ICT Group N.V.

HET ONTWIKKELEN VAN EEN EXTRA BEVEILIGING VOOR NETWERKCOMMUNICATIE VAN KLANTEN NAAR DE CLOUD VAN ICT GROUP

Afstudeerverslag

Ramiro Rayn Jagai
10065954
Technische Informatica
De Haagse Hogeschool
Versie 1.0
1-6-2017

Referaat

Jagai R.R., Technische Informatica, “Het ontwikkelen van een extra beveiliging voor netwerkcommunicatie van klanten naar de cloud van ICT Group”, Afstudeerverslag, ICT Group N.V., De Haagse Hogeschool Delft, 2017.

Trefwoorden:

- Beveiliging
- Netwerkcommunicatie
- Multi-factor authentication
- Proof of Concept
- Afstudeerverslag

Voorwoord

Dit afstudeerverslag beschrijft mijn afstudeertraject die is uitgevoerd bij ICT Group gevestigd in Barendrecht. Tijdens de afstudeerperiode zijn er werkzaamheden verricht die betrekking hebben met de bouw van een proof of concept dat de netwerkcommunicatie van klanten van ICT Group een extra veiligheid aanbiedt.

Dit verslag is bestemd voor de examinatoren van de Haagse Hogeschool. In het verslag wordt het proces van de opdracht beschreven. De keuzes die gemaakt zijn tijdens het project en het onderzoek worden eveneens verantwoord in dit verslag.

Mijn dank aan mijn begeleider bij ICT Group dhr. Van ‘t Hof voor de begeleiding en feedback die hij mij heeft gegeven tijdens mijn afstudeerstage. Naast mijn begeleider wil ik ook graag mijn dank betuigen aan alle andere collega’s en stagiaires bij ICT Group waar ik ook altijd terecht kon voor advies.

Barendrecht, 1-6-2017
Ramiro Jagai

Verklarende woordenlijst

In de volgende lijst worden veel voorkomende woorden verklaard op alfabetische volgorde.

Woord	Betekenis
(X)OTP	(Time- of HMAC-gebaseerd) One-Time Password
Azure Cloud	Een service van Microsoft voor het hosten van websites, databases etc.
Biometrie	Unieke kenmerken van een persoon die gebruikt kunnen worden bij identificatie
BLE	Bluetooth Low-Energy
BR/EDR	Basic Rate/Enhanced Data Rate
CA	Certificaatautoriteit
ECC	Elliptische-Curve Cryptografie
ECDH	Elliptische-Curve Diffie-Hellman
HTTPS	HyperText Transfer Protocol Secure
MFA	Multi-Factor authenticatie
PKI	Public Key Infrastructuur
Raspberry Pi	Een compact programmeerbaar computer dat werkt op een enkele printplaat
RSA	Encryptie algoritme afkorting gebaseerd op de namen van de ontwikkelaars
Single Sign-On	Eenmalig inloggen op een bepaalde dienst
SPKI/SDSI	Simple Public-Key Infrastructure/Simple Distributed Security Infrastructure
SSL/TLS	Secure Sockets Layer/Transport Layer Security
U2F	Universal Second Factor
WOT	Web Of Trust

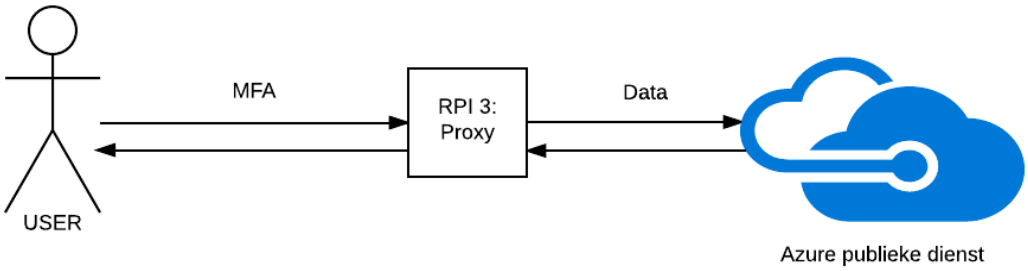
Inhoudsopgave

1 Inleiding

ICT Group heeft in haar 40 jaar bestaan een groot en divers portfolio van klanten opgebouwd. De klanten bevinden zich in een groot aantal uiteenlopende markten. De overkoepelende factor is technologie die versterkt wordt door software. Door een achtergrond in embeded softwareontwikkeling is Internet of Things een dagelijkse praktijk voor ICT Group geworden. Hierin zien zij een ontwikkeling dat bestaande "domme" voorwerpen steeds "slimmer" worden gemaakt.

