



**HoGent**

Faculteit Bedrijf en Organisatie

Verskillende veiligheidsmethodes voor Android HCE (Host-based Card Emulation)

Thibaut Maddelein

Scriptie voorgedragen tot het bekomen van de graad van  
professionele bachelor in de toegepaste informatica

Promotor:  
Leen Vuyge  
Co-promotor:  
Francesco Verheye

Instelling: CCV Lab

Academiejaar: 2018-2019

Tweede examenperiode



Faculteit Bedrijf en Organisatie

Verskillende veiligheidsmethodes voor Android HCE (Host-based Card Emulation)

Thibaut Maddelein

Scriptie voorgedragen tot het bekomen van de graad van  
professionele bachelor in de toegepaste informatica

Promotor:  
Leen Vuyge  
Co-promotor:  
Francesco Verheye

Instelling: CCV Lab

Academiejaar: 2018-2019

Tweede examenperiode



## Woord vooraf



## Samenvatting

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus.

Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.



# Inhoudsopgave

<b>1</b>	<b>Inleiding .....</b>	<b>13</b>
1.1	Probleemstelling	13
1.2	Onderzoeksvraag	14
1.3	Onderzoeksdoelstelling	14
1.4	Opzet van deze bachelorproef	14
<b>2</b>	<b>Stand van zaken .....</b>	<b>15</b>
2.1	Host-based Card Emulation (HCE)	15
2.2	Near Field Communication (NFC)	16
2.3	Secure Element (SE)	17
2.4	Beveiliging HCE	17
2.5	Beveiliging op hardware niveau	18
2.5.1	Biometric Factors .....	18

2.5.2	Secure Element .....	18
<b>2.6</b>	<b>Beveiliging op software niveau</b>	<b>19</b>
2.6.1	White Box Cryptography .....	19
2.6.2	Tamper-Proofed Software .....	19
2.6.3	Device Identity Solutions .....	19
2.6.4	Encryption .....	20
2.6.5	Tokenization .....	20
<b>2.7</b>	<b>Beveiliging op software en hardware niveau</b>	<b>20</b>
2.7.1	Security Frameworks/Trusted Execution Environment .....	20
<b>3</b>	<b>Methodologie .....</b>	<b>23</b>
<b>3.1</b>	<b>Beveiligingsmethodes</b>	<b>23</b>
<b>3.2</b>	<b>Opbouw Host-based Card Emulation applicatie</b>	<b>24</b>
<b>4</b>	<b>Conclusie .....</b>	<b>25</b>
<b>A</b>	<b>Onderzoeksvoorstel .....</b>	<b>27</b>
<b>A.1</b>	<b>Introductie</b>	<b>27</b>
<b>A.2</b>	<b>State-of-the-art</b>	<b>27</b>
<b>A.3</b>	<b>Methodologie</b>	<b>28</b>
<b>A.4</b>	<b>Verwachte resultaten</b>	<b>28</b>
<b>A.5</b>	<b>Verwachte conclusies</b>	<b>28</b>
	<b>Bibliografie .....</b>	<b>29</b>

## Lijst van figuren

2.1	Simulatie van een smart card via SE .....	16
3.1	Service meta-data in AndroidManifest .....	24



## Lijst van tabellen



# 1. Inleiding

De inleiding moet de lezer net genoeg informatie verschaffen om het onderwerp te begrijpen en in te zien waarom de onderzoeksvraag de moeite waard is om te onderzoeken. In de inleiding ga je literatuurverwijzingen beperken, zodat de tekst vlot leesbaar blijft. Je kan de inleiding verder onderverdelen in secties als dit de tekst verduidelijkt. Zaken die aan bod kunnen komen in de inleiding (Pollefliet, 2011):

- context, achtergrond
- afbakenen van het onderwerp
- verantwoording van het onderwerp, methodologie
- probleemstelling
- onderzoeksdoelstelling
- onderzoeksvraag
- ...

