Spis streści:

1.**NFC**- Near Field Communication (komunikacja bliskiego zasięgu) - to technologia bezprzewodowej komunikacji krótkiego zasięgu, ułatwiająca wymianę danych pomiędzy urządzeniami. NFC jest przede wszystkim nakierowane na wykorzystanie w telefonach komórkowych. Podstawowym zadaniem systemu jest umożliwienie dokonywania płatności bezdotykowych, ale stosowany jest również w procesie wymiany informacji i treści multimedialnych. NFC to sposób na ułatwienie wielu codziennych czynności. Wystarczy zbliżyć telefon do czytnika NFC, aby pobrać informacje czy multimedia, zakupić bilet na koncert, zapłacić za przejazd komunikacją miejską, parking czy przekąskę z automatu. W przeciwieństwie do innych obecnych na rynku płatności mobilnych, transakcje NFC nie wymagają weryfikacji i potwierdzenia poprzez wprowadzenie kodu PIN, co usprawnia dokonywanie płatności, zwiększa wygodę i przyspiesza usługę.

- Zasięg: <20 cm
- Częstotliwość: 13.56 MHz ± 7 kHz (Pasmo ISM)
- Maksymalna szerokość pasma sygnału: 2 MHz
- Przepustowość: 106, 212, 424 lub 848 kbit/s
- Tryb pracy:
- Tryb pasywny: Inicjujące urządzenie wytwarza pole elektromagnetyczne, a docelowe urządzenia odpowiada modulując to pole. W trybie tym urządzenie docelowe jest zasilane mocą pola elektromagnetycznego urządzenia inicjującego, dzięki czemu urządzenie docelowe działa jako transponder.
- Tryb aktywny: Oba urządzenia: inicjujące i docelowe komunikują się przez naprzemienne generowania swojego sygnału. Urządzenie wyłącza swoje pole elektromagnetyczne, gdy czeka na dane. W tym trybie oba urządzenia zwykle potrzebują zasilania.
- Odbiór i nadawanie w tym samym czasie

2**. NFC tagi:**

NFC Tags to urządzenia pasywne, które służą do komunikacji z czytnikami i odbiornikami w standardzie NFC. Mogą mieć formę np. plastikowych breloczków, zawieszek, opasek lub naklejek. Zawarty w NFC Tags chip zawiera dane, które, jeżeli zbliżyć do niego telefon lub inny odbiornik, przesyła je do niego. Dane te mogą zawierać wprowadzone wcześniej profile lub ustawienia (np. w telefonie) lub informacje (np. przy plakacie) .

Ilość zastosowań NFC Tags jest ograniczona tylko przez naszą fantazję. Wyobraźmy sobie przyszłość: wsiadamy do samochodu zbliżamy telefon do Tagu wiszącego przy np. lusterku – Tag aktywuje profil, włącza się nam GPS i/lub nawigacja. Wchodzimy do pracy, aktywujemy Tag przy biurku – włącza nam się ‘tryb głośny w telefonie”, na wyświetlaczu pojawiają się wszystkie spotkania na dziś. Jesteśmy w sklepie np. w księgarni, machamy telefonem w pobliżu Tagu i już mamy recenzję książki , informację o autorze i spis jego twórczości. Jesteśmy na wycieczce, z Tagu przyklejonego do punktu informacyjnego sczytujemy plan miasta, główne atrakcje, restauracje i hotele, kontakt do ambasady, rezydenta lub konsula – i to wszystko w języku polskim. Widzimy interesujący nas plakat kinowy lub teatralny – dzięki Tagowi wiemy już kiedy i gdzie jest wyświetlany interesujący nas film , znamy jego recenzję i zarys fabuły. Idziemy spać na koniec bogatego w doświadczenia dnia przykładamy telefon do Tagu, który wisi przy lampce nocnej – telefon zmienia profil na „cichy”, wyłącza Bluetooth oraz WiFi i ustawia budzik.

