Wydział Podstawowych Problemów Techniki
Politechnika Wrocławska

Kontrola dostępu do smartfona przy wykorzystaniu urządzenia smartband

Karolina Bąk
Nr indeksu: 244917

Praca inżynierska napisana
pod kierunkiem
Dr inż. Przemysława Błaśkiewicza

Politechnika
Wrocławska

Wrocław 2021

Spis treści

Wstęp

Praca swoim zakresem obejmuje projekt systemu stałej autoryzacji wykorzystujący prostą analizę behawioralną w czasie rzeczywistym na podstawie danych gromadzonych przez inteligentną opaskę dla smartfonów działających pod Androidem. Klucze bezpieczeństwa są rzadkim zjawiskiem w mobilnych systemach. Niska popularność tego rozwiązania może wynikać z ceny tych urządzeń, często niekompatybilnych ze smartfonem metodach połączenia oraz zastępowalności innymi metodami wieloskładnikowej autoryzacji.

Celem pracy jest zaprojektowanie i stworzenie aplikacji o następujących założeniach funkcjonalnych:

- użycie inteligentnej opaski (pot. smartband) jako klucza bezpieczeństwa,
- zapewnienie prywatności gromadzonych danych,
- monitorowanie zachowań użytkownika w czasie rzeczywistym,
- zabezpieczenie urządzenia mobilnego przed uzyskaniem dostępu przez osoby niepowołane,
- stworzenie praktycznego systemu stałej autoryzacji dedykowanego dla urządzeń mobilnych.

Istnieje garstka systemów o podobnej funkcjonalności: Gadgetbridge, aplikacja open-source zastępująca korzystające z chmury oprogramowanie producentów wybranych opasek, przy czym nie zapewnia żadnych funkcji z zakresu bezpieczeństwa; Haven, aplikacja monitorująca czynniki środowiskowe w celu uzyskania dowodów ataku, jednakże nie zabezpieczająca przed jego skutkami oraz Android Management API, usługa dla przedsiębiorstw umożliwiająca zarządzanie grupą urządzeń oraz zabezpieczenie ich poprzez zdalne wydawanie "polityk", przy czym rozwiązanie to nie jest aplikowalne poza środowiskami biznesowymi.

Praca składa się z czterech rozdziałów. W rozdziale pierwszym przedstawiono dogłębnie problem autoryzacji przy użyciu sprzętowych kluczy w smartfonach oraz braku prywatności wrażliwych danych gromadzonych przez smartbandy. Omówiono szczegółowo dane pobierane z sensorów w opasce oraz smartfonie. Scharakteryzowano zdarzenia, przy których ograniczany jest dostęp do urządzenia. Opisano rozwiązania podjęte w celu uniemożliwienia sabotażu aplikacji. Przeprowadzono analizę porównawczą istniejących rozwiązań z realizowanym systemem.

W rozdziale drugim przedstawiono szczegółowy projekt aplikacji w notacji UML. Wykorzystano diagramy przypadków użycia, klas, aktywności, sekwencji oraz stanów. Omówiono dokładnie projekt bazy danych. Wyczerpująco opisano protokoły komunikacji z opaską. Opisano w pseudokodzie i omówiono algorytmy blokujące dostęp do aplikacji.

W rozdziale trzecim określono technologie użyte w implementacji aplikacji: wybrany język programowania, wykorzystywane biblioteki, model opaski oraz typ bazy danych. Przedstawiono dokumentację techniczną wybranych kodów źródłowych.

W rozdziale czwartym przedstawiono wymagania aplikacji co do środowiska. Określono także sposób instalacji oraz konfiguracji aplikacji. Rozdział zawiera również przykłady działania dla użytkownika.

Końcowy rozdział jest podsumowaniem uzyskanych wyników.

Analiza zagadnienia

W niniejszym rozdziale omówiono mankamenty związane z uwierzytelnianiem na urządzeniach mobilnych oraz kwestie prywatności wrażliwych danych pochodzących z urządzeń typu smartband. Przedstawiono zarys systemu oraz jego założenia funkcjonalne. Określono, jakie dane będą rejestrowane przez aplikację oraz cel ich gromadzenia. Określono sposoby analizy danych pod kątem wykrywania sytuacji, w których smartfon jest pozostawiony bez nadzoru. Opisano mechanizmy podjęte w celu zabezpieczenia systemu oraz przechowywanych danych. Porównano istniejące rozwiązania z proponowanym w pracy, wskazując na innowacje oraz różnice.

2.1 Przedstawienie problemu

Kwestia bezpieczeństwa smartfonów jest w dzisiejszych czasach niezwykle ważną sprawą. Powszechnie stosowane metody ograniczenia dostępu, takie jak uwierzytelnienie przy użyciu hasła bądź odcisku palca, są niewystaraczające. Szczególnie jest to widoczne przy atakach fizycznych, gdzie na przykład można:

- wykorzystać nieprzytomność użytkownika, by użyć jego odcisku palca;
- poznać hasło w postaci symbolu na podstawie śladów palców na ekranie dotykowym;
- uzyskać dostęp, gdy użytkownik pozostawi odblokowane urządzenie bez opieki.