## 1.1 Probleemstelling

Uit je probleemstelling moet duidelijk zijn dat je onderzoek een meerwaarde heeft voor een concrete doelgroep. De doelgroep moet goed gedefinieerd en afgeleid zijn. Doelgroepen als “bedrijven,” “KMO’s,” systeembeheerders, enz. zijn nog te vaag. Als je een lijstje kan maken van de personen/organisaties die een meerwaarde zullen vinden in deze bachelorproef (dit is eigenlijk je steekproefkader), dan is dat een indicatie dat de doelgroep goed gedefinieerd is. Dit kan een enkel bedrijf zijn of zelfs één persoon (je co-promotor/opdrachtgever).

## 1.2 Onderzoeksvraag

Wees zo concreet mogelijk bij het formuleren van je onderzoeksvraag. Een onderzoeksvraag is trouwens iets waar nog niemand op dit moment een antwoord heeft (voor zover je kan nagaan). Het opzoeken van bestaande informatie (bv. “welke tools bestaan er voor deze toepassing?”) is dus geen onderzoeksvraag. Je kan de onderzoeksvraag verder specificeren in deelvragen. Bv. als je onderzoek gaat over performantiemetingen, dan

## 1.3 Onderzoeksdoelstelling

Wat is het beoogde resultaat van je bachelorproef? Wat zijn de criteria voor succes? Beschrijf die zo concreet mogelijk.

## 1.4 Opzet van deze bachelorproef

De rest van deze bachelorproef is als volgt opgebouwd:

In Hoofdstuk 2 wordt een overzicht gegeven van de stand van zaken binnen het onderzoeksdomein, op basis van een literatuurstudie.

In Hoofdstuk 3 wordt de methodologie toegelicht en worden de gebruikte onderzoekstechnieken besproken om een antwoord te kunnen formuleren op de onderzoeksvragen.

In Hoofdstuk 4, tenslotte, wordt de conclusie gegeven en een antwoord geformuleerd op de onderzoeksvragen. Daarbij wordt ook een aanzet gegeven voor toekomstig onderzoek binnen dit domein.



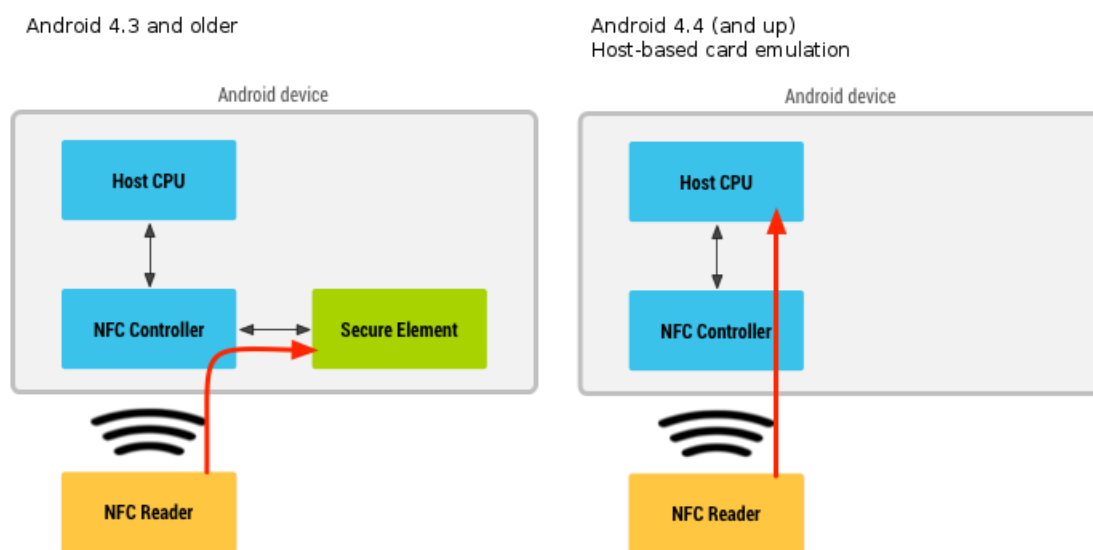
## 2. Stand van zaken

In dit hoofdstuk zal er een stand van zaken gegeven worden van het onderwerp, hoe het vroeger gebeurde en hoe het tegenwoordig gebeurt. Eerst en vooral zal sectie 2.1 wat meer uitleg geven over HCE en hoe het simuleren van draadloze smart cards gebeurt. In sectie 2.2 zal een korte uitleg gegeven worden over NFC technologie. Sectie 2.3 geeft een betere kijk op wat een Secure Element (SE) is. Ten slotte zal in sectie 2.4 uitgelegd worden welke manieren er bestaan om HCE te beveiligen.