Jeśli chodzi o ustawienia telefonu to korzystając z takich aplikacji jak NFC Action Launcher można zaprogramować NFC Tags do przesyłania następujących poleceń i ich kombinacji:

- włączyć/wyłączyć WiFi
- włączyć/wyłączyć Bluetooth
- uruchomić każdą zainstalowaną aplikację
- połączyć z dowolną dostępną siecią
- ustanowić nowe połączenie WiFi
- skonfigurować i włączyć Portable Hotspot
- włączyć/wyłączyć auto-synchronizację
- zmienić dzwonek telefonu, budzika
- zmienić profil
- zmienić głośność i wibrację
- zmienić sygnał wiadomości, powiadomień

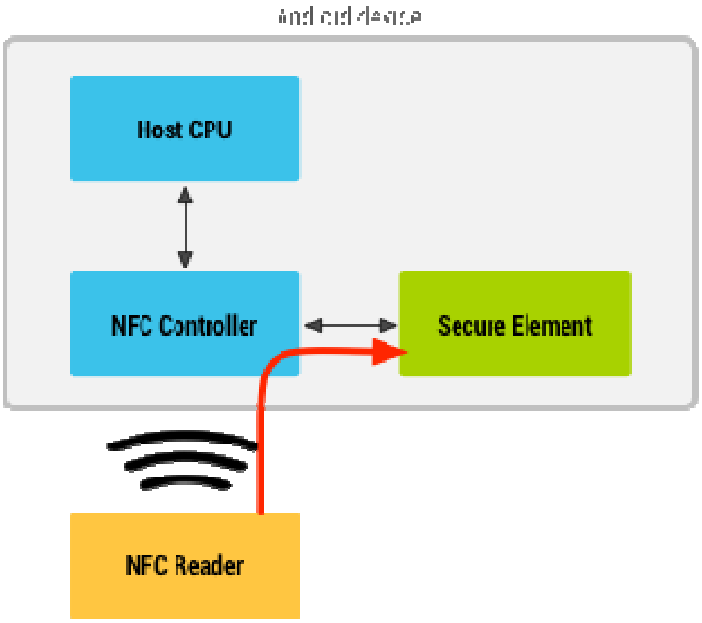
Obecnie są cztery rodzaje NFC Tags. Są im przyporządkowane numery 1-4. Każdy ma inny format i pojemność.
NFC Tag Typ 1: W standardzie ISO 14443A. Są to Tagi do odczytu, mogą być przeformatowane i zapisane na nowo. Maja 96 bajtow do 2 kilobajtów pamięci. Prędkość przesyłu danych: 106 kbit/s. Ze względu na prostotę ten typ Tagu jest idealny do wykorzystania w zużywalnych zastosowaniach , np. przy plakatach.
NFC Tag Typ 2: W standardzie ISO 14443A. Jak Typ 1 są to Tagi do odczytu, mogą być przeformatowane i zapisane na nowo. Pojemność tego typu to 48 bajtów do 2 kilobajtów. Prędkość przesyłu danych: 106 kbit/s.
NFC Tag Typ 3: Oparty jest na systemie Sony FeliCa (w standardzie ISO 18092). Pojemność pamięci: 2 kilobajty . Prędkość przesyłu danych: 212 kbit/s. Ten Typ jest lepszy w porównaniu z poprzednimi do bardziej zaawansowanych funkcji związanych z NFC. Przez to jest jednak droższy.
NFC Tag Typ 4: W standardzie ISO 14443A oraz ISO 14443B. Konfigurowane przy produkcji mogą być wyłącznie do odczytu lub do zapisania na nowo. Pojemność pamięci: 32 kilobajty. Prędkość przesyłu danych: od 106 do 424 kbit/s.

3.**Aplikacje NFC na androida:**
Mobi-NFC:
Platforma mobi-NFC umożliwia:
• łatwe korzystanie z ofert specjalnych i promocji,
• szybki dostęp do informacji interesujących użytkownika,
• wygodne zapisywanie potrzebnych danych oraz korzystanie z lokalnych ułatwień w punkcie odczytu znacznika
Aplikacja umożliwia korzystanie z takich funkcji, jak:
• automatyczne uruchomienie strony www,
• udział w ankietach i konkursach,
• pobieranie kodów promocyjnych, rabatowych, korzystanie z ofert specjalnych,
• odczyt informacji tekstowej,
• odczyt i dodanie do kalendarza elektronicznego informacji o wydarzeniu,
• włączenie nawigacji, która poprowadzi optymalną drogą pod wskazany przez znacznik adres,
• automatyczne przygotowanie telefonu do następujących działań:
• wybranie numeru telefonicznego,
• wysłanie SMS,
• wysłanie wiadomości e-mail,
• dodanie danych osoby lub firmy do kontaktów elektronicznych,
• automatyczną zmianę ustawień telefonu:
• przełączanie i konfigurowanie parametrów sieci WiFi
• przełącznie komunikacji Bluetooth
Aby skorzystać z platformy mobi-NFC, wystarczy zainstalować aplikację w smartfonie. Nie wymaga ona logowania ani podawania żadnych danych czy informacji osobistych. Korzystanie z aplikacji jest anonimowe.