Zważywszy na fakt, iż telefony komórkowe stają się coraz bardziej powszechne[11] oraz zastępują komputery jako urządzenia wykorzystywane do łączenia się z siecią (około połowa ruchu sieciowego pochodzi z urządzeń mobilnych [3]), przechowują one wiele wrażliwych danych o swoich użytkownikach. Dlatego koniecznością jest wprowadzenie dodatkowego systemu zabezpieczeń, w szczególności wykorzystanie uwierzytelnienia wielopoziomowego, w celu zabezpieczenia urządzenia przed dostępem osób niepowołanych. Często można spotkać się z wykorzystaniem smartfonów jako autentykatorów (kody SMS oraz dedykowane aplikacje) do innych systemów informatycznych. Natomiast do autoryzacji dostępu do smartfona nie wykorzystywane są żadne dodatkowe autentykatory. Głównymi przeszkodami do ich implementacji są:

- niepraktyczność;
- monofunkcyjność.

Nieporęczność fizycznych kluczy bezpieczeństwa objawia się szczególnie w ich formie - są to niewielkie urządzenia przypominające pamięć USB lub kartę płatniczą. Dzięki temu łatwo je zgubić lub o nich zapomnieć, co uniemożliwia użytkownikowi dostęp do systemu. Często też w smartfonie brakuje niezbędnej do odczytania klucza infrastruktury, na przykład czytnika smart cardów czy modułu NFC. [4]. Do niepraktyczności tego rozwiązania przyczynia się także wyżej wynieniona monofunkcyjność kluczy. Oferują one jedynie autoryzację użytkownika przy użyciu kluczy kryptograficznych bądź jednorazowych kodów i służą do logowania na stronach internetowych oraz przy autoryzacji w niektórych aplikacjach, co znacznie ogranicza pole ich zastosowania. Dlatego potrzebne jest świeże spojrzenie na tą technologię, które pozwoli stworzyć przystępne systemy stałej autoryzacji bazujące na wielu czynnikach środowiskowych oraz wykorzystujące powszechnie używane urządzenia by skutecznie zabezpieczyć dane szerokiej bazy użytkowników smartfonów.

Platformy mobilne jako dynamicznie rozwijające się technologie wspierają wachlarz urządzeń peryferyjnych, które mogły by zostać użyte jako klucz bezpieczeństwa. Jednym z nich jest **inteligentna opaska**,

potocznie zwana “smartband” bądź “fitness tracker”. Jest to urządzenie o kształcie zegarka na rękę monitorujące aktywność użytkownika taką, jak: ilość wykonanych kroków, puls czy sen. Dane gromadzone przez opaski są przesyłane protokołem Bluetooth do aplikacji towarzyszącej udostępnionej przez producenta, skąd zostają przesłane na zewnętrzne serwery. Nie jest to bezpieczne rozwiązanie, zważywszy na: wrażliwość powyższych informacji, powiązanie ich z danymi osobowymi użytkownika oraz fakt, że mogą zostać udostępnione osobom trzecim [1]. Z tego powodu konieczny jest rozwój systemów przechowywania informacji o aktywności pochodzących z inteligentnych opasek, które zapewnią użytkownikowi prywatność i nie będą bezpośrednio powiązane z producentem danego urządzenia.

2.2 Opis systemu

Zniwelowanie słabych punktów autentykacji przy użyciu kluczy sprzętowych jest niezwykle ważne przy implementacji tego rozwiązania w urządzeniach mobilnych. W pracy skupiono się na usprawnieniu poniższych niedoskonałości tej technologii:

- prawdopodobieństwo utraty urządzenia autoryzującego;
- niska powszechność kluczy sprzętowych;
- ograniczona możliwość stałej autoryzacji.

Proponowany system opiera się na wykorzystaniu smartbanda jako inteligentnego klucza sprzętowego. Autoryzacja odbywa się poprzez analizę danych o aktywności pobieranych z opaski w krótkich odstępach czasu. Po wykryciu sytuacji, gdzie smartfon jest prawdopodobnie poza nadzorem użytkownika, następuje uruchomienie blokady wybranych aplikacji do momentu wprowadzenia poprawnego hasła w systemie. Uniemożliwienie dostępu dokonuje się poprzez monitorowanie, która aplikacja znajduje się na pierwszym planie systemu Android. W przypadku wykrycia niedozwolonego programu użytkownik zostaje przeniesiony do aktywności odpowiedzialnej za autoryzację.

Wybór smartbanda do pełnienia funkcji klucza został podyktowany dużą popularnością urządzeń tego typu. Na rynku dostępnych jest wiele niskobudżetowych modeli, które gromadzą dane wystarczające do dość dokładnego określenia aktywności użytkownika. Z tego powodu inteligentne opaski są idealne, by oprzeć o nie system stałej autoryzacji. Dużą zaletą smartbanda jest jego niepozorna forma, czyli zegarek na rękę. Użytkownicy noszą go przez dużą część dnia, a nawet w nocy, przez co znacznie zmniejsza się ryzyko jego utraty bądź kradzieży. Kolejnym atutem tego urządzenia jest fakt, iż pełni ono znacznie więcej funkcji niż klucz sprzętowy. Dzięki temu smartband jest znacznie bardziej praktyczny dla użytkownika.

2.2.1 Założenia funkcjonalne

Wymagania postawione w niniejszej pracy, to:

- implementacja systemu blokującego dostęp do aplikacji;
- zapewnienie komunikacji z inteligentną opaską przy wykorzystaniu protokołu Bluetooth Low Energy;
- lokalne przechowywanie danych o aktywności użytkownika;
- zabezpieczenie aplikacji przed najczęstszymi atakami;
- wdrożenie prostej analizy zachowania użytkownika w czasie rzeczywistym.