### 2.1 Host-based Card Emulation (HCE)

Sinds de komst van Android 4.4 oftewel Android KitKat introduceerde Google een nieuwe technologie genaamd Host-based Card Emulation (HCE). HCE technologie maakt het mogelijk om Near Field Communication (NFC) technologie te gebruiken zonder de aanwezigheid van een secure element. Wanneer het Android toestel zich in card emulation (CE) mode bevindt en tegen een draadloze leer of point-of-sale (POS) terminal gehouden wordt, heeft het toestel de mogelijkheid om allerlei soorten draadloze smart cards te simuleren. Deze contactloze smart cards worden in vele situaties gebruikt zoals bij draadloos betalen, loyalty systemen, ticketing, toegang tot gebouwen ... (Alliance, 2014).

HCE maakt het dus mogelijk voor NFC-apparaten om draadloze smart cards te simuleren. Om gebruik te kunnen maken van HCE heeft Android verschillende libraries en APIs (Application Programming Interface) geïmplementeerd in het besturingssysteem. Deze libraries en APIs worden overschreven door de applicaties die hier gebruik van willen maken en die op de CPU van het apparaat draaien. Deze applicaties kunnen dan APDU (Application Protocol Data Unit) commando's en antwoorden uitwisselen met een NFC



Figuur 2.1: Simulatie van een smart card via SE

POS. Wanneer men vroeger gebruik wilde maken van de NFC technologie kon dit alleen door een Secure Element (SE) die ingebouwd zat in het apparaat zoals een SIM-kaart. De applicaties werden geïnstalleerd op deze SE die dan de APDU's afhandelde om zo draadloze smart cards veilig te kunnen simuleren. De APDU's die verstuurd worden van een NFC-lezer worden opgevangen door de NFC-antenne van het apparaat en wordt doorgegeven via de NFC controller naar het SE en omgekeerd (zie figuur 2.1). Met HCE is het de bedoeling dat de nood van een SE verwijderd wordt uit deze operatie, in plaats van de APDU's door te geven naar het SE worden deze doorgegeven naar de CPU van het apparaat en omgekeerd (zie figuur 2.1) (Alattar, 2014).

## 2.2 Near Field Communication (NFC)

Near Field Communication (NFC) is een technologie die communicatie vanop korte afstanden, meestal nul tot vier centimeter maar kan tot 20 centimeter oplopen, mogelijk maakt. NFC apparaten kunnen actief of passief zijn, wanneer het NFC apparaat actief is dan gebruikt dit apparaat zijn eigen energiebron om zijn radio frequentie te genereren. Een passief NFC apparaat gebruikt de energiebron van een actief NFC apparaat om data te versturen, passieve NFC apparaten kunnen ook enkel maar antwoorden op aanvragen die vanaf een actief NFC apparaat verstuurd wordt. Transacties worden automatisch gestart door twee NFC apparaten elkaar te laten raken of deze twee dicht bij elkaar te houden (Alattar, 2014).

NFC heeft drie verschillende operating modes: Peer to Peer mode, Reader/Writer mode en Contactless Card Emulation. Peer To Peer mode biedt de mogelijkheid om data te versturen tussen twee NFC apparaten aan snelheden tot 424 Kbit/s. Reader/Writer mode maakt het

mogelijk dat twee NFC apparaten gebruikt kunnen worden voor het lezen/schrijven van tags en draadloze smart cards, hierbij is de snelheid van het versturen van de data maar 106 Kbit/s. Contactless Card Emulation laat de NFC apparaten draadloze smart cards of tags simuleren die gelezen of naar geschreven kunnen worden door een NFC lezer (Alattar, 2014).