4.**Aplikacje NFC na Windows Phone**:Nokia NFF Writer: Aplikacja pozwala na proste tworzenie tagów NFC. Posiada wbudowany, predefiniowany zestaw tagów, dzięki którym możemy „zaprogramować" specjalnie tagi NFC. Aby stworzyć tag wystarczy z głównego menu wybrać *Compose* a następnie wybrać jedną z dostępnych kategorii: *contact*, *social*, *video+music*, *location+web*, *apps* oraz *settings*. W ten sposób w łatwy i szybki sposób stworzymy tagi, za pomocą których wykonamy telefon, wyślemy wiadomość, otworzymy stronę internetową, uruchomimy sklep Windows Phone, przejdziemy do ustawień telefonu i wiele więcej

5.**NFC-emulacja karty-bezpieczny element**
Wiele urządzeń z systemem Android oferuje funkcjonalność NFC jednocześnie wspiera emulacje kart NFC. W większości przypadków karta jest emulowana przez osobny chip w urządzeniu nazywany elementem bezpieczeństwa. Wiele kart SIM dostarczanych przez operatorów również zawierają element bezpieczeństwa.
Android 4.4 wprowadza dodatkową metodę emulacji kart która nie wymaga bezpiecznego elementu zwanego emulacja karty host-based. To pozwala dowolnej aplikacji android emulacje na komunikacje z czytnikiem NFC. Dokument ten opisuje jak emulacja karty host-based (HCE) działa na Androidzie i jak można tworzyć aplikacje które używają techniki NFC.
Emulacja karty – bezpieczny element
Kiedy emulacja kart NFC jest dostarczana poprzez użycie bezpiecznego elementu, emulowana karta jest zabezpieczona w bezpiecznym elemencie w urządzeniu przez

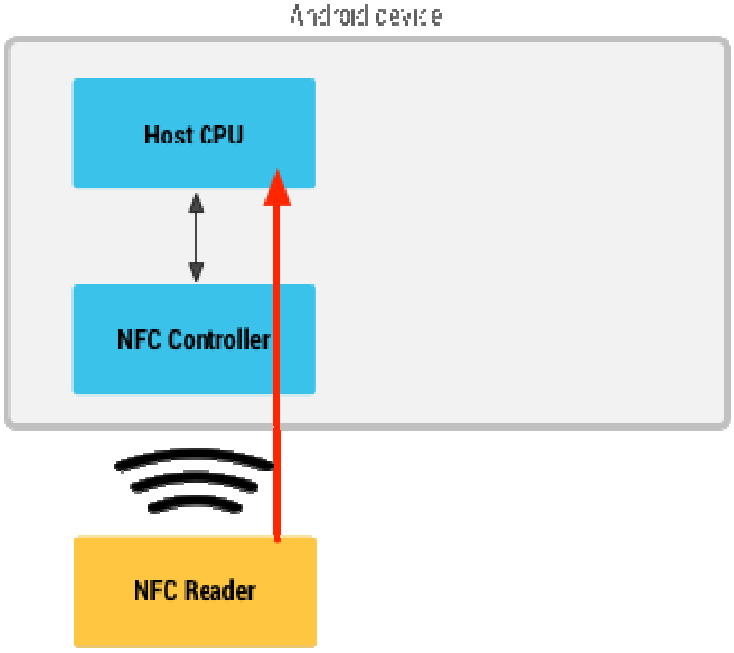
aplikacje Android. Następnie kiedy użytkownik trzyma uprzedzenie nad terminalem NFC kontroler NFC w urządzeniu kieruje dane z czytnika prosto do bezpiecznego elementu. Rysunek przedstawia tę koncepcję



Sam bezpieczny element realizuje komunikacje z terminalem NFC, a aplikacja Android nie jest zaangażowana w transakcje. Po zakończeniu transakcji aplikacja Android może zapytać bezpieczny element o status transakcji i poinformować użytkownika.

