпечну» відстань до Bluetooth розрахувавши її опираючись на рівень сигналу між пристроями.

Оскільки на даний момент, для пристроїв, що функціонують по стандартам Wi-Fi і Bluetooth 4.0, RSSI (Received Signal Strength Indicator) є єдиним параметром, що дозволяє виміряти відстань від пристрою до базової станції або маяка, цей показник і буде використано для вирішення проблеми.

В якості базового алгоритму позиціонування використано Frequency Hopping Spread Spectrum. Відповідно до цього алгоритму (рис. 3.2), в Bluetooth несуча частота сигналу стрибкоподібно змінюється 1600 разів в секунду (всього виділяється 79 робочих частот шириною в 1 МГц).

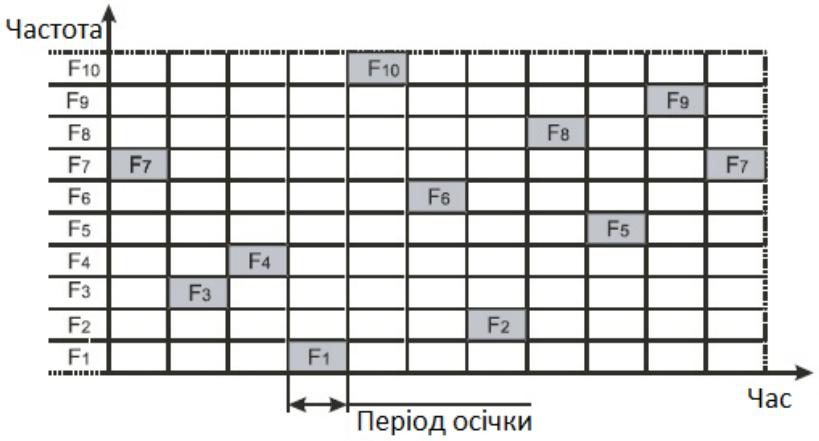

Рис. 3.2. Принцип дії алгоритму Frequency Hopping Spread Spectrum [34]

З метою виявлення місцерозташування в закритому просторі, пропонується метод, який передбачає декілька точок доступу у статичному положенні і на відомій додатку відстані (рис. 3.3).

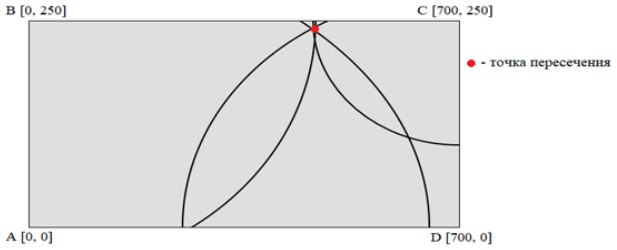

Рис. 3.3. Виявлення місцерозташування у замкнутому просторі

Ідея методу полягає в наступному. Знаючи відстань від точок, створюємо уявний прямокутник. Після цього скануємо точки на потужність сигналу і зберігаємо отримані результати у базу даних. Дізнавшись відстань від кожної точки, стає можливим знайти місцеположення телефону в уявному прямокутнику. Зібравши достатню кількість даних, можна оцінити місцеположення по усередненому сигналу і перевести його у відстань. Для переведення RSSI в одиниці вимірювання довжини зазвичай використовується наступна формула (рис. 3.4):

$$P_d = P_0 - 10 \cdot n \cdot \lg\left(\frac{d}{d_0}\right),$$

Рис. 3.4. Переведення RSSI в одиниці вимірювання довжини

де P_0 - потужність сигналу пристрою, виміряна на одиничній відстані, n – коефіцієнт втрат потужності сигналу при розповсюджені в середовищі (для повітря n = 2; збільшується за наявності перешкод), d - відстань від пристрою до смартфона, d_0 - відстань від пристрою до точки, на якій виконувалось вимірювання потужності сигналу.

Дане рівняння випливає з формули передачі Фріїса для поширення радіосигналу в вільному просторі, але є недостовірним для закритих приміщень [35].

У зв’язку з вищезазначеним, для підвищення точності позиціонування в роботі пропонується використання фільтра Калмана для послідовності вимірювань. Це дозволить знизити шуми та отримати оцінки невідомих змінних, що є потенційно більш точними за базові на самих лише вимірюваннях.

Всі досліди були проведені з маяками, які знаходилися в полі зору, так як маяки, які знаходяться за стінами будуть скриті шумом від цих об’єктів, а для таких розрахунків потрібно використовувати інші моделі та фільтри.

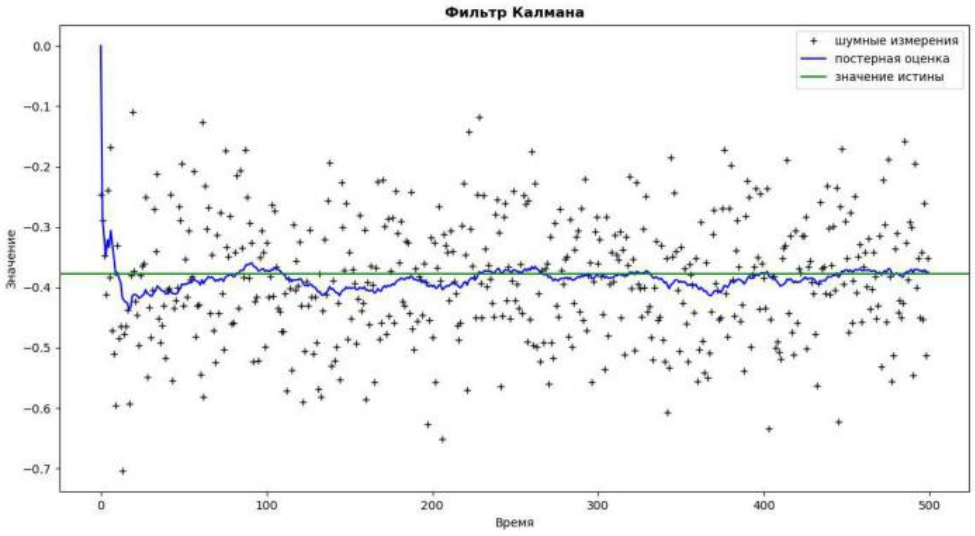

Рис. 3.5. Результат тестування фільтра Калмана на зашумлення

Було проведено кілька досліджень (рис. 3.5), завдяки яким можна було вивчити точність даних, що виводяться і правильності налаштування системи

перетворення сигналу в відстань. Повний пакет вихідних даних на код програми надано у [36]. Точність методу помітно падає, якщо один з пристроїв швидко рухається. Це пов'язано з тим, що між першим і останнім виміром проходить приблизно 3 секунди.

3.2 Опис технічних засобів

Для досягнення поставленої цілі запропонований метод автентифікації буде впроваджено в програмний продукт. Продукт буде сумісний з ОС Windows як однієї з найпопулярніших ОС у світі[37].