Een aantal klanten van ICT Group zijn logistieke bedrijven in het havengebied van Rotterdam. Deze bedrijven bewaren de logistieke container informatie bij ICT Group. ICT Group vindt dat de verbinding dat nu gebruikt wordt een “onveilige” verbinding met hun eigen cloud. ICT Group is op zoek naar een generieke en vervangbare oplossing dat bij de klanten kan worden geplaatst zonder dat software op klantlocaties worden aangepast. ICT Group wil ook dat de oplossing medewerkers van klanten een veilige manier van authentiseren aanbiedt, zodat toegang tot gevoelige container informatie veilig gebeurd.

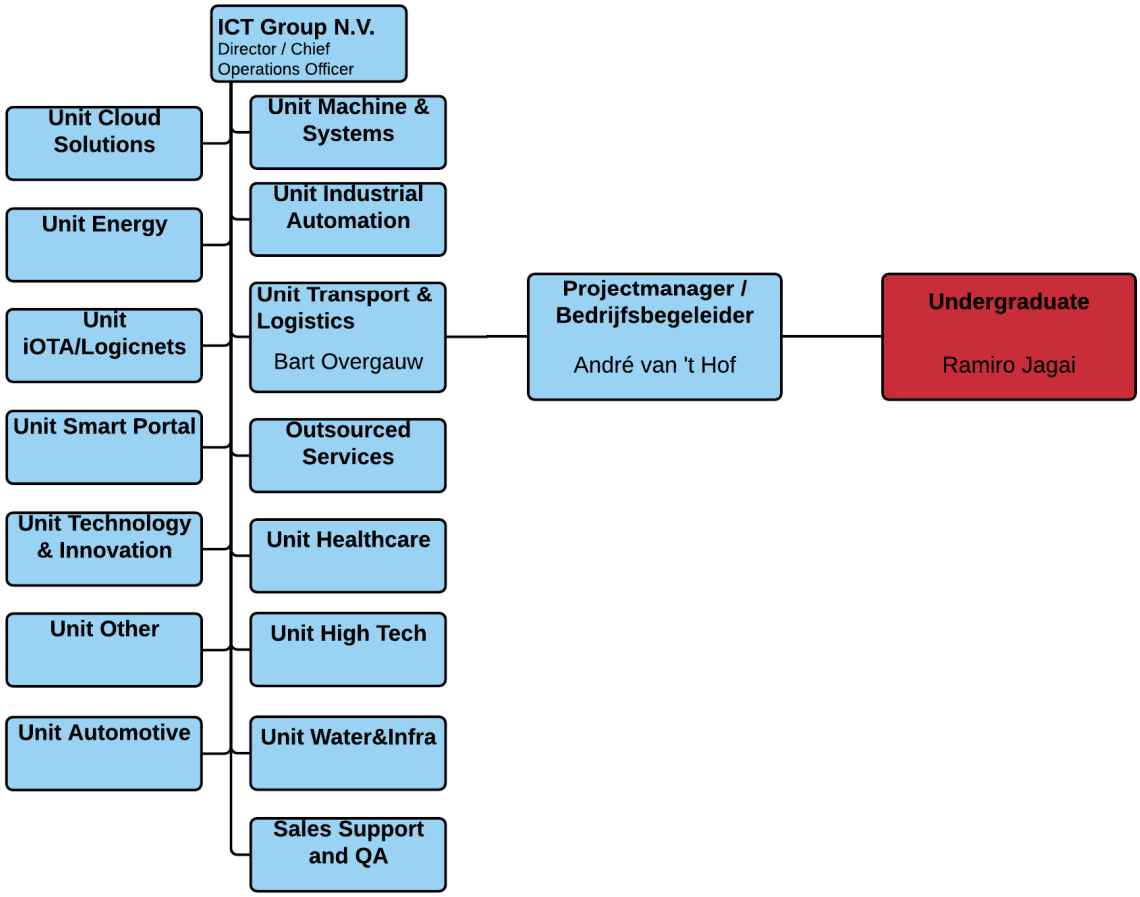
Dit afstudeerverslag beschrijft het ontwikkelen van extra netwerkbeveiliging voor klanten van ICT Group naar de Azure cloud van ICT. In het afstudeerverslag wordt eerst de organisatie beschreven waar de opdracht is uitgevoerd. Ten tweede wordt er een opdrachtomschrijving beschreven waarin de aanleiding, de probleemstelling, de doelstelling, het resultaat en projectgrenzen worden beschreven. Ten derde zal er een uitleg worden gegeven over de aanpak van de opdracht. Er worden kenmerken uit het project beschreven en een uitleg gegeven waarom de gekozen aanpak geschikt is voor dit project. Daarna wordt de oriëntatiefase beschreven dat een verheldering van de opdracht, risico’s en een planning van het project beschrijft. Als vijfde wordt er een definitiefase beschreven. De definitiefase beschrijft de analyse over het onderwerp, het achterhalen van de eisen van het informatiesysteem, het bepalen van een definitieve systeemarchitectuur en de uit te voeren incrementen. Het zesde hoofdstuk beschrijft de ontwikkelfase per increment. Er wordt per increment een beschrijving van het ontwerp, de realisatie en het testen gegeven. Vervolgens wordt er een conclusie van het proof of concept beschreven. Als laatste wordt er een persoonlijke evaluatie beschreven over het project.