Host-based emulacji karty

Kiedy karta jest emulowana używając emulacji karty host-based dane są kierowane do hosta CU na którym aplikacje Android działają bezpośrednio w przeciwieństwie do trasowania ramek protokółu NFC do bezpiecznego elementu. Rysunek przestawia jak działa emulacja karty host-based

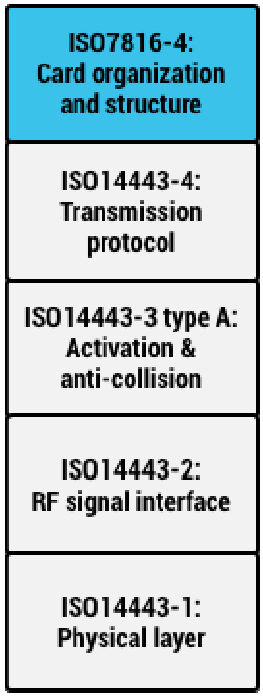


6.**Obsługiwane karty NFC i protokoły**

Normy NFC oferują wsparcie dla wielu rożnych protokołów i są różne typy kart, które mogą być emulowane.

Android 4.4 wspiera kilka protokołów, które są powszechne na dzisiejszym rynku. Wiele istniejących kart bezkontaktowych jednocześnie bazuje na tych protokołach takich jak bezstykowe karty płatnicze. Protokoły te są również wspierane przez wiele czytników NFC w sklepach włączając w to urządzenia NFC Android funkcjonujące jako czytniki ich samych. To pozwala budować rozwiązania NFC.

Android 4.4 wspiera emulowanie kart które bazują na specyfikacji ISO-DEP (w oparciu o ISO / IEC 14443-4) i proces Application Protocol Data Units (APDUs), jak określono w 7816-4 specyfikacją ISO / IEC. Wsparcie dla technologii NFC-B (ISO/IEC 14443-4 Type B) jest opcjonalne. Uwarstwienie tych wszystkich specyfikacji jest pokazane na rysunku



7.**HCE usługi**

Architektura HCE w Androidzie bazuje wokół komponentów serwisów Androida znanych jako "HCE services". Jedna z kluczowych zalet tego serwisu jest to ze to może działać w tle bez interfejsu użytkownika. Jest to naturalne dopasowanie dla wielu aplikacji HCE jako lojalność kart tranzytowych których użytkownik nie potrzebuje do włączenia aplikacji aby tego użyć. Zamiast dotykać urządzenia czytnik NFC uruchamia odpowiedni serwis (jeśli juz nie jest uruchomiony) i wykonuje transakcje w tle. Oczywiście można uruchamiać dodatkowe interfejsy użytkownika takie jak powiadomienia użytkownika z serwis.

Selekcja serwisów.

Kiedy użytkownik dotknie urządzeniem czytnik NFC system Android musi wiedzieć z którym serwisem HCE czytnik NFC chce sie komunikować. Z pomocą przychodzi specyfikacja ISO/IEC 7816-4. Definiuje to drogę wyboru aplikacji skoncentrowanej na Androida(AID).

AID składa sie z 16 bajtów. Jeśli emulujesz kartę dla istniejącej infrastruktury NFC , AID którego te czytniki szukają są dobrze znane i publicznie zarejestrowane np VISA lub MASTERCARD). Jeśli chcemy wyprodukować mową intrastrukturę czytnika dla własnej aplikacji trzeba zarejestrować własny AID. Procedura rejestracji jest zdefiniowana w specyfikacji ISO/IEC 7816-5 . Google zaleca rejestrowanie AID-ow jako 7816-5 jeśli tworzysz aplikacje HE dla androida co zapobiegnie kolizji z innymi aplikacjami z grupy AID.

8.**Implementacja serwisu HCE.**

Aby emulować kartę NFC używając emulacji karty host-based trzeba stworzyć komponent serwisu który przechwytuje transakcje NFC.

Sprawdzanie wsparcia dla HCE.

Aplikacja może sprawdzać kiedy urządzenie wspiera HCE sprawdzając FEATURE_NFC_HOST_CARD_EMULATION. Powinno używać Tagu: <uses-feature> w manifeście aplikacji aby zadeklarować używanie przez aplikacje HCE i czy jest to potrzebne dla funkcjonowania aplikacji czy nie.