Для реалізації продукту на ОС Windows використанні наступні засоби:

1) мова програмування – C#;
2) програмна платформа .NET Framework;
3) Універсальна платформа Windows.

3.2.1 Мова C#

C # (вимовляється як "C Sharp") - проста, сучасна, об'єктно-орієнтована мова програмування. C # належить до відомого сімейства мов C, і здається знайомим кожному, хто працював з C, C ++, Java або JavaScript.

C # є об'єктно-орієнтованою мовою, але вона також підтримує компонентно-орієнтоване програмування. Розвиток сучасних додатків має тенденцію до створення програмних компонентів у вигляді автономних пакетів, що реалізують певні функції. Важливою особливістю таких компонентів є модель програмування на основі властивостей, методів та подій. Кожен компонент має атрибути, що забезпечують декларативну інформацію про компонент, а також вбудовані елементи документації. C # надає лінгвістичні конструкції, які безпосередньо підтримують цю концепцію роботи. Це робить C # чудовим для створення та використання програмних компонентів.

Ось лише декілька функцій мови C #, які забезпечують надійність та стійкість програми: збір сміття автоматично звільняє пам’ять для пошкодже-

них та невикористаних об’єктів; обробка винятків забезпечує структурований та розширюваний спосіб виявлення та обробки помилок; строге введення мови не дозволяє їй вдаватися до неініціалізованих змін, виходити за межі індексованих масивів або неконтрольованого кастингу типів.

У C # існує одна тип системи. Усі типи C #, включаючи примітивні типи, такі як int і double, успадковуються від одного кореневого типу об'єкта. Таким чином, усі типи використовують загальний набір операцій, і значення будь-якого типу можна зберігати, передавати та обробляти подібним чином. Крім того, C # підтримує спеціальні типи посилань та типи значень, дозволяючи як динамічне розподіл пам'яті для об'єктів, так і зберігання спрощених структур у стеці. [38]

У C # основними поняттями організаційної структури є програми, простори імен, типи, члени та збірки. Програма C # складається з одного або декількох файлів. Програма оголошує типи, що містять членів. Ви можете організувати ці типи в простори імен. Прикладами ti-pol є класи та інтерфейси. Учасники включають поля, методи, властивості та події. Коли програми компілюються на C #, вони упаковуються у збірки. Компіляція - це файл, зазвичай із розширенням .exe або .dll, якщо він реалізує програму або бібліотеку відповідно. [39]

У C # є два типи: посилальні типи та типи значень. Змінні типу значення містять дані безпосередньо, а змінні типу посилання зберігають посилання на запитувані дані, які називаються об’єктами. Дві змінні посилального типу можуть посилатися на один і той же об'єкт, тому можуть бути випадки, коли операції з однією змінною впливають на об'єкт, на який посилається інша змінна. Кожна змінна типу значення має власну копію даних, і операції з однією змінною не можуть впливати на іншу (крім ref та out).

Типи значень у C # поділяються на прості типи, перераховані типи, типи структур та типи, що допускають нульові значення. Типи посилань у C # поділяються на типи класів, типи інтерфейсів, типи масивів та типи делега-

тів.

Далі наводиться загальна інформація про систему типів на C #.

1) типи значень;
   - прості типи;
     - цілочисельний зі знаком: sbyte, short, int,long;
     - цілочисельний без знака: byte, ushort, uint,ulong;
     - символи Юнікоду: char;
     - десяткове значення з плаваючою комою: float,double;
     - десяткове значення з підвищеною точністю: decimal;
     - логічне значення: bool;
   - типи перерахування;
     - користувацькі типи в форматі enum E {...};
   - типи структур;
     - користувацькі типи в форматі struct S {...};
   - типи значень, що допускають значення NULL;
     - розширення інших типів значень, що допускають значення null;
2) посилальні типи;
   - типи класів;
     - вихідний базовим класом для всіх типів: object;
     - рядки Юникода: string;
     - користувацькі типи в форматі class C {...};
   - типи інтерфейсу;
     - користувацькі типи в форматі interface I {...};
   - типи масивів;
     - одно- і багатовимірні, наприклад, int[] і int[,];
   - тип делегатів;
     - користувацькі типи в форматі delegate int D(...).[40]

Класи є основним типом в мові C#. Клас являє собою структуру даних,

яка об'єднує в собі значення (поля) і дії (методи і інші функції-члени). Клас надає визначення для динамічно створюваних екземплярів класу, які також іменуються об'єктами. Класи підтримують механізми успадкування та поліморфізму, які дозволяють створювати похідні класи, що розширюють і уточнюють визначення базових класів.

Нові класи створюються за допомогою оголошень класів. Оголошення класу починається з заголовка, в якому вказані атрибути і модифікатори класу, ім'я класу, базовий клас (якщо є) і інтерфейси, реалізовані цим класом. За заголовком між роздільниками { і } слідує тіло класу, в якому послідовно оголошуються всі члени класу.

На рис 3.6 зображено оголошення класу з ім'ям Point.

```
public class Point
{
    public int x, y;
    public Point(int x, int y)
    {
        this.x = x;
        this.y = y;
    }
}
```

Рис. 3.6 Приклад оголошення класу

Екземпляри класів створюються за допомогою оператора new, який виділяє пам'ять для нового екземпляра, викликає конструктор для ініціалізації цього екземпляру і повертає посилання на екземпляр (рис. 3.7).

```
Point p1 = new Point(0, 0);
Point p2 = new Point(10, 20);
```

Рис. 3.7 Приклад створення об’єкту

Члени класу можуть бути статичними членами або членами екземпляру. Статичні члени належать класу в цілому, а члени екземпляра належать конкретним об'єктам (екземплярам класів).

Види членів, які можуть міститися в класі.

константи - константні значення, пов'язані з класом;

1) поля - змінні класу;

2) методи - обчислення і дії, які може виконувати клас;

3) властивості - дії, пов'язані з читанням і записом іменованих властивостей класу;

4) індексатори - дії, які реалізують індексування екземплярів класу, щоб звертатися до них як до масиву;

5) події - повідомлення, які можуть бути створені цим класом;

6) оператори - підтримувані класом оператори перетвореннь і виразів;

7) конструктори - дії, необхідні для ініціалізації екземплярів класу або класу в цілому;

8) дестроктори - дії, що виконуються перед остаточним видаленням екземплярів класу;

9) типи - вкладені типи, оголошені в класі.

Кожен член класу має певний рівень доступу. Він визначає, з якої області програми можна звертатися до такого члена.