## 2.3 Secure Element (SE)

Een Secure Element is een hardware element binnenin een Android apparaat. Het doel van het Secure Element is om applicaties en al hun data veilig op te slaan, het Secure Element is hardwarematig beveiligd dus zorgt het voor de ideale plaats om de applicaties in op te slaan. Er bestaan drie verschillende soorten SE's: Universal Integrated Circuit Card (UICC), ingebouwde SE (eSE) en een microSD (Lepojevic, Pavlovic & Radulovic, 2014). UICC is een meer geavanceerde vorm van een gewone SIM-kaart. ESE is een ingebouwde smart card die gesoldeerd is aan het moederbord van het apparaat. Ondanks dat een Secure Element één van de meest veilige manieren is om een applicatie en zijn data in op te slaan is het niet de gemakkelijkste. Niet ieder apparaat is in het bezit van een Secure Element. Dit zorgt er dus voor dat maar een klein aantal apparaten kunnen genieten van de voordelen van een Secure Element, iedere vorm van een Secure Element wordt ook beheerd door een andere producent wat het moeilijker maakt voor applicatieontwikkelaars om het SE te gebruiken. Door deze tekortkomingen kwam de nood voor Host-based Card Emulation technologie en de daarbij horende beveiliging voor HCE.

## 2.4 Beveiliging HCE

Met de komst van Host-base Card Emulation sinds Android 4.4 is er geen nood meer aan een Secure Element. HCE brengt wel een aantal beveiligingsrisico's met zich mee. Bij een simulatie van smart cards via SE wordt de applicatie geïnstalleerd op het Secure Element, aangezien het Secure Element hardwarematig beveiligd is tegen fraude is de applicatie automatisch ook beveiligd tegen veiligheidsbedreigingen. Wanneer er gekozen wordt voor een HCE gebaseerde simulatie wordt de applicatie gewoon op het apparaat geïnstalleerd en is deze niet meer beveiligd tegen bedreigingen van andere applicaties die ook op het apparaat staan. Doordat de communicatie tussen de NFC-controller en de HCE-applicatie niet meer beveiligd is kan de communicatie tussen deze twee onderschept worden door andere malware applicaties. Malware applicaties vormen een groot gevaar voor uw Android toestel, deze applicaties kunnen het besturingssysteem aanvallen. Het risico van een aanval op het besturingssysteem wordt ook vergroot door het Android toestel te rooten. De malware applicatie kan ook de mogelijkheid hebben om het toestel zelf te rooten of de gebruiker in de val te lokken om dit te doen (Alliance, 2014).

De naam rooting of rooten is afkomstig van de term root die ook in de Linux wereld gebruikt wordt. Android is gebaseerd op de kernel van Linux, wanneer de gebruiker in Linux (dus ook in android) root rechten heeft, heeft deze "rootgebruiker" alle rechten om

acties uit te voeren. Het is dus duidelijk dat het niet handig of veilig is wanneer een gebruiker zomaar alle rechten heeft, dit voorkomt dat de gebruiker bewust of onbewust schade kan aanrichten aan het systeem. Één van de meest voorkomende redenen waarom android gebruikers hun toestel rooten is zodat ze alternatieve kernels kunnen installeren. Door de kernel van een toestel te veranderen kan de snelheid en de accuduur van een toestel enorm te verbeteren, via een alternatieve kernel heeft men ook de mogelijkheid om de kloksnelheid van de processor aan te passen, de kleuren van het scherm of de kracht van de GPU (Graphics Processing Unit) aan te passen.

Om deze bedreigingen tegen te gaan biedt de technologie de dag van vandaag een waaier aan mogelijke oplossingen die hiervoor ingezet kunnen worden:

- White box cryptography
- Tamper proofed software
- Biometric factors
- Device identity solutions
- Security frameworks/trusted execution environment
- Encryption
- Tokenization
- Bijkomende beveiliging voorzien door een SE

## 2.5 Beveiliging op hardware niveau

### 2.5.1 Biometric Factors

Biometric factors of biometrische factoren kunnen gebruikt worden bij de authenticatie van een gebruiker in Host-based Card Emulation applicaties. Biometrische factoren worden meestal gebruikt in samenwerking met andere authenticatie middelen. Wat biometrische factoren zo aantrekkelijk maakt bij de authenticatie van een gebruiker is voornamelijk de gebruiksvriendelijkheid, zeker in vergelijking met het bijhouden van meerdere wachtwoorden. Er bestaan verschillende soorten biometrische factoren die gebruikt kunnen worden: vingerafdruk, gezichtsherkenning, irisscan en stemherkenning. Deze biometrische factoren zitten al een tijdje geïntegreerd in de meeste laptops en smartphones en kunnen dus gemakkelijk op applicatie niveau gebruikt worden. Een bijkomend probleem is de privacy en beveiliging omtrent de biometrische data die ook in acht genomen zal moeten worden bij het implementeren van de applicatie (Alliance, 2014).