Figuur 1.1 Visualisatie van de opdracht

2 Organisatie

ICT Group bestaat al sinds 1978 en is opgericht onder de naam ICT-automatisering. De naam van ICT is in de loop van de jaren vaak veranderd en sinds 2016 is de officiële naam van het bedrijf ICT Group N.V. Het hoofdkantoor van ICT is gevestigd in Barendrecht en het bedrijf heeft meerdere vestigingen door heel het land. ICT kan worden verdeeld in verschillende units die gerelateerd zijn aan de branches waarin zij werkzaam zijn. De units zijn Automotive & Mobility, Water & Infrastructure, Healthcare, Industry, Transport & Logistics, High Tech, Energy en Manufacturing. In Figuur 2.1 wordt een organogram van de ICT Group N.V. weergegeven.



Figuur 2.1 Organogram ICT Group N.V.

Belanghebbenden

Tabel 2.1 laat zien wie er allemaal betrokken zijn bij deze opdracht vanuit de organisatie ICT Group.

Naam	Functie	Functie project
Dhr. Ramiro Jagai	Student	Afstudeerder
Dhr. André van 't Hof	Projectmanager	Bedrijfsbegeleider
Dhr. Bart Lamot	Innovation manager	Opdrachtgever

Tabel 2.1 Belanghebbenden ICT Group

3 Opdrachtomschrijving

In dit hoofdstuk wordt de opdracht omschreven die zal worden uitgevoerd bij ICT Group. De omschrijving van de afstudeeropdracht kan worden verdeeld in een aanleiding, probleemstelling, doelstelling en resultaten. Deze omschrijving is opgesteld aan de start van het afstudeerproject om voor de student, opdrachtgever en begeleider een duidelijk beeld te creëren van de opdracht (zie bijlage, “Plan van aanpak”).

3.1 Aanleiding

ICT Group heeft in haar 40 jaar bestaan een groot en divers portfolio van klanten opgebouwd. De klanten bevinden zich in een groot aantal uiteenlopende markten zoals bijvoorbeeld energie, water & infrastructuur, healthcare en logistiek. De overkoepelende factor is technologie die versterkt wordt door software.

Door een achtergrond in embedded software ontwikkeling is Internet of Things een dagelijkse praktijk voor ICT Group geworden. Hierin zien zij een ontwikkeling dat bestaande "domme" voorwerpen steeds "slimmer" worden gemaakt. Er is gebleken dat er een vraag naar een product is om de volgende situatie op te lossen bij klanten van ICT Group:

Een aantal klanten van ICT Group zijn logistieke bedrijven in het havengebied van Rotterdam. Deze bedrijven bewaren de logistieke container informatie bij ICT Group. ICT Group gebruikt nu een directe onveilige verbinding met de hun eigen cloud.