Існує шість рівнів доступу:

1) public - доступ не обмежений;

2) protected - доступ можливий з цього класу і з класів, успадкованих від нього;

3) internal - доступ обмежений тільки поточною збіркою (.exe, .dll);

4) protected internal - доступ обмежений зовнішнім класом, класами, які є похідними від зовнішкього класу, або класами в тій же збірці;

5) private - доступ можливий тільки з цього класу;

6) private protected - доступ обмежений зовнішнім класом або класами, які є похідними від нього в тій же збірці.[41]

3.2.2 Програмна платформа .NET Framework

.NET Framework - програмна платформа, випущена компанією Microsoft в 2002 році. Основою платформи є загальномовне середовище ви-

конання Common Language Runtime (CLR), яка підходить для різних мов програмування. Функціональні можливості CLR доступні в будь-яких мовах програмування, що використовують це середовище[42].

При розробці платформи .NET Framework враховувалися наступні цілі.

1. Забезпечення узгодженої об'єктно-орієнтованого середовища програмування для локального збереження і виконання об'єктного коду, для локального виконання коду, розподіленого в Інтернеті, або для віддаленого виконання.

2. Надання середовища виконання коду, в якій:

2.1 зведена до мінімуму ймовірність конфліктів в процесі розгортання програмного забезпечення і управління його версіями;

2.2 гарантується безпечне виконання коду, включаючи код, створений невідомим або не повністю довіреною стороннього постачальника;

2.3 виключаються проблеми з продуктивністю середовищ виконання скриптів або інтерпретується коду;

3. забезпечуються єдині принципи розробки для різних типів додатків, таких як додатки Windows і веб-додатки;

4. забезпечується взаємодія на основі промислових стандартів, яке гарантує інтеграцію коду платформи .NET Framework з будь-яким іншим кодом.[42]

Платформа .NET Framework складається з загальномовного середовища виконання (середовища CLR) і бібліотеки класів .NET Framework. Основою платформи .NET Framework є середовище CLR. Середовище виконання можна вважати агентом, який керує кодом під час виконання і надає основні служби, такі як управління пам'яттю, управління потоками і віддалена взаємодія. При цьому середовищем накладаються умови суворої типізації та інші види перевірки точності коду, що забезпечують безпеку і надійність. Фактично основним завданням середовища виконання є управління кодом. Код, який

звертається до середовища виконання, називають керованим кодом, а код, який не обертається до середовища виконання, називають некерованим кодом. Бібліотека класів є комплексною об'єктно-орієнтованої колекцією повторно використовуваних типів, які застосовуються для розробки додатків - починаючи з звичайних додатків, що запускаються з командного рядка, і додатків з графічним інтерфейсом (GUI) і закінчуючи додатками, що використовують останні технологічні можливості ASP.NET, такі як веб-форми і веб-служби XML.[42]

Платформа .NET Framework може розміщуватися некерованими компонентами, які завантажують середу CLR у власні процеси і запускають виконання керованого коду, створюючи таким чином програмне середовище, що дозволяє використовувати кошти як керованого, так і некерованого виконання. Платформа .NET Framework не тільки надає кілька базових середовищ виконання, але також підтримує розробку базових середовищ виконання незалежними виробниками.

Наприклад, ASP.NET розміщує середовище виконання і забезпечує масштабування середовища для керованого коду на стороні сервера. ASP.NET працює безпосередньо з середовищем виконання, щоб забезпечити виконання додатків ASP.NET і веб-служб XML.

Оглядач Internet Explorer може служити прикладом некерованого додатка, що розміщує середу виконання (у вигляді розширень типів MIME). Розміщення середовища виконання в браузері Internet Explorer дозволяє впроваджувати керовані компоненти або елементи управління Windows Forms в HTML-документи. Таке розміщення середовища дозволяє виконувати керований мобільний код і користуватися його істотні переваги, зокрема виконанням в умовах неповної довіри і ізольованим зберіганням файлів.

На Рис 3.8 демонструється взаємозв'язок середовища CLR і бібліотеки класів з одними додатками і всією системою. На малюнку також показано, як керований код працює в межах ширшої архітектури.[42]

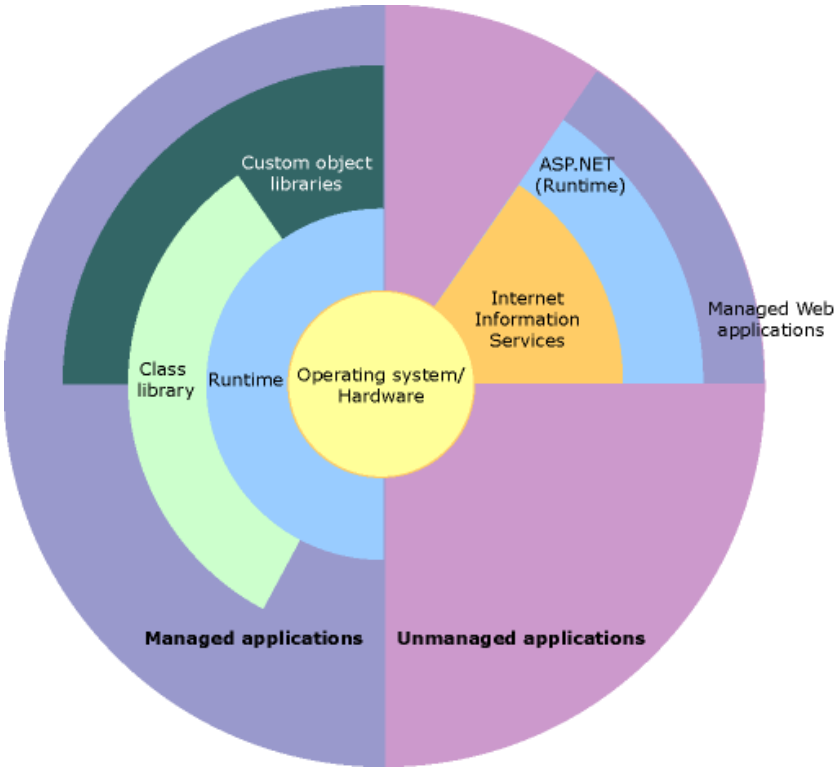

Рис 3.8 Взаємозв'язок середовища CLR і бібліотеки класів з одними додатками і всією системою

Бібліотека класів платформи .NET Framework являє собою колекцію типів, які тісно інтегруються із середовищем CLR. Бібліотека класів є об'єктно-орієнтованої. Вона надає типи, від яких керований код користувача може успадковувати функції. Це не тільки спрощує роботу з типами .NET Framework, але і скорочує час вивчення нових засобів платформи .NET Framework. Крім того, компоненти незалежних виробників можна легко поєднувати з класами платформи .NET Framework.

Наприклад, в класах колекцій .NET Framework реалізується набір інтерфейсів для розробки призначених для користувача класів колекцій. Призначені для користувача класи колекцій легко об'єднуються з класами .NET Framework.