### 2.5.2 Secure Element

HCE maakt het mogelijk om communicatie met de NFC-controller te laten verlopen zonder een SE, maar direct op de processor van het toestel, HCE specificeert niet waar de data opgeslagen moet worden. De data kan zowel in de cloud als op een SE opgeslagen worden dus een combinatie van de twee is zeker ook mogelijk wanneer een SE aanwezig is in het toestel. Data die op het SE opgeslagen worden zijn beveiligd door het gebruik van

cryptografische sleutels. De beveiliging van een SE kan nog versterkt worden door gebruik te maken van een TEE (zie 2.7.1). Het blijft wel nog altijd veiliger de applicatie en de data enkel op een SE uit te voeren en op te slaan (Alliance, 2014).

## 2.6 Beveiliging op software niveau

### 2.6.1 White Box Cryptography

White box cryptography is een manier om keys die in het geheugen van een toestel zitten of in de code zitten te verbergen voor aanvallers. Bij het ontwikkelen van software worden vaak keys gebruikt voor encryptie maar deze kunnen gemakkelijk achterhaald worden in een applicatie via het doorlopen en analyseren van de code van de applicatie. Via white box cryptography wordt de key als het ware omgevormd naar een stuk onleesbare tekst, dit resultaat wordt verkregen door de key samen te smelten met een cipher tekst via een deterministisch algoritme. De key en de cipher tekst vormen een onleesbaar stuk tekst die niet zomaar achterhaalt kan worden. Een deterministisch algoritme is een algoritme waarbij een bepaalde input altijd voor dezelfde output zorgt. Aangezien de keys niet meer leesbaar zijn is het dus ook niet meer mogelijk om de keys zomaar te achterhalen via het doorlopen van de code van de applicatie.

### 2.6.2 Tamper-Proofed Software

Tamper proofing software is een extra beveiligingslaag in de software die ervoor zorgt dat het moeilijker is voor de aanvaller om de code statisch of dynamisch aan te passen of reverse engineering toe te passen op de code. Dit kan gebeuren op verschillende manieren, runtime integrity checking, breakpoint defenses, obfuscation, anti-debug, ... Wanneer er een aanval wordt gedaan en ook effectief wordt waargenomen door de software zal het tamper-proofed systeem een antwoord produceren waardoor de aanvallende software faalt en de aanval dus vermeden wordt (Alliance, 2014).

### 2.6.3 Device Identity Solutions

Device identity solutions gebruiken online diensten voor authenticatie van een toestel die zorgen voor een extra beveiligingslaag voor de Host-based Card Emulation applicaties. Fast Identity Online (FIDO) is een voorbeeld van zo'n onlinedienst. FIDO maakt gebruik van publieke sleutel cryptografie technieken voor online authenticatie, het toestel van de gebruiker creëert een sleutelpaar waarbij de privésleutel bijgehouden wordt en de publieke sleutel geregistreerd wordt bij de online dienst. Het authenticeren van het toestel bij de online service wordt gedaan door middel van de privésleutel die enkel lokaal ontgrendeld kan worden via biometrische factoren of door het ingeven van een PIN-code. FIDO ondersteunt veel verschillende technologieën die naast elkaar gebruik kunnen worden zoals tokenisatie en one-time-password oplossingen. PayPal was één van de eerste die gebruik heeft gemaakt van vingerafdruk verificatie op de Samsung Galaxy S5 met FIDO Ready

software (Alliance, 2014).