2.2.2 Charakterystyka gromadzonych danych

System opiera się w głównej mierze o informacje rejestrowane przez inteligentną opaskę. Zaliczają się do nich:

- liczba wykonanych kroków danego dnia;

- aktualna wartość pulsu;
- moment zaśnięcia;
- moment zdjęcia opaski.

Powyższe informacje pozwalają określić stan fizyczny użytkownika, co jest kluczowe dla działania systemu. Dodatkowo dane o aktywności są uzupełniane o wartość sensora liczącego kroki w telefonie. Dzięki temu możliwa jest detekcja sytuacji, w których osoba eksploatująca może nie być w stanie nadzorować swojego telefonu, na przykład podczas snu bądź po pozostawieniu go na biurku w pracy. Przechowywane są także podstawowe informacje o opasce takie, jak: adres MAC oraz stan baterii. Umożliwia to ponowne połączenie z opaską oraz monitorowanie stanu urządzenia w aplikacji.

Oprócz informacji o aktywności system przechowuje listę zainstalowanych aplikacji. Pozwala to użytkownikowi dostosować działanie systemu do własnej preferencji. Najważniejszą przechowywaną informacją jest hasz hasła użytkownika, które jest wymagane do odblokowania dostępu do wybranych wcześniej aplikacji.

2.2.3 Analiza aktywności

Ważną częścią pracy jest wykrywanie sytuacji, w których smartfon jest poza nadzorem. Aby było to możliwe system bada aktywność użytkownika, korzystając z określonych w powyższym podrozdziale danych, pod kątem czterech zdarzeń:

- opaska traci połączenie ze smartfonem;
- użytkownik zasypia;
- występują znaczne rozbieżności pomiędzy zarejestrowanymi krokami;
- użytkownik zdejmuje opaskę.

Utrata połączenia wykrywana jest na podstawie metod nasłuchujących zmiany w statusie połączenia Bluetooth. Sen wykrywany jest poprzez otrzymanie powiadomienia z opaski o zarejestrowaniu odpowiedniego zdarzenia. Rozbieżności w rejestrowanych krokach monitorowane są przez porównanie tempa wzrostu kroków mierzonych przez smartbanda oraz telefon, a zdjęcie opaski rozpoznaje się poprzez brak wykrywanego pulsu, bądź poprzez otrzymanie powiadomienia ze smartbanda. W przypadku wykrycia jednej z powyższych sytuacji następuje automatyczne uruchomienie blokady aplikacji.

2.2.4 Zabezpieczenie systemu

Aby proponowany system zapewniał ochronę przed dostępem przez osoby niepowołane musi działać nieprzerwanie i być odporny na wyłączenie go przez atakującego. W tym celu usługi systemu są zaimplementowane jako *Foreground Service*, by działać stale w tle w zgodzie z limitami obowiązującymi od Androida Oreo[8]. Wykorzystano technologię *Wake Lock*[10] w celu umożliwienia aplikacji pozostania w stanie pełnej sprawności w przypadku, gdy telefon przechodzi w *Doze Mode*[9]. Wdrożono także *BroadcastReceiver*, który jest odpowiedzialny za monitorowanie restartów urządzenia. Po wykryciu ukończonego uruchomienia smartfona, usługa blokująca oraz gromadząca dane są restartowane według stanu sprzed wyłączenia urządzenia.

System jest także odporny na najpopularniejsze podatności w aplikacjach mobilnych związanych z danymi medycznymi[2]. Dzięki lokalnemu przechowywaniu informacji zapewniona jest odporność na ataki za pośrednictwem sieci. System zabezpieczono przed *Intent spoofing*, dzięki wykorzystaniu jedynie dokładnie sprecyzowanych Intentów oraz zabezpieczeniu komponentów przed exportem do innych aplikacji. By zapewnić bezpieczeństwo gromadzonych danych baza danych oraz plik przechowujący hasło zostały zaszyfrowane przy użyciu algorytmu szyfrowania AES. Natomiast mniej ważne informacje są przechowywane w *SharedPreferences*, do których dostęp ma tylko projektowany system.

2.3 Analiza porównawcza istniejących systemów

Na rynku znajduje się wąskie grono rozwiązań o podobnych funkcjonalnościach. Poniżej zaprezentowano najciekawsze z nich. Określono ich zalety oraz wady, a także porównano je z systemem zaprezentowanym w pracy.

2.3.1 Yubikey

Yubikey to nowoczesne klucze sprzętowe produkowane przez Yubico, wykorzystywane jako część wieloskładnikowej autoryzacji bądź autentykacji bazowanej na jednorazowych hasłach w szerokim gronie serwisów internetowych oraz systemów operacyjnych. Wspierają wiele protokołów kryptograficznych i autentykacyjnych, w tym: WebAuthn, FIDO2, U2F, smart cardy kompatybilne z PIV oraz Yubico OTP. Modele dedykowane urządzeniom mobilnym do komunikacji ze smartfonem wykorzystują moduł NFC, USB-C oraz złącze Lightning. Autoryzacja odbywa się poprzez umieszczenie klucza w złączu USB-C bądź przystawienie go do tyłu telefonu dla urządzeń z włączonym NFC[15].

Yubikey posiada wiele zalet. Jest wspierany przez dużą liczbę serwisów i systemów, dzięki czemu wachlarz aplikacji autentykacyjnych oraz kodów SMS czy wiadomości e-mail można zastąpić jednym urządzeniem. Pomaga uniknąć wykradnięcia haseł poprzez phishing czy przechwycenie SMSa. Jest prosty w użyciu dla użytkownika i nie wymaga ładowania. Jest również odporny na wodę oraz zgniecenie.