3.2 Probleemstelling

Het probleem is dat klanten momenteel hun logistieke data niet veilig opslaan bij ICT Group.

3.3 Doelstelling

Het doel is dat klanten met behulp van een extra beveiligde verbinding logistieke data in de Cloud van ICT Group kunnen opslaan. Op deze wijze is de data van de klant beter beveiligd. ICT Group wil dat software die bij klanten staat zo min mogelijk aangepast moet worden.

3.4 Resultaten

Een zwaar beveiligde verbinding tussen de klant en ICT Group.

3.5 Projectgrenzen

Door meteen aan het begin van een project de grenzen vast te leggen wordt er een afbakening gedaan van het resultaat. Zo ontstaat er geen onduidelijkheid aan het eind van het project over wat de student heeft uitgevoerd.

De volgende punten vallen binnen de projectgrenzen:

- Het project wordt ontwikkeld op een Raspberry Pi met als OS Linux.
- Er wordt een externe multifactor authenticatie gebruikt die de verbinding met de Cloud tot stand brengt. Er wordt onderzoek gedaan naar de beste, niet – kopieerbare oplossing.
- De data moet ook worden beveiligd, daarvoor zal een encryptiemethode worden gekozen.
- Er mogen geen aanpassingen worden gedaan in bestaande software van klanten.

4 Aanpak afstudeeropdracht

In dit hoofdstuk wordt er gekeken naar de kenmerken van de afstudeeropdracht. Daarna worden verschillende methodes bekeken die passend lijken voor de opdracht. Met de kenmerken van de opdracht wordt er een afweging gemaakt en een methode geselecteerd.

4.1 Strategie en Methode

Om de afstudeeropdracht beheersbaar en overzichtelijk te maken wordt er een methode gekozen. Om een afweging te maken in verschillende methoden worden naar een aantal belangrijke factoren[1] van de opdracht gekeken aan de hand daarvan een keuze in een ontwikkelmethode en strategie kan worden gedaan.

- **Projectgrootte** – De grootte van het project is klein en wordt uitgevoerd door één student. De methode moet beheersbaar en simpel blijven zodat het uitvoerbaar is voor één student.
- **Gestructureerdheid van het probleem** – Het probleemdomein in dit project is gestructureerd. Het is duidelijk wat het probleem is en de oplossing kan met behulp van het uitvoeren van onderzoek en met een proof of concept worden aangetoond.
- **Taakinzicht van de gebruikers** – De opdrachtgever heeft aangegeven dat het systeem bedoeld is om klanten een veilige verbinding aan te bieden. De eisen van het systeem zijn consistent uitgelegd aan de student. Het is onwaarschijnlijk dat de eisen zullen veranderen in de toekomst. Wel moet de opdracht worden verhelderd, zodat specifieke wensen van de opdrachtgever duidelijk worden.
- **Deskundigheid student in probleemdomein** – De student heeft nog onvoldoende kennis over het onderwerp. Er zal een literatuuronderzoek moeten worden uitgevoerd over het onderwerp.
- **Veranderlijkheid van de probleemruimte en omgeving** – De opdrachtgever was vrij duidelijk in het product dat ze willen hebben. De software van klanten wordt zo min mogelijk aangepast, dus zal het probleem nog altijd blijven bestaan.

Selecteren ontwikkelstrategie

Voor het ontwikkelen van producten bestaan ook verschillende methodes die kunnen worden toegepast in projecten. Om een juiste keuze te maken bij het selecteren van diverse ontwikkelmethoden is er als eerste een aantal bekende ontwikkelstrategieën geselecteerd. Deze zijn vervolgens vergeleken met de kenmerken van het project. Er dient gekeken te worden in hoeverre de kenmerken van het project het best aansluiten binnen een ontwikkelstrategie. Hieronder volgen de categorieën waar het project op aan kan sluiten.