#### 2.6.4 Encryption

Encryptie is een methode die het mogelijk maakt om data onleesbaar te versturen en niet als gewone tekst. Het ophalen van data die in gewone tekst verstuurd wordt is één van de grootste vormen van data breaches in card-present en card-not-present betalingsomgevingen, wanneer kaarten gelezen of de gegevens ingelezen worden in een web-based formulier. Er bestaan twee verschillende soorten encryptie end-to-end encryptie (E2EE) en point-to-point encryptie (P2PE). Deze methodes zorgen ervoor dat de data geëncrypteerd is bij NFC-lezer en het versturen van de data. Voor betalingsapplicaties kan encryptie gecombineerd worden met tokenisatie (zie 2.6.5), hierbij wordt het kaartnummer van een bankkaart geëncrypteerd en dan wordt het geëncrypteerde nummer gebruikt om een token te gaan genereren (Alliance, 2014).

#### 2.6.5 Tokenization

Tokenisatie is een methode die een willekeurige waarde teruggeeft van bijvoorbeeld een kaartnummer, rijksregisternummer, ... Tokenisatie op zich wordt al lang gebruikt, maar heeft de laatste jaren meer aandacht gekregen door het toenemende aantal data breaches. Door het lekken van al die data is er een grote nood aan tokenisatie om betalingsgegevens te beschermen tegen fraude en vervalsing. Verschillende grote spelers binnen de beveiligingsindustrie hebben nieuwe standaarden opgesteld voor tokenisatie, één daarvan is EMVCo. De tokenisatie standaarden die uitgebracht zijn door EMVCo voor betalingen kunnen heel domeinspecifiek zijn en bevat cryptogrammen die HCE use cases kunnen isoleren. Door dit toe te passen kan voorkomen worden dat de tokens gebruikt kunnen worden binnen andere betalingskanalen (Alliance, 2014).

### 2.7 Beveiliging op software en hardware niveau

#### 2.7.1 Security Frameworks/Trusted Execution Environment

Trusted Execution Environment (TEE) is een veilige plaats in de hoofd processor of coprocessor van het toestel waar data kan opgeslagen en verwerkt worden. De bedoeling van een TEE is het uitvoeren van geautoriseerde beveiligingssoftware in een vertrouwde omgeving. TEE bestaat niet enkel uit software maar ook uit hardware die bescherming bieden tegen aanvallen vanuit het besturingssysteem in het toestel. Gevoelige applicaties die beschermd moeten worden tegen het besturingssysteem van het toestel worden opgeslagen in de TEE en helpt ook bij de controle van toegangsrechten tot de applicaties. TEE heeft zijn eigen besturingssysteem, hierdoor kan de TEE niet aangetast worden wanneer het besturingssysteem van het toestel aangetast is. De TEE kan voor een extra beveiligingslaag zorgen voor HCE applicaties (Alliance, 2014):

- **PIN/wachtwoord ingave.** De TEE kan extra bescherming aanbieden aan de hand van het invoeren van een PIN-code of een wachtwoord. Bij de TEE is de invoer van de PIN-code of een wachtwoord volledig gescheiden van de invoer van het toestel, hierdoor kan de invoer niet onderschept worden door malware applicaties die zich op het besturingssysteem van het toestel bevinden.
- **Secure storage of credentials.** De TEE implementeert cryptografische operaties binnen de secure execution environment en zorgt voor het veilig opslaan van sleutels. Hierin kunnen dus tokens/sleutels van betalingsapplicaties in opgeslagen worden, zo zijn deze beter beveiligd tegen aanvallen dan wanneer ze opgeslagen worden in het besturingssysteem van het toestel.
- **Secure transfer protocol endpoint.** Het is mogelijk om in de TEE een geëncrypteerd beveiligd kanaal op te zetten aan de kant van de terminal. Dit betekent dat de APDU's geëncrypteerd verstuurd kunnen worden van de terminal en de HCE applicatie via een draadloze interface naar de TEE. Een tweede geëncrypteerd beveiligd kanaal kan opgezet worden tussen de TEE en een Cloud applicatie. Hierdoor zijn de sleutels en data enkel zichtbaar binnen de vertrouwde applicatie, dit biedt een hoger niveau van beveiliging aan dan wanneer de applicatie uitgevoerd zou worden op het besturingssysteem van het toestel.