Як і очікується від об'єктно-орієнтованої бібліотеки класів, типи .NET Framework дозволяють вирішувати типові завдання програмування, включаючи роботу з рядками, збір даних, підключення до баз даних і доступ до файлів. На додаток до цих звичайним завданням бібліотека класів містить типи,

які підтримують багато спеціалізовані сценарії розробки. Платформа .NET Framework може використовуватись для розробки наступних типів додатків і служб:

1. Консольні додатки.
2. Додатки з графічним інтерфейсом Windows (Windows Forms.
3. Додатки Windows Presentation Foundation (WPF).
4. Програми ASP.NET.
5. Служби Windows.
6. Сервісноорієнтовані програми для використання Windows Communication Foundation (WCF).
7. Додатки, що підтримують бізнес-процеси Windows Workflow Foundation (WF).[42]

3.2.3 Універсальна платформа Windows

UWP - це один з багатьох способів створення клієнтських додатків для Windows. Додатки UWP використовують API WinRT для надання потужних користувальницьких інтерфейсів і розширених асинхронних функцій, які ідеально підходять для пристроїв, підключених до Інтернету.

UWP - це один з варіантів створення додатків, які працюють на пристроях Windows 10 і можуть використовуватися на інших платформах. Додатки UWP можуть використовувати API Win32 і класи .NET.[43]

Основні переваги програми UWP:

1. Безпека. Додатки UWP керують, до яких ресурсів пристрою та даних вони мають доступ. Користувач повинен дозволити такий доступ.
2. Можливість використовувати загальний API на всіх пристроях під управлінням Windows 10.
3. Можливість використання окремих пристроїв і адаптації користувацького інтерфейсу до різних розмірів екрану, розширення і роздільної здатності.
4. Доступність в Microsoft Store, на всіх пристроях (або тільки тих,

яких ви вкажете), що працюють під управлінням Windows 10. У Microsoft Store передбачено кілька способів монетизувати ваш додаток.

5. Можливість встановлювати і видаляти додатки без ризику для комп'ютера або "деградації" ПО.

6. Захопливість: можливість використовувати живі плитки, push-повідомлення і призначені для користувача дії, які взаємодіють з тимчасовою шкалою Windows і функцією "Продовжити з місця зупинки" Кортани, для підтримки інтересу користувачів до додатка.

7. Можливість програмування на C #, C ++, Visual Basic і JavaScript. Для інтерфейсу можна використовувати WinUI, XAML, HTML або DirectX.[43]

3.3 Впровадження удосконаленого методу у програмний продукт на ОС Windows

Для досягнення поставленої цілі запропонований метод автентифікації буде впроваджено в програмний продукт.

Програмний продукт написано мовою C# з використанням .NET Framework
та UWP.

Перш за все нам потрібно визначитися з графічним інтерфейсом користувача. Для зручного користування да демонстрації роботи програми він повинен містити список видимих Bluetooth пристроїв, текстовий блок для виводу важливої для користувача інформації та три кнопки: для налаштування вибраного пристрою, для перемикання між режимами відображення (список містить пристрої, що вже спаровані, чи ті, що не утворюють пари) та кнопка скидання поточних налаштувань.

Для створення графічного інтерфейсу використовується Visual Studio Designer – це частина інтегрованої середи розробки Visual Studio призначена для генерації коду графічного інтерфейсу в інтерактивному режимі.

Спираючись на вимоги продукту було розроблено наступний графічний інтерфейс (рис. 3.9):