Dużą wadą Yubikey jest jego cena. Modele zapewniające autoryzację na smartfonach kosztują na tą chwilę minimum 45€ bez podatku VAT[14], czyli około 200 zł. Dla zwykłego użytkownika może być to zbyt duża kwota, gdy może skorzystać z darmowych wariantów dwuskładnikowej autoryzacji. Forma klucza (małe urządenie przypominające pendrive) sprzyja jego łatwemu zgubieniu, co uniemożliwia dostęp do serwisów, które z niego korzystały. By temu zaradzić producent zaleca posiadać zapasowy klucz, co wiąże się z dodatkowym wydatkiem rzędu 200 zł. Nie należy także zapominać o tym, że nie wszystkie telefony wspierają NFC oraz USB-C. Podczas, gdy rynek smartfonów dąży do wdrożenia powszechnie standardu USB-C, w przypadku NFC nie wszędzie jest on potrzebny. Ów moduł służy głównie do płatności mobilnych, dlatego na przykład w Chinach, gdzie powszechny jest system płatności przez kody QR[13], jest po prostu zbędny. Zważając na popularność chińskich telefonów na światowym rynku prawdopodobnym jest, iż nawet nowe modele nie będą wspierać technologii NFC, przez co utrudnią, a nawet uniemożliwią korzystanie z kluczy Yubikey.

W proponowanym rozwiązaniu jako klucz sprzętowy zostało wykorzystane urządzenie, które eliminuje wymienione wyżej wady Yubikey. Inteligentna opaska jest przeznaczona do noszenia na ręce, dzięki czemu ciężej ją zgubić lub ukraść. Smartband komunikuje się ze smartfonem poprzez wykorzystywany powszechnie moduł Bluetooth, co pozwoli wdrożyć system w znacznie szerszym gronie urządzeń. Kolejnym atutem wybranego urządzenia jest jego cena. Inteligentną opaskę można nabyć za mniej niż 100 zł, co sprawia, że jest przystępna dla wielu użytkowników. Najważniejszą różnicą między Yubikey a proponowanym systemem jest sposób autentykacji. Dzięki zastosowaniu opaski można stale autoryzować użytkownika bazując na wielu zmiennych czynnikach w przeciwieństwie do Yubikey, które jest jedynie nośnikiem przechowującym klucz, który jest wykorzystywany przy pojedynczych logowaniach bądź jako dodatkowa autoryzacja przy wrażliwych czynnościach.

2.3.2 Haven

Haven jest darmową aplikacją open-source dla urządzeń działających pod systemem Android zaprojektowaną w celu monitorowania aktywności wokół urządzenia, korzystając z jego wbudowanych sensorów. Przy wykryciu zmian w środowisku aplikacja gromadzi zdjęcia oraz nagrania dźwięku, po czym wysyła je poprzez Signala bądź Tora do użytkownika. Aplikacja została stworzona z myślą o dziennikarzach śledczych, którzy są narażeni na ataki ze strony policji bądź innych intruzów[5].

Główną zaletą Haven jest to, iż potrafi zastąpić drogie fizyczne systemy bezpieczeństwa. Do korzystania z tego rozwiązania wystarczy stary telefon z Androidem oraz opcjonalnie karta SIM, by zapewnić dostęp

do mobilnego Internetu. Zapewnia to użytkownikowi tani, a także łatwy w przenoszeniu system pozwalający monitorować na przykład pokój w hotelu. Pozwala to zdobyć dowody w przypadku ataku typu “evil maid”, czyli gdy osoba trzecia uzyskuje fizyczny dostęp do urządzenia, wykorzystując nieobecność właściciela, w celu wykradnięcia danych bądź zainstalowaniu szpiegującego oprogramowania [12].

Wadą Haven jest zdecydowanie fakt, że nie zapobiega ona atakom, tylko zdobywa dowody ich wystąpienia. Kolejnym mankamentem aplikacji jest jej nadmierna czułość. Wykrywane są mikroruchy smartfona oraz drobne dźwięki, przez co korzystanie z Haven w głośniejszych środowiskach, jak na przykład w biurze może wiązać się z setkami fałszywie wykrytych zdarzeń.

Zaproponowany w pracy system również skupia się na atakach wykonywanych poprzez fizyczny dostęp do urządzenia, lecz w zupełnie inny sposób. Proponowana aplikacja ma na celu wykorzystanie danych ze smartbanda w celu zabezpieczenia smartfona, gdy użytkownik nie jest w stanie nadzorować go samodzielnie. W przeciwieństwie do Haven, które wykorzystuje telefon do zbierania informacji, ale w żadnym stopniu nie korzysta z nich żeby uniemożliwić dostęp do urządzenia. Oba rozwiązania monitorują przeróżne wydarzenia rejestrowane przez dostępne sensory. Podczas, gdy Haven skupia się na środowisku, proponowana aplikacja skupia się na samym użytkowniku. Haven również w żadnym stopniu nie analizuje zbieranych danych, gdyż jego głównym zadaniem jest jedynie raportowanie tego, co dzieje się wokoło. Z kolei proponowane rozwiązanie w pewnym stopniu bierze pod lupę gromadzone dane i na ich podstawie określa, kiedy uruchomić blokadę urządzenia.