Implementacja serwisu.

Android 4.4 przychodzi z konwencją Service class która może być użyta jako podstawy implementacji serwisu HCE: hostapduservice class. Dlatego pierwszym krokiem jest rozszerzenie" HostApduService"

```
public class MyHostApduService extends HostApduService {
   @Override
   public byte[] processCommandApdu(byte[] apdu, Bundle extras) {
     ...
   }
   @Override
   public void onDeactivated(int reason) {
     ...
   }
}
```

HostApduService deklaruje dwie metody abstrakcyjne które muszą być nadpisane i zaimplementowane.
processCommandApdu() jest wywołane gdy czytnik NFC wysyła Application Protocol Data Unit (APDU) do serwisu. APDUs jest zdefiniowane w specyfikacji ISO/IEC 7816-4 APDU są paczkami na poziomie aplikacji które są wymieniane miedzy czytnikiem NFC a serwisem HCE. Czytnik NFC wysyła komendę APDU i czeka na odpowiedz od APDU.
Specyfikacja ISO/IEC 7816-4 również definiuje koncepcje wielu logicznych kanałów gdzie możesz mieć wiele równoległych APDU wymienianych w osobnych kanałach logicznych. HCE implementacji Androida wpiera pojedyncze kanały logiczne wiec jest jednowątkowa wymiana APDU.
Android korzysta z pomocy w celu określeniu które usługi HCE chce użyć. Zwykle najpierw APDU czytnik NFC wysyła do urządzenia "SELECT AID". APDU zawiera AID z którym czytnikiem chce komunikować się. Android wypakowuje AID z APDU rozpoznając to przez serwis HE a następnie APDU dla rozpoznanego serwisu.
Można wysyłać odpowiedz APDU zwracając bajty odpowiedzi APDU z processCommandApdu(). Ta metoda będzie wywołana w głównym wątku aplikacji który nie może być zablokowany. Więc jeśli nie można przetworzyć i zwrócic odpowiedzi APDU nic nie zwracaj. Można wtedy wykonać niezbędną prace w innym wątku i użyć metody sendResponseApdu() zdefiniowanej w klasie HostApduService wysyłając odpowiedz kiedy skończymy.
Android będzie utrzymywać spedycje nowych APDUs z czytnika do usług:
czytnik NFC wysyła inne "select aid" APDU które system operacyjny rozpoznaje jako inny serwis.
Link NFC miedzy czytnikiem NFC a twoim urządzeniem jest zniszczony.
W obu tych przypadkach implementacja onDeactivated() w klasie jest wywoływana z argumentem wskazującym który z tych obu przypadków wystąpił.
Jeśli chcesz pracować z istniejąca infrastruktura czytnika musisz zaimplementować istniejący protokół na poziomie aplikacji którego czytnik będzie oczekiwał w twoim serwisie HCE.
Jeśli tworzysz nowa infrastrukturę czytnika którą kontrolujesz możesz zdefiniować swój własny protokół i sekwencje APDU. Generalnie limitując liczebność APDU i rozmiar danych, które muszą być wymieniane, to daje pewność ze użytkownik musi tylko trzymać urządzenie nad czytnikiem NFC przez krotki okres czasu. Rozsądne górne granice to ok 1 KB, które zostaną wymienione w ciągu 300 ms.

9.**Obsługa i rejestracja deklaracji manifestu**

Usługa musi być zadeklarowana w manifeście jak zwykle, ale należy niektóre dodatkowe części dodać do deklaracji serwisu. Po pierwsze, aby zakomunikować , że platforma jest serwisem HCE trzeba zaimplementować interfejs HostApduService , deklaracja usług musi zawierać filtr przeznaczony do działania SERVICE_INTERFACE . Dodatkowo , aby zakomunikoiwać platformie , które grupy AIDS są wymagane przez ten serwis ,tag <meta-data> SERVICE_META_DATA musi być zawarty w deklaracji usług , wskazując na źródło XML z dodatkowymi informacjami na temat usługi HCE . Wreszcie , należy ustawić atrybut Android : exported na TRUE , i wymagać ”android.permission.BIND_NFC_SERVICE " w swojej deklaracji serwisu . Ten ostatni zapewnia ze tylko zewnętrzne aplikacje które trzymają "android.permission.BIND_NFC_SERVICE" mogą być bindowane do serwisu od kiedy "android.permission.BIND_NFC_SERVICE" jest pozwoleniem systemu. Efektywnie zapewnia to ze system operacyjny Android może być bindowany do serwisu. Przykładowa deklaracja manifestu :