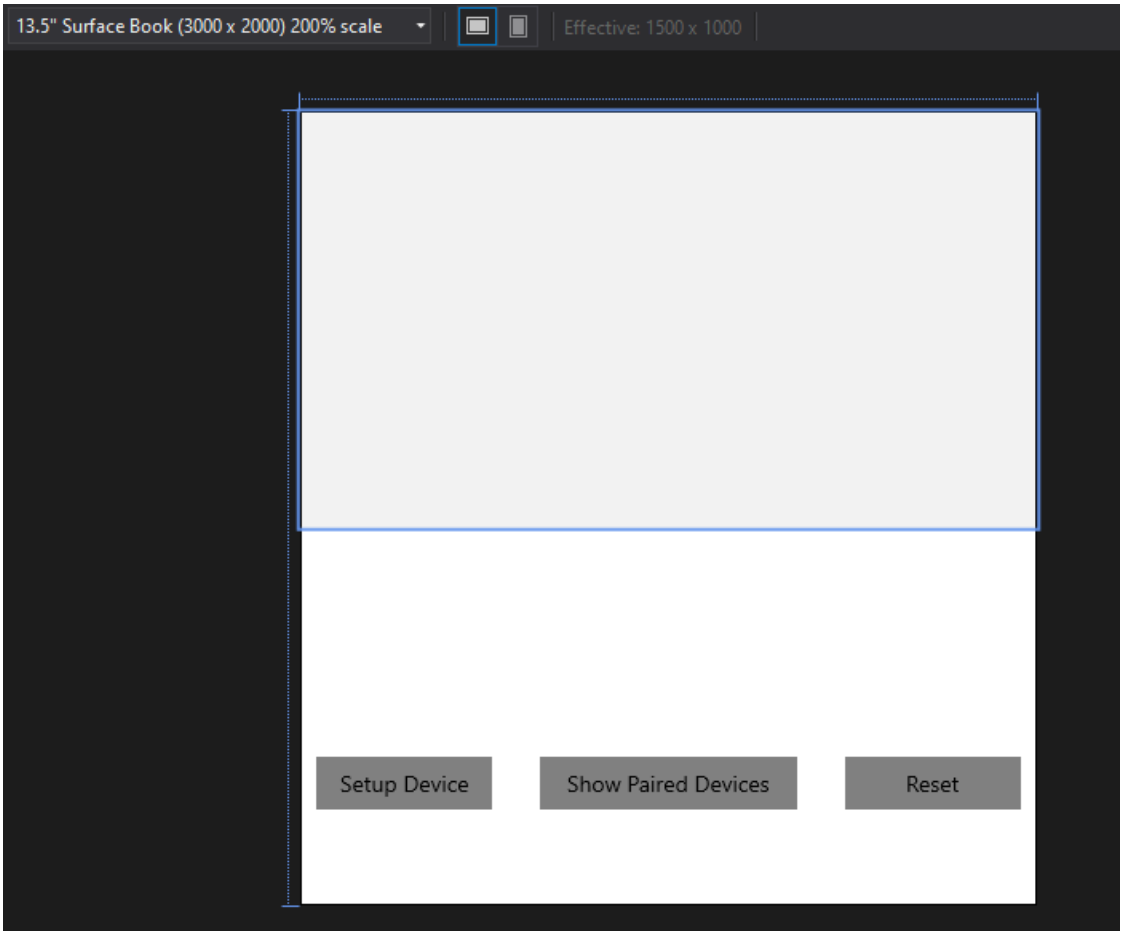

Рис. 3.9 Розроблений графічний інтерфейс

На основі цього Visual Studio Designer згенерував фрагмент коду зображений на рис 3.10.

```
<Page
    x:Class="BluetoothAuthenticatorUI.MainPage"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:local="using:BluetoothAuthenticatorUI"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    mc:Ignorable="d"
    Background="{ThemeResource ApplicationPageBackgroundThemeBrush}" Height="497" Width="461">
    <Grid>
        <ListBox Margin="0,0,0,236" Name="DevicesList"/>
        <Button Content="Setup Device" Height="33" Margin="10,404,0,0" VerticalAlignment="Top"
                Width="110" Foreground="Black"
                Background="Gray"  Visibility="Visible" Click="Button_Click" Name="PairButton"/>
        <Button Content="Show Paired Devices" Height="33" Margin="0,404,0,0" VerticalAlignment="Top"
                Width="161"  Foreground="Black" Background="Gray"  Visibility="Visible"
                RenderTransformOrigin="0.549,0.52" Click="Button_Click_1"
                Name="ChangeListButton" HorizontalAlignment="Center"/>
        <TextBlock HorizontalAlignment="Center" Margin="0,276,0,0" Text=""
                   TextWrapping="Wrap" VerticalAlignment="Top" Height="109"
                   Width="442" Name="ComunicationLabel" FontSize="20"/>
        <Button Content="Reset" Height="33" Margin="341,404,0,0" VerticalAlignment="Top"
                Width="110" Foreground="Black" Background="Gray"
                Visibility="Visible" Click="ResetButton_Click" x:Name="ResetButton"/>
    </Grid>
</Page>
```

Рис. 3.10 Фрагмент коду графічного інтерфейсу

Після створення графічного інтерфейсу потрібно додати функції, що будуть обробляти вхідні від користувача події. У випадку даного ПЗ користувач може взаємодіяти тільки з трьома кнопками, тому необхідно обробити події натискання на них. Для цього було створено три функції (рис.3.11), що будуть реагувати на натискання кожної з кнопок відповідно.

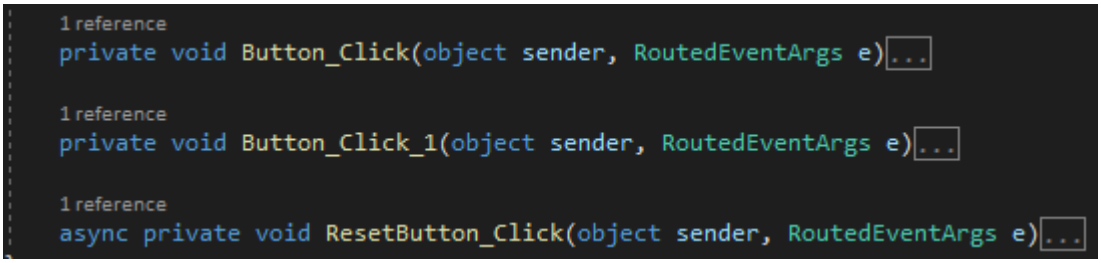

Рис. 3.11 Обробники подій кнопок

Для взаємодії з Bluetooth модулем комп'ютера та пошуку Bluetooth пристроїв будуть використані можливості UWP. Для цих цілей UWP пропонує наступні класи:

- BluetoothDevice
- DeviceInformation

BluetoothDevice - представляє пристрій Bluetooth.

DeviceInformation - цей клас дозволяє отримати доступ до відомих властивостей пристрою, а також додаткових властивостей, зазначених під час перерахування пристрою[44].

За допомогою методів цих класів ми маємо доступ до усіх Bluetooth пристроїв: як спарованих з комп'ютером, так і тих, що знаходяться в зоні дії Bluetooth сигналу. Із доступних Bluetooth пристроїв складатиметься список, що буде відображатися в компонеті ListBox графічного інтерфейсу.

Кнопка «Show paired/unpaired devices» відповідає за перемикання між списками пристроїв: спарованих (рис. 3.12),

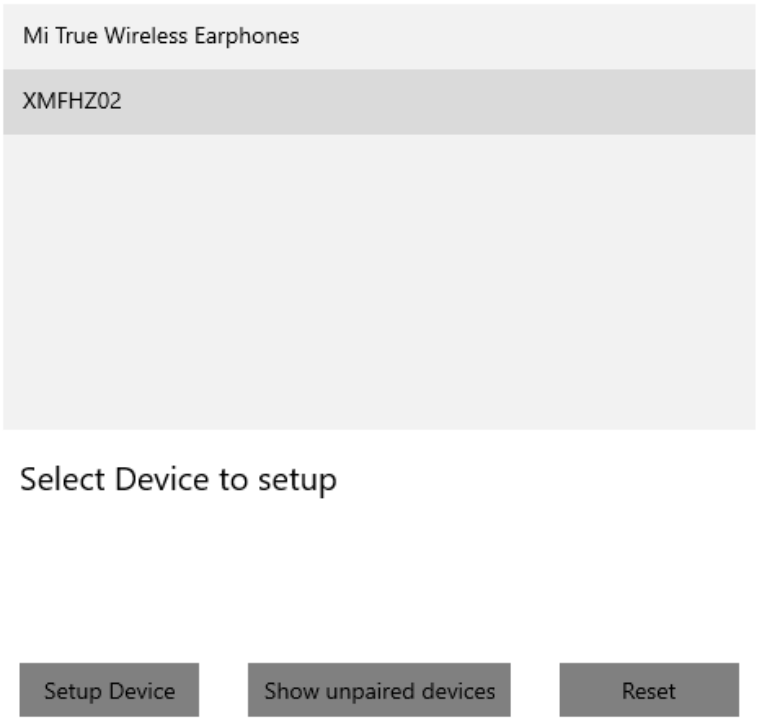

Рис. 3.12 Список спарованих пристроїв

та решти, що доступні по Bluetooth зв’язку (рис. 3.13).

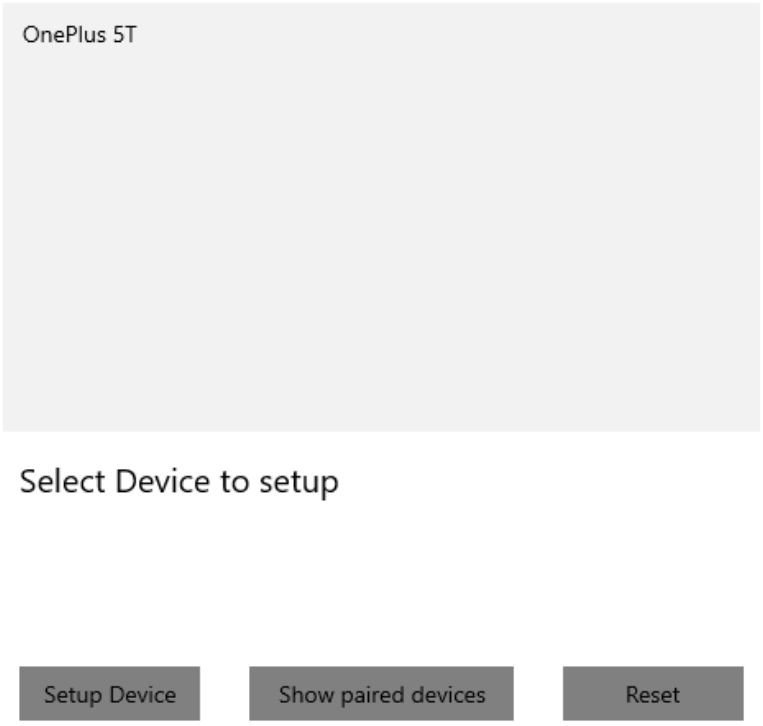

Рис. 3.13. Список доступних Bluetooth пристроїв

При виборі одного з пристроїв користувач матиме можливість налаштувати автентифікацію за допомогою цього пристрою.