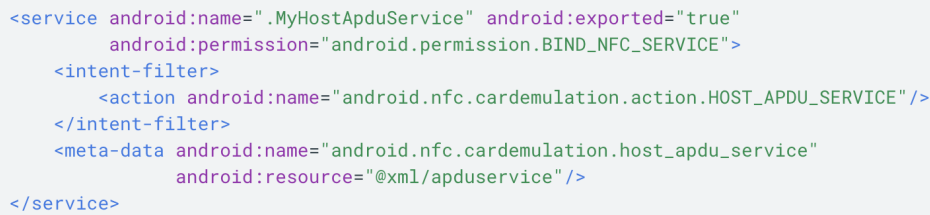


## 3. Methodologie

In dit hoofdstuk zal er meer uitleg gegeven worden over het opstellen van de verschillende proof of concepts en waarom voor bepaalde implementaties gekozen zijn. Hierbij zal ook besproken worden hoe een basis Host-based Card Emulation applicatie opgebouwd wordt.

### 3.1 Beveiligingsmethodes

Voor dit onderzoek werd er gekozen voor drie verschillende beveiligingsmethodes uit 2.4 die verder uitgewerkt zullen worden als proof of concept. De drie gekozen beveiligingsmethodes zijn biometric factors, encryption en tokenization. Door het kiezen voor deze drie beveiligingsmethodes kan er al een mooie vergelijking gemaakt worden tussen één beveiligingsmethode op hardware niveau namelijk biometric factors (in dit geval vingerscan) en twee beveiligingsmethodes op software niveau. Voor het testen van de verschillende gekozen beveiligingsmethodes is er natuurlijk een Android applicatie nodig waarbij het simuleren van een smartcard geïmplementeerd is aan de hand van Host-based Card Emulation. Google heeft een groot aanbod aan voorbeeld applicaties waarin verschillende technieken geïmplementeerd zijn via de best practices, waaronder ook een applicatie waarin HCE geïmplementeerd is die voor dit experiment gebruikt zal worden. In deze standaard applicatie zullen de drie beveiligingsmethodes geïmplementeerd en vergeleken worden.



```
<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>
```

Figuur 3.1: Service meta-data in AndroidManifest

## 3.2 Opbouw Host-based Card Emulation applicatie

Bij het ontwikkelen van een Host-based Card Emulation applicatie moeten er een paar dingen aangepast worden in de AndroidManifest en er moet een Apdu service klasse aangemaakt worden. Wanneer een standaard öne activityapplicatie gemaakt is moeten er twee permissies toegevoegd worden in de Android Manifest, `<uses-feature android:name=android.hardware.nfc.hce android:required="true"/>` en `<uses-permission android:name=android.permission.NFC/>`, de eerste permissie laat de applicatie toe om hardware HCE technologie te gebruiken en de tweede permissie laat het toe om de NFC technologie van het apparaat te gebruiken. De Apdu service moet ook gedeclareerd worden in de AndroidManifest zie 3.1. Bij de declaratie van de service wordt er toestemming gegeven aan externe applicaties om verbinding te maken met de service aan de hand van `android:exported="true"`, de permissie `android.permission.BIND_NFC_SERVICE` zorgt ervoor dat enkel externe applicatie met deze permissie verbinding kunnen maken met de service.

Bij het implementeren van de apdu service klasse wordt de klasse uitgebreid met de HostApduService klasse van android. De HostApduService klasse bevat twee abstracte methodes die overschreven worden in de eigen service klasse namelijk `processCommandApdu` en `onDeactivated`. De `processCommandApdu` methode wordt aangeroepen wanneer een NFC reader een APDU stuurt naar de apdu service en dan zal de service een APDU als antwoord terug sturen. Er moet wel rekening mee gehouden worden dat de `processCommandApdu` methode aangeroepen wordt op de main thread van de applicatie, wanneer het toestel dit niet aankan mag de main thread natuurlijk niet geblokkeerd worden. Wanneer de main thread het niet aankan moet de methode null teruggeven en een andere methode `sendApduResponse` die het antwoord op een andere thread terug zal sturen. De `onDeactivated` methode geeft terug of de apdu is teruggestuurd op de main thread of een andere thread wanneer de `processCommandApdu` wordt aangeroepen.