2.3.3 Android Management API

Android Management API jest częścią Android Enterprise, inicjatywy dostarczającej deweloperom narzędzi pozwalających budować rozwiązania dla przedsiębiorstw w celu zarządzania flotą mobilnych urządzeń[6]. Program ten jest dedykowany dostawcom usług zarządzania mobilnością w przedsiębiorstwie (EMM). Deweloperzy zapewniają swoim klientom lokalną bądź opartą na chmurze konsolę EMM. Wewnątrz konsoli klienci generują tokeny rejestracji urządzeń oraz tworzą zasady zarządzania (policies). Zasada zarządzania reprezentuje grupę ustawień rządzących zachowaniem zarządzanego urządzenia oraz zainstalowanymi aplikacjami. Następnie urządzenia są zapisywane do systemu przy użyciu wcześniej stworzonych tokenów. Podczas rejestracji, każde urządzenie instaluje aplikację towarzyszącą API, Android Device Policy. Kiedy do danego urządzenia są przyporządkowane zasady, powyższa aplikacja automatycznie wdraża je.

Android Management API daje niespotykane poza aplikacjami systemowymi możliwości zarządzania urządzeniem. Pozwala między innymi na:

- wyłączenie określonych modułów komunikacji takich, jak Bluetooth, Wi-Fi, SMS, rozmowy czy USB;
- blokadę instalacji bądź dezinstalacji aplikacji;
- dostosowanie aplikacji dostępnych w sklepie Play;
- wymuszenie określonych ustawień sieci;
- masowe nadanie pozwoleń aplikacjom;
- określenie sposobów autoryzacji oraz wymogów hasła;
- włączenie aplikacji w trybie Kiosk;
- zdalne wymazanie danych z urządzenia[7].

Główną wadą tego API jest brak możliwości zastosowania go poza przedsiębiorstwami. Urządzeniom korzystającym z tego rozwiązania zasady narzucane są odgórnie przez administratora, więc przy ogromnej liczbie smartfonów, gdzie każdy wymagałby innego zestawu zasad oraz ich aktualizacji na bieżąco, zarządzanie byłoby kłopotliwe. Jest to fundamentalny problem, przez który proponowany system nie może wykorzystać możliwości zabezpieczających Android Management API.

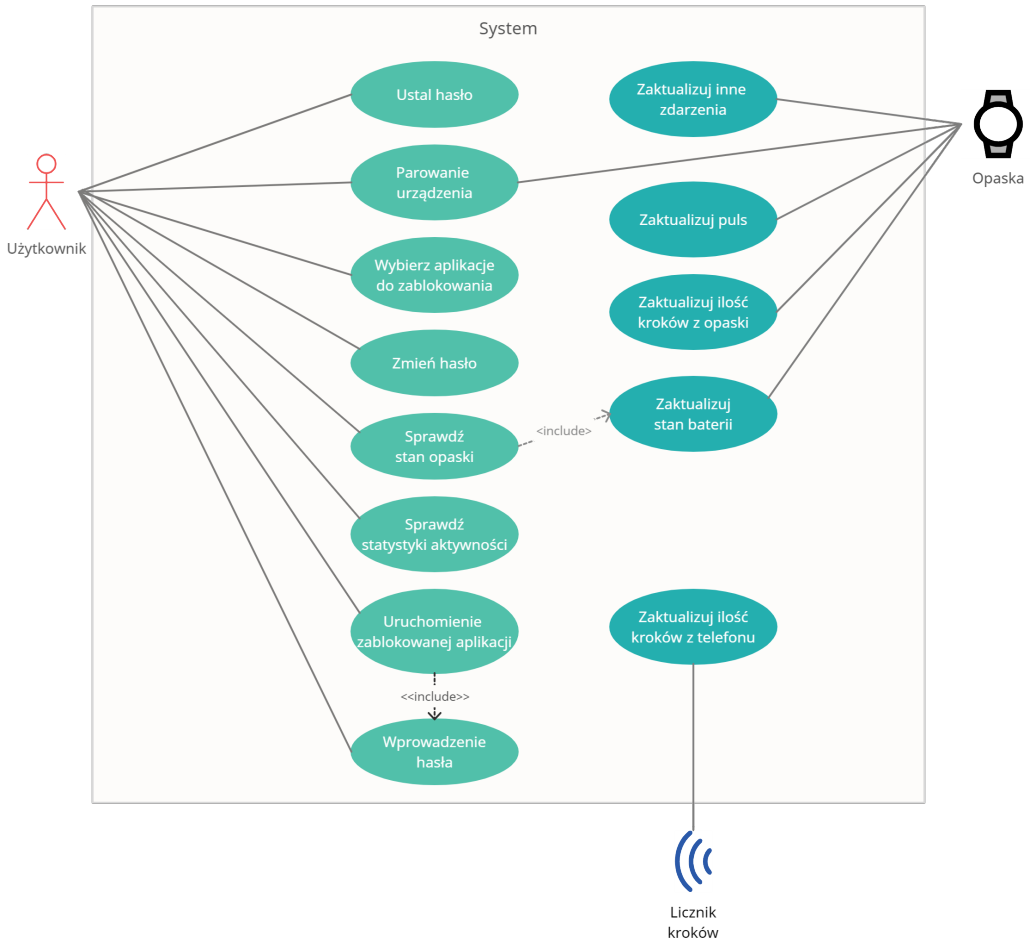
Porównując oba rozwiązania można zauważyć, iż są swoimi przeciwieństwami. Tworzony system jest z założenia czysto lokalny oraz tworzony z myślą o zwykłych użytkownikach, dlatego wszystkie mechanizmy zabezpieczające również muszą być aplikowalne bez udziału zewnętrznych serwisów, a także konfigurowalne przez użytkownika. Z kolei przy wykorzystaniu tego API konieczna jest rejestracja urządzenia w systemie EMM danego przedsiębiorstwa, a zasady dotyczące bezpieczeństwa są narzucane z góry.