Для цього в обробник кнопки «Setup device» потрібно додати код, що матиме наступні цілі:

- перевірить чи знаходиться даний пристрій серед спарованих раніше пристроїв, чи потрібно спершу розпочати процес утворення пари.
- якщо попередній крок не викликав помилок Bluetooth адреса вибраного пристрою буде збережена у пам’ять комп’ютера
- при наявності будь якої помилки, потрібно сповістити користувача про неї

Bluetooth адреса є унікальною інформацією, що буде використовуватись в подальшому для автентифікації користувача. Саме тому зберігати її в відкритому вигляді не можна. Для вирішення цієї проблеми Bluetooth адреса буде захешована за допомогою хеш-функції SHA256. Для цієї задачі буде використано клас SHA256CryptoServiceProvider, що являється частиною .NET Framework

SHA256CryptoServiceProvider - Визначає об'єкт-оболонку для доступу до реалізації алгоритму SHA256, що надається постачальником служб шифрування[45].

Реалізація обробника натиску кнопки, описаної вище, знаходиться на рис. 3.14.

```
private void Button_Click(object sender, RoutedEventArgs e)
{
    DeviceInformation info = currentList.Find(i => i.Name == DevicesList.SelectedValue?.ToString());
    if (info == null)
        return;
    BluetoothDevice.FromIdAsync(info.Id).Completed = async (device, d) =>
    {
        var bluetoothDevice = device.GetResults();
        {
            DevicePairingResult result = bluetoothDevice.DeviceInformation.Pairing.
                PairAsync(DevicePairingProtectionLevel.None).GetAwaiter().GetResult();
            if (!(result.Status == DevicePairingResultStatus.Paired ||
                result.Status == DevicePairingResultStatus.AlreadyPaired))
            {
                await Dispatcher.RunAsync(Windows.UI.Core.CoreDispatcherPriority.Normal, () =>
                {
                    ComunicationLabel.Text = "Can't pair device";
                });
            }
            else if (bluetoothDevice.IsInRange())
            {
                Windows.Storage.StorageFolder storageFolder = Windows.Storage.ApplicationData.Current.LocalFolder;
                Windows.Storage.StorageFile sampleFile = await storageFolder.CreateFileAsync(tokenFileName);
                await Windows.Storage.FileIO.WriteBufferAsync(sampleFile, CryptographicBuffer.CreateFromByteArray(
                    (new SHA256CryptoServiceProvider()).ComputeHash(Encoding.UTF8.GetBytes(bluetoothDevice.BluetoothDeviceId.Id))));
                await Dispatcher.RunAsync(Windows.UI.Core.CoreDispatcherPriority.Normal, () =>
                {
                    ComunicationLabel.Text = "Authenticated";
                    PairButton.IsEnabled = false;
                    ChangeListButton.IsEnabled = false;
                });
            }
            else
            {
                await Dispatcher.RunAsync(Windows.UI.Core.CoreDispatcherPriority.Normal, () =>
                {
                    ComunicationLabel.Text = "Device is not available!";
                });
            }
        }
    };
}
```

Рис. 3.14 Код обробника кнопки «Select Device»

В разі якщо було вибрано пристрій, з яким досі не створено пари, відбудеться процес створення цієї пари. Користувач побачить вікно, зображене на рис. 3.15.

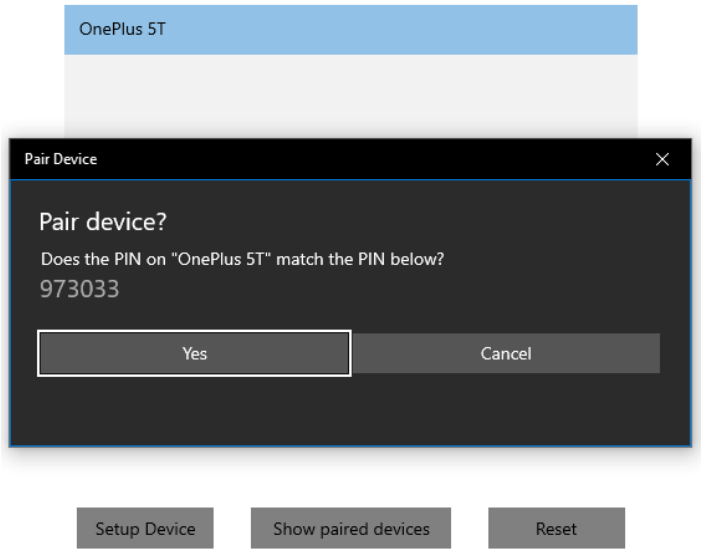

Рис. 3.15 Вікно підтвердження утворення пари

Схоже вікно з’являється і на пристрої, з яким утворюється пара. Після надання згоди, з’явиться вікно про успішне налаштування пристрою (рис. 3.16).

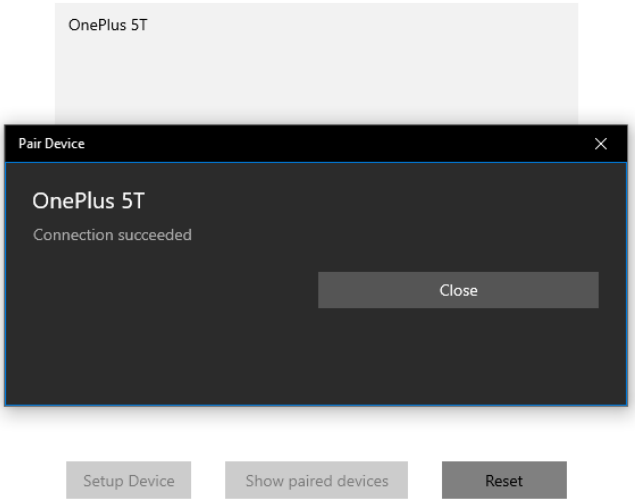

Рис. 3.16 Вікно про успішне налаштування

Налаштування пристрою пройдено успішно. Задля перевірки автентифікації треба перезапустити програму. В разі наявності попередньо налаштованого пристрою в діапазоні дії Bluetooth сигналу користувач отримає повідомлення про успішну автентифікацію (рис. 3.17).