<service android:name=".MyHostApduService" android:exported="true" android:permission="android.permission.BIND_NFC_SERVICE"> <intent-filter> <action android:name="android.nfc.cardemulation.action.HOST_APDU_SERVICE"/> </intent-filter> <meta-data android:name="android.nfc.cardemulation.host_apdu_service" android:resource="@xml/apduservice"/> </service>

Ten meta-data tag odwołuje się do apduservice.xml file. Przykład jak tylko plik z pojedyncza deklaracja grupy AID zawierającą dwie własności :

<host-apdu-service xmlns:android="http://schemas.android.com/apk/res/android" android:description="@string/servicedesc" android:requireDeviceUnlock="false"> <aid-group android:description="@string/aiddescription" android:category="other"> <aid-filter android:name="F0010203040506"/> <aid-filter android:name="F0394148148100"/> </aid-group> </host-apdu-service>

<host-adpu-service> tag powinien zawierać atrybut <android:description> który zawiera przyjazny użytkownikowi opis tego serwisu, który może być wyświetlony w interfejsie użytkownika.

RequireDeviceUnloc atrybut może być użyty dla wyjaśnienia ze urządzenie musi być odblokowane zanim serwis będzie mógł odbierać APDU.

<host-apdu-service> musi zawierać jeden lub więcej <aid-group> tagow. Każdy <aid-group> tag jest wymagany do:

zawierania atrybutu android:description który zawiera przyjazny użytkownikowi opis mogący byc wyświetlony w interfejsie użytkownika. AID musi być wyspecyfikowany w heksadecymalnym formacie i zwierać nawet pewna liczbę znaków.

10.**Płatności**

Android uważa serwisy HCE, które deklarują grupę AID z kategorii "payment" jako aplikacje płatnicze. Wersja android 4.4 zawiera menu opcji wyższego poziomu zwane "tap&pay", które wyliczają wszystkie takie aplikacje płatnicze. W menu opcji użytkownik może wybrać różne aplikacje płatnicze które będą wywołane jak terminal płatniczy zostanie przyłożony. Wymagane wnioski dla aplikacji płatniczych.

W celu zapewnienia bardziej atrakcyjnego wizualnie interfejsu użytkownika, aplikacje płatnicze HCE wymagają dostarczenia dodatkowych wniosków dla ich serwisu tzw banner serwis. Wniosek powinien mieć 260x96 DP i powinien być wyspecyfikowany w meta-danych xml dodając atrybut android:apduServiceBanner do tagu <host-apdu-service>

```
<host-apdu-service  xmlns:android = "http://schemas.android.com/apk/res/android"
      android:description = "@string/servicedesc"
      android:requireDeviceUnlock = "false"
      android:apduServiceBanner = "@drawable/my_banner" >
   <aid-group  android:description = "@string/aiddescription"
            android:category = "payment" >
      <aid-filter  android:name = "F0010203040506" />
      <aid-filter  android:name = "F0394148148100" />
   </aid-group>
</host-apdu-service>
```

11.**Blokada ekranu wyłączony ekran i Zachowanie**

Aktualna implementacja Androida wyłącza kontroler NFC i procesor aplikacji kiedy ekran urządzenia jest wyłączony. Serwisy HCE nie będą działać jeśli ekran jest wyłączony. Serwisy HCE mogą funkcjonować z zablokowanego ekranu jednak jest to kontrolowane przez atrybut android:requireDeviceUnlock w tagu <host-apdu-service> serwisu HCE. Domyślnie odblokowywanie urządzenia nie jest wymagane i serwis sie wykona nawet kiedy urządzenie jest zablokowane. Jeśli ustawimy atrybut android:requireDeviceUnlock na TRUE dla serwisu HCE Android wymusi na użytkowniku odblokowanie urządzenia kiedy przyłoży je do czytnika NFC .Po odblokowaniu Android wyświetli okno dialogowe informujące o wymaganiu przyłożenia jeszcze raz aby dokończyć transakcje, jest to wymagane ponieważ użytkownik mógł wziąść urządzenie z czytnika w celu odblokowania