Projekt aplikacji

W tym rozdziale przedstawiono szczegółowy projekt systemu korzystając z notacji UML oraz uwzględniając założenia funkcjonalne z rozdziału 2. Scharakteryzowano przypadki użycia oraz towarzyszące im scenariusze.

3.1 Przypadki użycia

Poniżej przedstawiono ogólny diagram przypadków użycia 3.1. Szczegółowe scenariusze zostały zdefiniowane w odpowiednich podsekcjach tego podrozdziału.



Rysunek 3.1: Diagram przypadków użycia w systemie.

3.1.1 Ustal hasło

W tym przypadku użytkownik systemu ustala hasło, które jest potrzebne przy odblokowywaniu dostępu do zablokowanych aplikacji. Aktorami są Użytkownik oraz System. Obecny przypadek użycia jest inicjowany przy pierwszym uruchomieniu aplikacji, kiedy nie istnieje jeszcze zaszyfrowany plik, w którym będzie przechowywane hasło. Po wykonaniu przypadku w systemie zostaje zarejestrowane hasło użytkownika, które będzie później wykorzystane w celu dezaktywacji blokady aplikacji. Scenariusz składa się z następującego przepływu głównego:

1. Aplikacja wyświetla formularz zawierający pola Hasło oraz Powtórz hasło.
2. Użytkownik wprowadza identyczne hasła do podanych pól oraz zatwierdza wprowadzone dane przyciskiem znajdującym się poniżej.
3. System sprawdza zgodność haseł oraz czy spełniają wymagane kryteria.
4. System tworzy zaszyfrowany plik i zapisuje w nim hasz hasła.

Alternatywnie:

3. Jeśli wprowadzone hasła nie są zgodne:
    1. Zostaje wyświetlony komunikat o błędzie.
    2. Następuje powrót do kroku numer 2 w głównym przepływie.

3.1.2 Parowanie urządzenia

W tym przypadku użytkownik systemu skanuje otoczenie, korzystając z modułu Bluetooth w poszukiwaniu najbliższej opaski MiBand, a następnie nawiązuje z nią pierwsze połączenie. Aktorami są Użytkownik, System oraz Opaska. Przypadek ten występuje przy pierwszym uruchomieniu systemu, kiedy zostało już ustalone hasło odblokowujące. Po zakończeniu system jest sparowany z opaską MiBand, z którą komunikacja jest kluczowym punktem działania systemu. W systemie zostaje także zapisany adres MAC opaski, dzięki czemu będzie można z łatwością ponownie połączyć się z nią. Główny przepływ składa się z następujących kroków:

1. Użytkownik naciska przycisk “Skan”.
2. System rozpoczyna skanowanie urządzeń Bluetooth Low Energy w celu znalezienia Opaski.
3. System wyświetla znalezione Opaski w formie listy.
4. Użytkownik wybiera Opaskę, do której się podłączy poprzez naciśnięcie na jej nazwę.
5. System tworzy więź z wybraną Opaską i inicjuje pierwsze połączenie.

Alternatywnie:

3. Jeśli nie znaleziono żadnej Opaski:
    1. Zostaje wyświetlony komunikat o błędzie.
    2. Następuje powrót do kroku numer 1 w głównym przepływie.

5. Jeśli wystąpi błąd połączenia z Opaską:
    1. Następuje powrót do kroku numer 4 w głównym przepływie.

3.1.3 Wybierz aplikacje do zablokowania

3.1.4 Zmień hasło

3.1.5 Sprawdź stan opaski

3.1.6 Sprawdź statystyki aktywności

3.1.7 Uruchomienie zablokowanej aplikacji

3.1.8 Wprowadzenie hasła

3.1.9 Zaktualizuj inne zdarzenia

3.1.10 Zaktualizuj puls

3.1.11 Zaktualizuj ilość kroków z opaski

3.1.12 Zaktualizuj stan baterii

3.1.13 Zaktualizuj ilość kroków z telefonu

3.2 Diagramy klas

W tej sekcji należy przedstawić diagramy klas dla odpowiednich elementów systemu zidentyfikowane na podstawie wcześniejszych rozważań

3.3 Diagramy aktywności

W tej sekcji należy przedstawić diagramy aktywności dla elementów systemu i odpowiednich procesów wynikające z wcześniejszej analizy.

3.4 Diagramy sekwencji

W tej sekcji należy przedstawić diagramy sekwencji dla obiektów systemu zidentyfikowanych na podstawie wcześniejszych rozważań. Należy wykorzystać nazewnictwo wprowadzone w poprzednich rozdziałach, w szczególności odpowiadające definicjom wprowadzonych klas.

3.5 Diagramy stanów

W tej sekcji należy przedstawić diagramy stanów w których może znaleźć się system. Diagramy te są szczególnie istotne przy projektowaniu systemów czasu rzeczywistego.

3.6 Projekt bazy danych

W tej sekcji należy przedstawić projekt bazy danych. Należy omówić wycinek rzeczywistości i odpowiadające mu zidentyfikowane elementy systemu, których wartości będą podlegać utrwalaniu. Należy przedyskutować wybór typów danych dla atrybutów poszczególnych obiektów. Należy uzasadnić wybór platformy DBMS. Dla relacyjnych baz danych należy przedyskutować jej normalizację.