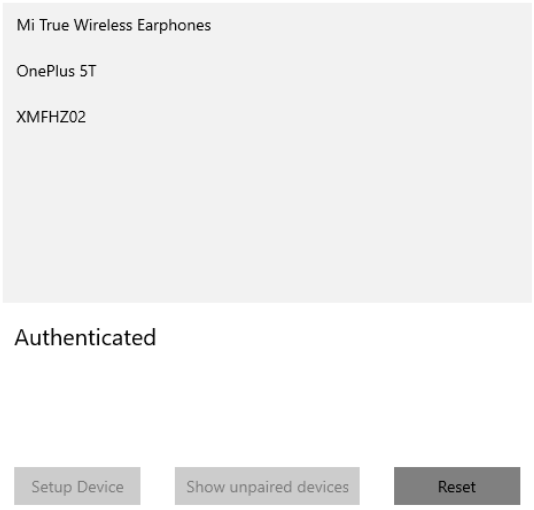

Рис. 3.17 Успішна автентифікація

В разі його відсутності, чи якщо відстань до пристрою занадто велика, буде виведено повідомлення про помилку автентифікації (рис.3.18). Для симуляції такого стану достатньо вимкнути Bluetooth на налаштованому пристрої.

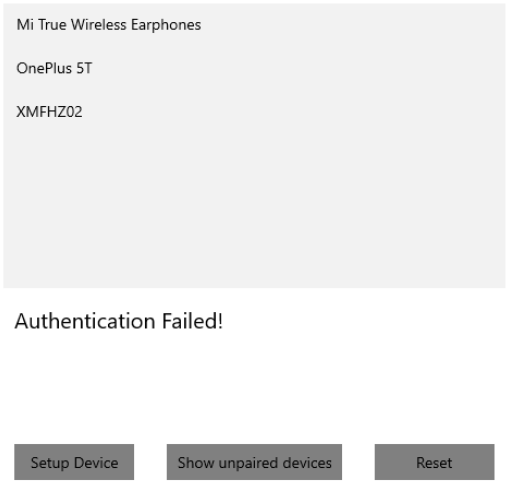

Рис.3.18 Помилка автентифікації

Кнопка «Reset» видаляє усі дані про попередньо налаштований пристрій для того, щоб користувач мав змогу його змінити чи повністю видалити.

3.4 Переваги удосконаленого методу

Усі з наведених у розділі 3.1 проблем багатофакторної автентифікації успішно можна вирішити за рахунок використання Bluetooth технології, для заміщення фактору «Я маю» в цьому процесі.

Перш за все, такий підхід вирішить проблему втомленого користувача, адже один з кроків у процесі багатофакторної автентифікації буде виконуватися автоматично та не потребуватиме додаткових дій від користувача. На відміну від варіанту з одноразовим паролем, користувач не буде відволікатися на повторне введення паролю.

Також цей метод перевершує існуючі тим, що підтвердження особистості користувача відбувається максимально швидко за рахунок проведення цього процесу локально, та відсутності затримки у вигляді часу на очікування СМС від системи.

Це також позбавляє від проблеми наявності зовнішніх факторів, що не залежать від користувача, проте впливають на процес автентифікації, таких як якість мережі в момент автентифікації.

В той же час будь-який Bluetooth пристрій підтримує одні і ті ж стандарти (профілі), що вирішує проблему сумісності пристроїв. На рис 3.19[32] зображена сумісність і підтримка функцій різних специфікацій Bluetooth.

Специфікація	Bluetooth 1.1	Bluetooth 1.2	Bluetooth 2.0	Bluetooth 2.1 plus EDR (Enhanced Data Rate)	Bluetooth 3.0	Bluetooth 4.0
Голосовий виклик	Yes	Yes	Yes	Yes	Yes	Yes
Скинути виклик	Yes	Yes	Yes	Yes	Yes	Yes
Останній набраний номер	Yes	Yes	Yes	Yes	Yes	Yes
Покращена стійкість до радіочастотного впливу		Yes	Yes	Yes	Yes	Yes
Діапазон 10 метрів	Yes	Yes	Yes	Yes	Yes	Yes
Діапазон 100 метрів			Yes	Yes	Yes	Yes

Рис. 3.19. Підтримка функцій різними версіями Bluetooth

На відміну від генераторів одноразових паролей даний метод автентифікації не є одноразовим та не вимагає наявності спеціалізованих додаткових пристроїв у користувача. Він також не вимагає додаткових дій зі сторони користувача, оскільки процес автентифікації відбувається автоматично за умови, що зареєстрований пристрій знаходиться в доступному діапазоні Bluetooth передавача.

3.5 Висновки до розділу

У третьому розділі дипломної роботи, було розроблено та впроваджено в програмний модуль удосконалений метод автентифікації за допомогою Bluetooth пристроїв. Після проведення досліджень та тестування програмного модуля, можна зробити висновок: удосконалений метод та розроблений модуль цілком відповідає поставленому питанню, має всі необхідні властивості для безпечної та зручної автентифікації.

Розроблений програмний модуль має зручний інтерфейс користувача, що покращує досвід користувача при його користуванні. Інтерфейс був розроблений з метою мінімізації дій користувача. Розроблений програмний модуль надає користувачу можливість власноруч керувати налаштуваннями пристрою для використання його в подальшому для автентифікації.

Удосконалений метод автентифікації позбавлений недоліків існуючих методів, що вплинуло на зручність та простоту процесу автентифікації, без нагальної потреби втручання користувача. Це позитивно вливає на досвід користування будь-яким програмним продуктом, де буде впроваджено даний метод, знижуючи ризики створення хибних даних в системі моніторингу атак від помилок надійних користувачів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Authentication. URL: http://www.webopedia.com/TERM/A/authentication.html (дата звернення: 25.09.2020).

2. H. Abie, "semanticscholar". URL: https://pdfs.semanticscholar.org/3733/2607f7a7ac828 4c514845957fd00583e5614.pdf (дата звернення: 25.09.2020).

3. A Review on Authentication Methods. URL: https://hal.archives-ouvertes.fr/hal-00912435/PDF/A_Review_on_Authentication_Methods.pdf (дата звернення: 25.09.2020).

4. Authentication protocol. URL: https://en.wikipedia.org/wiki/Authentication_protocol (дата звернення: 25.09.2020).

5. Vanek T.. "Autentizacní telekomunikacních a datových sítích". CVUT Prague. 4 March 2016.

6. Extensible Authentication Protocol. URL: https://doubleoctopus.com/security-wiki/protocol/extensible-authentication-protocol/ (дата звернення: 25.09.2020).

7. AAA protocols. URL: http://www.cisco.com/c/en/us/td/docs/net_mgmt/cisco_secure_access_control_system/5-1/user/guide/acsuserguide/rad_tac_phase.html (дата звернення: 26.09.2020).