- **Lineair** – Bij een lineaire ontwikkelstrategie wordt er in fasen gewerkt en is het alleen mogelijk naar de volgende fase te gaan als alles af is van de voorgaande fase. Als de ontwikkelaar begonnen is met een bepaalde fase mag hij niet meer teruggaan naar een vorige fase. Uit elke fase komt een volledig product.

- **Iteratief** – Bij alleen iteratief software ontwikkelen wordt het ontwikkelproces steeds herhaald en elke herhaling levert een voorlopig volledige versie van de software op, waarna er om feedback wordt gevraagd om de software aan te passen. Als er dan als feedback wordt aangegeven dat het systeem niet voldoet, dan wordt dat gedeelte van het systeem weggegooid. Er kunnen dan wel bruikbare delen van de vorige iteratie worden gebruikt bij het opbouwen van de nieuwe versie van het systeem. Het proces wordt steeds herhaald totdat de klant tevreden is of er geen tijd meer over is of het project buiten het budget valt.

- **Incrementeel** – Bij incrementeel software ontwikkelen wordt de software in delen opgebouwd. Elke iteratie levert een increment op. Een increment is het doorlopen van de opgestelde disciplines wat een deel werkende code van de complete software oplevert. Dit proces herhaalt zich steeds totdat er een compleet werkend systeem uitkomt.

[1] (H.C. vd Bosch, 2011)

Het gebruik van iteratieve en incrementele strategieën worden vaak toegepast bij grote projecten die soms jaren kunnen duren of als het probleemdomein van het project niet duidelijk is. Zo kan door middel van herhalingen het project worden opgesplitst in iteraties om de grootte van een oplevering te beperken. Na elke herhaling wordt het ook duidelijker wat het probleemdomein is. In dit project is dat niet nodig, omdat de grootte van het project klein is en de structuur van het probleem duidelijk. Een lineaire aanpak lijkt het meest geschikt voor het toepassen bij het eerste gedeelte van het project, omdat het eerste gedeelte vooral bestaat uit literatuuronderzoek en vaststellen van eisen en wensen. Het tweede gedeelte bestaat uit het ontwikkelen van een proof of concept. Bij een watervalmethode wordt gevraagd in één keer alles op te leveren. Hierdoor kan de totale overzicht vervagen bij de student. Er is besloten het ontwikkelen van het proof of concept incrementeel te doen, zodat het ontwikkelproces voor de student overzichtelijk blijft.

Selecteren ontwikkelmethode

Om een juiste keuze te maken in een ontwikkelmethode wordt er eerst gekeken naar bestaande ontwikkelmethoden en of deze op zichzelf al voldoen naar de eisen van dit project. De opdrachtgever wil in dit geval een volledig onderzoek, ze hebben niks aan een voorlopige versie of een project dat nog niet af is. Dit kunnen ze niet presenteren aan klanten van ICT Group die de opstelling zouden kunnen gebruiken binnen hun bedrijf. Binnen de incrementele strategie zijn de volgende methodes geselecteerd die populair zijn in gebruik bij projecten: RUP, SCRUM, RAD en de Waterval-methode. Hieronder volgt een beschrijving van de voor- en nadelen bij het gebruik van de methoden in dit project.

Rational Unified Process (RUP) – De ontwikkelmethode RUP is een grote en complexe ontwikkelmethode. RUP bestaat uit verschillende fases die weer bestaan uit diverse disciplines. Alle disciplines leveren artefacten op die weer gedocumenteerd moeten worden. RUP wordt vaak gebruikt bij grote projecten waarbij het niet duidelijk is wat er ontwikkeld moet worden. Elke iteratie zorgt voor een tussentijdse oplevering. Na elke oplevering wordt er opnieuw gekeken naar de disciplines om te verifiëren of deze nog steeds overeenkomen met de wensen van de opdrachtgever. Als er veranderingen zijn in de wensen van de opdrachtgever kan ontwikkeling worden aangepast naar deze wensen. In dit project zijn de wensen van de opdrachtgever duidelijk in wat voor product ze willen hebben.