## 4. Conclusie



# A. Onderzoeksvoorstel

Het onderwerp van deze bachelorproef is gebaseerd op een onderzoeksvoorstel dat vooraf werd beoordeeld door de promotor. Dat voorstel is opgenomen in deze bijlage.

## A.1 Introductie

Tegenwoordig worden smart-cards zoals bankkaarten, klantenkaarten, ... maar al te vaak vervangen door digitale versies op smartphones en smartwatches. Dit verloopt via NFC of "Near Field Communication". Hiervoor wordt er normaal gezien een secure-element gebruikt die zorgt voor de beveiliging en de communicatie. Sinds Android 4.4 (KitKat) is het mogelijk om dit te doen via HCE of "Host-based card emulation". Eén van de nadelen van deze technologie is het veiligheidsrisico die het met zich meebrengt. Android HCE maakt het gemakkelijker om de NFC technologie te gebruiken omdat er geen nood meer is aan een secure-element, waardoor er dus ook geen commerciële overeenkomst meer moet afgesloten worden met de secure-element verdelers. Het doel van deze paper is om na te gaan hoe men Android HCE op een veilige manier kan gebruiken voor allerlei betalingssystemen.

## A.2 State-of-the-art

Door het gebruik van Android HCE is er een groot veiligheidsrisico. Dit doordat er geen gebruik meer wordt gemaakt van een secure-element die zorgt dat alle gegevens van de gebruiker veilig worden opgeslagen. Volgens de paper van Smart Card Alliance (Alliance, 2014) zijn de twee meest voorkomende manieren om met Android HCE te

werken via de cloud met een token en zonder token. Zonder token werken wordt niet als veilig aanschouwd doordat betalingsgegevens gemakkelijk ontdekt kunnen worden door malware. Het gebruik maken van een token verlaagt de kans niet tot het ontdekken van betalingsgegevens door malware, maar het verlaagt wel de impact van een eventuele ontdekking door het vervangen van statische betalingsgegevens met een token. Nadelen van het gebruiken van Android HCE (Lepojevic e.a., 2014) in vergelijking met een normale Smart Card is dat HCE niet werkt zonder stroom. Wanneer er geen verbinding kan gemaakt worden met de cloud kan HCE de authenticatie niet voltooien.

### A.3 Methodologie

Om de veiligheid te kunnen testen zal er een proof of concept opgesteld worden met drie verschillende beveiligingsmethodes namelijk Tokenisatie, Encryptie en biometrische factoren. Hierbij kan er dan vergeleken worden hoe veilig ze zijn ten opzichte van elkaar en ten opzichte van het gebruiken van een secure-element. Bij het vergelijken van de verschillende methoden zal er gekeken worden naar welke gegevens leesbaar zijn, hoe gemakkelijk het is om deze methoden te omzeilen, ... Er zal gebruikgemaakt worden van OWASP The Mobile Security Testing Guide om zo de verschillende veiligheidsproblemen op te sporen in de applicatie. Verder wordt ook de moeilijkheidsgraad gemeten van iedere implementatie zodat er een besluit kan gemaakt worden welke oplossing bij welke use case het best gebruikt kan worden.

### A.4 Verwachte resultaten

Het resultaat zal bestaan uit verschillende proof of concepts waarbij het gebruik van Android HCE toch op een veilige manier benut kan worden. Op basis van de resultaten zal dan ook beslist kunnen worden welke implementatie het beste is voor welke use case.

### A.5 Verwachte conclusies

Verwacht wordt dat de verschillende implementaties tot beveiligde applicaties leiden waarbij NFC technologie gebruikt kan worden via Android HCE. Alsook dat de applicatie met de biometrische factoren het veiligste zal zijn en hierbij de user experience het beste zal zijn omdat dit ook een groter gevoel van veiligheid geeft.

## Bibliografie

- Alattar, M. (2014). Host-based Card Emulation: development, security, and ecosystem impact analysis.
- Alliance, S. C. (2014). A Smart Card Alliance Mobile & NFC Council White Paper - Host Card Emulation (HCE) 101.
- Lepojevic, B., Pavlovic, B. & Radulovic, A. (2014). Implementing NFC Service Security.
- Pollefliet, L. (2011). *Schrijven van verslag tot eindwerk: do's en don'ts*. Gent: Academia Press.