3.7 Opis protokołów

- inicjalizacja komunikacji z opaską

W projektowanym systemie główną rolę gra inteligentna opaska. Komunikuje się ona ze smartfonem przy użyciu technologii Bluetooth Low Energy oraz protokołu ATT. BLE w porównaniu do klasycznego połączenia Bluetooth wykorzystuje znacznie niższe zasoby energii zachowując podobny zasięg, dzięki czemu znalazło szerokie zastosowanie w urządzeniach peryferyjnych.

3.7.1 Protokół ATT

Protokół Attribute pozwala urządzeniom odczytywać i zapisywać drobne dane przechowywane na serwerze. Przechowywane wartości są nazywane atrybutami. Atrybuty identyfikowane są poprzez UUID, aby określić typ przechowywanych w nich danych. UUID mogą być powszechnie znanymi numerami zdefiniowanymi w oficjalnej specyfikacji Bluetooth albo być określone przez producenta jako 128-bitowa liczba. Wiadomości w tym protokole są przesyłane przez kanały L2CAP, znanymi jako nośniki ATT. W ATT występują dwie role: Klienta oraz Serwera. Urządzenie może pełnić obie te role jednocześnie. Serwer przechowuje atrybuty oraz akceptuje żądania, komendy oraz potwierdzenia pochodzące od klienta. Serwer wysyła także odpowiedzi na żądania, a gdy zostanie skonfigurowany przez wyższą warstwę, wysyła asynchronicznie powiadomienia do klienta, kiedy występują na nim określone zdarzenia.

3.7.2 GATT

Komunikacja aplikacji z opaską odbywa się przy wykorzystaniu GATT (Generic Attribute Profile). Jest to technologia zbudowana na protokole Attribute (ATT), która ustala przebieg powszechnych operacji oraz framework dla danych transportowanych przez ATT. GATT określa format danych przesyłanych przez protokół Attribute. Atrybuty są formatowane jako Usługi oraz Charakterystyki. Usługi są zbiorem danych oraz przypisanych im zachowań niezbędnych do zapewnienia określonej funkcji urządzenia. Z kolei Charakterystyki są wartościami użytymi w Usłudze wraz z ich właściwościami oraz informacjami o tym jak są wyświetlane bądź reprezentowane. Dzięki wykorzystaniu określonej struktury danych przez GATT możliwe jest przeglądanie dostępnych Usług oraz Charakterystyk, nawet gdy klient nie jest wyspecjalizowany pod dany serwer.

3.7.3 Komunikacja z MiBand 3

3.8 Opis algorytmów

W tej sekcji należy wymienić i przedyskutować algorytmy wykorzystywane w systemie. Algorytmy należy przedstawić w pseudokodzie (wykorzystać pakiet `algorithm2e`). Omówienia poszczególnych kroków algorytmów powinny zawierać odwołania do odpowiednich linii pseudokodu. Dla zaproponowanych autorskich algorytmów należy przeprowadzić analizę ich złożoności czasowej i pamięciowej.

Implementacja aplikacji

4.1 Opis technologii

Do implementacji systemu został użyty język Kotlin w wersji 1.3.61. Interfejs graficzny zaprojektowano w oparciu o komponenty pochodzące z biblioteki AndroidX oraz Material. Do nawigacji w głównej aktywności aplikacji wykorzystano NavigationUI. Do implementacji bazy danych użyto biblioteki Room zapewniającej poziom abstrakcji nad SQLite. Wykorzystano bibliotekę Hilt/Dagger w celu wstrzykiwania dependencji w obrębie aplikacji w celu zredukowania tak zwanego "boilerplate code".

W systemie wykorzystano inteligentną opaskę Xiaomi MiBand 3. Komunikacja z nią została zaimplementowana na podstawie nieoficjalnego SDK opartego o bibliotekę RxAndroid pozwalającą tworzyć asynchroniczne programy bazujące na wydarzeniach korzystając z obserwowalnych sekwencji.

4.2 Omówienie wybranych kodów źródłowych

Instrukcja obsługi

W tym rozdziale należy omówić zawartość pakietu instalacyjnego oraz założenia co do środowiska, w którym realizowany system będzie instalowany. Należy przedstawić procedurę instalacji i wdrożenia systemu. Czynności instalacyjne powinny być szczegółowo rozpisane na kroki. Procedura wdrożenia powinna obejmować konfigurację platformy sprzętowej, OS (np. konfiguracje niezbędnych sterowników) oraz konfigurację wdrażanego systemu, m.in. tworzenia niezbędnych kont użytkowników. Procedura instalacji powinna prowadzić od stanu, w którym nie są zainstalowane żadne składniki systemu, do stanu w którym system jest gotowy do pracy i oczekuje na akcje typowego użytkownika.

5.1 Instalacja i konfiguracja

W tej sekcji omówię wymagania środowiskowe aplikacji oraz kwestie konfiguracyjne jak parowanie opaski czy ustawienie hasła.