8. Liu J. Introduction to Diameter. www.ibm.com. IBM. 24 January 2006.

9. Kerberos: The Network Authentication Protocol. URL: http://web.mit.edu/kerberos/ (дата звернення: 26.09.2020).

10. What is multi factor authentication. URL: https://www.loginradius.com/blog/2019/06/what-is-multi-factor-authentication/ (дата звернення: 26.09.2020).

11. FIPS 113. URL: https://csrc.nist.gov/publications/detail/fips/113/archive/1985-05-30 (дата звернення: 26.09.2020).

12. Sarah Al-Shareeda. Authentication Technologies for Cloud Computing, IoT and Big Data. 2019. URL: https://digital-library.theiet.org/content/books/10.1049/pbse009e_ch3 (дата звернення: 27.09.2020).

13. Закон України 2155-VIII «Про електронні довірчі послуги» від 05.10.2017. URL: https://zakon.rada.gov.ua/laws/show/2155-19/ed20171005#n9 (дата звернення: 27.09.2020).

14. НД ТЗІ 2.5-004-99 «Критерії оцінки захищеності інформації в комп'ютерних системах від несанкціонованого доступу» URL: http://www.dsszzi.gov.ua/dsszzi/control/uk/publish/article;jsessionid=5A88E6FEAB902185A2100ECC92151B59?showHidden=1&art_id=101853&cat_id=89734&ctime=1344501024406 (дата звернення: 28.09.2020).

15. Authentication. URL: https://www.acunetix.com/websitesecurity/authentication/ (дата звернення: 28.09.2020).

16. Authentication attacks and countermeasures. URL: https://sites.google.com/a/pccare.vn/it/security-pages/authentication-attacks-and-countermeasures (дата звернення: 28.09.2020).

17. WAP Authentication. URL: https://www.ibm.com/support/knowledgecenter/en/SSB2MG_4.6.0/com.ibm.ips.doc/concepts/wap_authentication.htm (дата звернення: 29.09.2020).

18. Electronic signature authentication. URL: https://www.docusign.com/esignature/electronic-signature-authentication/ (дата звернення: 29.09.2020).

19. Anonymous identification with cancelable biometrics. URL: https://ieeexplore.ieee.org/document/5297678 (дата звернення: 29.09.2020).

20. Fractional biometrics: Safeguarding privacy in biometric applications. URL: https://www.researchgate.net/publication/220066921_Fractional_biometrics_Safeguarding_privacy_in_biometric_applications (дата звернення: 29.09.2020).

21. Fake iris detection using structured light. URL: https://www.researchgate.net/publication/261153461_Fake_iris_detection_using_structured_light (дата звернення: 29.09.2020).

22. The Power of Personality The Comparative Validity of Personality Traits, Socioeconomic Status, and Cognitive Ability for Predicting Important Life Outcomes. URL: https://www.researchgate.net/publication/237822262_The_Power_of_Personality_The_Comparative_Validity_of_Personality_Traits_Socioeconomic_Status_and_Cognitive_Ability_for_Predicting_Important_Life_Outcomes (дата звернення: 29.09.2020).

23. Understanding password recommendations. URL: https://docs.microsoft.com/en-us/microsoft-365/admin/misc/password-policy-recommendations?view=o365-worldwide (дата звернення: 20.10.2020)

24. Password Authentication: Avoiding 4 Common Vulnerabilities. URL: https://swoopnow.com/password-authentication (дата звернення: 20.10.2020)

25. Smart Card Authentication | Raise Your Security Levels to a Higher Standard. URL: https://www.parallels.com/blogs/ras/smart-card-authentication (дата звернення: 25.10.2020)

26. Jain, A., Bolle, R. and Pankanti S. (1999). BIOMETRICS: Personal Identification in Networked Society. Kluwer Academic Publishers.

27. What are the Advantages & Disadvantages of a Digital Certificate. URL: https://www.techwalla.com/articles/what-are-the-advantages-disadvantages-of-a-digital-certificate (дата звернення: 02.11.2020)

28. Чиганов А.В. Современные проблемы «сильной» аутентификации / Молодой учёный № 4 (138) / 2017 – с.188-190

29. Americans are sick and tired of passwords and security questions. URL: https://macdailynews.com/2018/07/09/americans-are-sick-and-tired-of-passwords-and-security-questions/ (дата звернення: 06.10.2020)

30. Stephanie Ho, Brian Ng, Justin Kwong, Frank Wu «Security Analysis of Bluetooth Enabled Mobile Devices»

31. Daniel Filizzola, Sean Fraser, Nikita Samsonau «Security Analysis of Bluetooth Technology»

32. Is bluetooth 4.0 backwards compatible with bluetooth 2.0+EDR? URL: https://superuser.com/questions/588186/is-bluetooth-4-0-backwards-compatible-with-bluetooth-2-0edr

33. Що таке адреса Bluetooth (BD_ADDR): https://macaddresschanger.com/what-is-bluetooth-address-BD_ADDR (дата звернення: 04.12.2020).

34. Метод FHSS URL: http://itservis.ru/dokum/lan/wlan_metod_fhss.php (дата звернення: 04.12.2020).

35. Wang Y. Bluetooth positioning using RSSI and triangulation methods / Y. Wang, Xu Yang, Y. Zhao, Y. Liu, L. Cuthbert // 2013 IEEE 10th Consumer Communications and Networking Conference (CCNC). – 2013.

36. Севост’янов О.Р. Програмний код тестування URL: https://github.com/AS-13/test-Kalman-filter

37. САМЫЕ ПОПУЛЯРНЫЕ ОПЕРАЦИОННЫЕ СИСТЕМЫ МИРА 2020 URL: https://marketer.ua/stats-operating-system-2020/

38. A Tour of the C# Language [Електронний ресурс]. – Режим доступу: https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/

39. Структура программы [Електронний ресурс]. – Режим доступу: https://docs.microsoft.com/ru-ru/dotnet/csharp/tour-of-csharp/program-structure

40. Типы и переменные [Електронний ресурс]. – Режим доступу: https://docs.microsoft.com/ru-ru/dotnet/csharp/tour-of-csharp/types-and-variables

41. .NET Free. Cross-platform. Open source. URL:https://dotnet.microsoft.com/

42. Общие сведения о платформе .NET URL: https://docs.microsoft.com/ru-ru/dotnet/framework/get-started/overview

43. Что такое приложение UWP? URL: https://docs.microsoft.com/ru-ru/windows/uwp/get-started/universal-application-platform-guide

44. DeviceInformation Class URL: https://docs.microsoft.com/en-us/uwp/api/windows.devices.enumeration.deviceinformation?view=winrt-19041

45. SHA256CryptoServiceProvider Клас URL: https://docs.microsoft.com/ru-ru/dotnet/api/system.security.cryptography.sha256cryptoserviceprovider?view=net-5.0