5.1.1 Wymagania sprzętowe

System android 9; moduł BT; opaska MiBand 3;

5.1.2 Pierwsze uruchomienie

- dodać screeny do punktów 1. Gdy aplikacja jest uruchamiana po raz pierwszy należy ustawić jej w ustawieniach "Usage stats permission". Jeśli przy uruchomieniu nie ma tego pozwolenia aplikacja przekieruje w odpowiednie miejsce w ustawieniach jednak będzie to wymagało ponownego uruchomienia aplikacji. 2. Następnie jeśli aplikacja jest uruchamiana z aktywnym "Usage stats permission"to użytkownik jest proszony o pozwolenie na lokalizację. Jest ono niezbędne do poprawnego działania BT. 3. Kiedy uzyskane zostaną wszystkie potrzebne pozwolenia użytkownik zostaje przeniesiony do aktywności, w której jest tworzone hasło, które będzie wykorzystywane przy odblokowywaniu trybu Łockdown". 4. Użytkownik wprowadza dwa razy hasło i dotyka przycisk (drzwi ze strzałką) 5. Po utworzeniu hasła użytkownik zostaje przekierowany do aktywności odpowiadającej za parowanie opaski MiBand 3. 6. Użytkownik klika przycisk scan for devices 7. Następnie uruchamiany jest skan urządzeń ble z filtrem na mi band 3 8. Jeśli urządzenie zostanie odnalezione pojawi się na ekranie. Jeżeli nie zostanie odnalezione należy powtórzyć skan dotykając przycisk scan for devices 9. Po naciśnięciu na odpowiednie urządzenie na liście znalezionych urządzeń zostaje uruchomiona usługa odpowiadająca za komunikację z MiBand i użytkownik przechodzi do głównego widoku aplikacji 10. Usługa MiBandService łączy się z urządzeniem i uruchamia sekwencję inicjującą urządzenie 11. Po udanej inicjacji urządzenia opaska zaczyna przesyłać informacje o pulsie, krokach, zaśnięciu oraz zdjęciu opaski.

5.2 Przykłady użycia

W tej sekcji przedstawię jak działa aplikacja od strony użytkownika poprzez opis czynności potrzebnych do wykonania określonych zadań, np. sprawdzenie stanu opaski, zmiana blokowanych aplikacji czy sprawdzenie statystyk.

Podsumowanie

W podsumowanie należy określić stan zakończonych prac projektowych i implementacyjnych. Zaznaczyć, które z zakładanych funkcjonalności systemu udało się zrealizować. Omówić aspekty pielęgnacji systemu w środowisku wdrożeniowym. Wskazać dalsze możliwe kierunki rozwoju systemu, np. dodawanie nowych komponentów realizujących nowe funkcje.

W podsumowaniu należy podkreślić nowatorskie rozwiązania zastosowane w projekcie i implementacji (niebanalne algorytmy, nowe technologie, itp.).

Propozycje rozwoju w przyszłości:

- Wyłączenie łączności wifi i danych mobilnych podczas blokowania (niemożliwe jeśli nie masz roota bądź systemowej aplikacji albo korzystasz z android 10+)
- Wsparcie dla większej ilości urządzeń
- Dokładniejsze statystyki aktywności
- Maskowanie aplikacji - zmodyfikowanie wyglądu i komunikatów by przypominała zwykłą aplikację towarzyszącą opasce

Bibliografia

[1] F. Almenárez-Mendoza, L. Alonso, A. Marín-López, P. Cabarcos. Assessment of fitness tracker security: A case of study. *Proceedings*, 2:1235, 10 2018.

[2] Y. Cifuentes, L. Beltrán, L. Ramírez. Analysis of security vulnerabilities for mobile health applications. *International Journal of Health and Medical Engineering*, 9(9):1067 – 1072, 2015.

[3] J. Clement. Share of global mobile website traffic 2015-2021. https://www.statista.com/statistics/277125/share-of-website-traffic-coming-from-mobile-devices/.

[4] E. De Cristofaro, H. Du, J. Freudiger, G. Norcie. Two-factor or not two-factor? a comparative usability study of two-factor authentication. *USEC*, 09 2013.

[5] Freedom of the Press Foundation, Guardian Project. Haven: Keep watch. Strona internetowa projektu https://guardianproject.github.io/haven/.

[6] Google. Android Management API. Uzyskano z poradnika w dokumentacji Android Management API https://developers.google.com/android/management/introduction.

[7] Google. Android Management API Policies. Uzyskano z dokumentacji Android Management API https://developers.google.com/android/management/reference/rest/v1/enterprises.policies.

[8] Google. Background Service Limitations. Uzyskano z opisu wersji Androida Oreo https://developer.android.com/about/versions/oreo/background.html#services.

[9] Google. Optimize for Doze and App Standby. Uzyskano z poradnika w dokumentacji Androida https://developer.android.com/training/monitoring-device-state/doze-standby.

[10] Google. Wake Lock. Uzyskano z dokumentacji Androida https://developer.android.com/reference/kotlin/android/os/PowerManager.WakeLock.

[11] S. O'Dea. Number of smartphone users worldwide from 2016 to 2026. https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/.

[12] J. Rutkowska. Evil maid goes after truecrypt! Uzyskano z bloga autorki https://blog.invisiblethings.org/2009/10/15/evil-maid-goes-after-truecrypt.html.

[13] H. Shen, C. Faklaris, H. Jin, L. Dabbish, J. I. Hong. 'i can't even buy apples if i don't use mobile pay?': When mobile payments become infrastructural in china. *Proc. ACM Hum.-Comput. Interact.*, 4(CSCW2), Paz. 2020.

[14] Yubico Inc. Yubikey 5 nfc. Uzyskano ze strony producenta 08.06.2021 https://www.yubico.com/pl/product/yubikey-5-nfc/.

[15] Yubico Inc. Yubikey for mobile. Uzyskano ze strony producenta https://resources.yubico.com/53ZDUYE6/as/q4bsus-2mej80-29grce/YubiKey˙for˙Mobile˙Solution˙Brief.pdf.

Zawartość płyty CD

W tym rozdziale należy krótko omówić zawartość dołączonej płyty CD.