SCRUM – SCRUM is een veel gebruikte methode binnen ICT Group. De kracht van de methode ligt in het gebruik van teams en verschillende rollen te verdelen binnen het team. Volgens SCRUM worden er ook dagelijks met het team SCRUM–meetings gehouden om de voortgang van het project te bespreken. Ondanks dat SCRUM iteratief en incrementeel kan worden gebruikt, wordt er binnen SCRUM een maximale tijdsduur van 30 dagen aangegeven per iteratie. Bij deze afstudeeropdracht is het nog niet duidelijk wat precies de tijdsduur is van een iteratie. Het houden van dagelijkse meetings met de opdrachtgever heeft geen toegevoegde waarde voor dit project en de verschillende rollen kunnen niet worden verdeeld als er maar één ontwikkelaar aan het project werkt. Op veel kenmerken van dit project voldoet SCRUM niet. SCRUM verliest zijn kracht als de methode wordt aangepast naar de kenmerken van het project.

Rapid Application Development (RAD) – Het gebruik van RAD wordt aangeraden bij kleine tot middelgrote projecten, waar snel een werkend product uit moet komen. Bij RAD komt door de snelle ontwikkeling al snel een werkend product. Dit is voordelig als het product ontwikkeld moet worden binnen een korte tijd.

Waterval – De watervalmethode valt onder de strategie lineair ontwikkelen en is een rechttoe rechtaan ontwikkelproces. Het gebruik van de watervalmethode is simpel en er kan al aan de start van het proces een duidelijke planning ontstaan. Het nadeel van het deze methode is het terug gaan in het proces. Er kan niet makkelijk terug worden gegaan naar een vorig afgeronde fase.

Bij de methode RUP en SCRUM wordt vooral de focus gelegd op het ontwikkelen van softwareproducten. Bij dit project is het vooral het onderzoek dat leidt tot een proof of concept. Het proof of concept bestaat uit het toepassen van een MFA en encryptiemethoden. Het kan zijn dat er een deel software of scripts worden opgeleverd, maar dat is een klein gedeelte van het proof of concept. Een ander punt is dat RUP vaak wordt

toegepast op grote projecten en als het probleemdomein onduidelijk is. Dit project is klein en duidelijk, waardoor RUP geen geschikte methode lijkt. Het gebruik van SCRUM wordt gedaan in scrumteams. Dit project wordt uitgevoerd door één student, waardoor SCRUM ook afvalt.

Een medewerker van ICT Group heeft aangegeven om te kijken naar ontwikkelmethode RAD. De ontwikkelmethode werkt vooral door het snel opleveren van prototypes. Dit leek aan het begin van de opdracht geschikt, maar vrij snel is er ook weer beslissing gemaakt om van RAD af te zien. Het was niet nodig verschillende prototypes op te leveren, maar uiteindelijk een proof of concept dat het probleem van de opdrachtgever oplost.

In het afstudeerplan (zie bijlage, “afstudeerplan”) wordt er bij de aanpak een beschrijving gegeven dat handig leek voor het project. Deze beschrijving lijkt op de waterval-methode en lijkt toch het meest geschikt voor dit project. Het is duidelijk wat de opdrachtgever wil hebben en het is geen groot project. Het ontwikkel gedeelte wordt wel in incrementen gedaan, zodat er een overzicht blijft in de op te leveren producten.

4.2 Fasebeschrijving

In de vorige paragraaf is er een besluit genomen om een methode te gebruiken dat bestaat uit een gedeelte lineair en een gedeelte incrementeel ontwikkelen. De methode wordt in deze paragraaf uitgewerkt tot een fasebeschrijving. Naast de fases worden de activiteiten en conclusies die uit een fase komen uitgelegd.

Fasen	Activiteiten	Conclusies & doel
Oriëntatiefase	▪ Verhelderen van de opdracht ▪ Risico’s vaststellen ▪ Planning maken van project	Pas als de opdracht duidelijk is kan er goed onderzoek worden gedaan naar het onderwerp. Activiteiten van deze fase worden beschreven in het PVA.

Definitiefase	▪ Analyse probleemdomein
	  ○ Literatuuronderzoek
	  ○ Gebruik van USB, NFC, Bluetooth (als fysieke sleutel)
	  ○ Uitzoeken encryptiemethoden (PKI, AES, 3DES)
	  ○ Mogelijkheden in VPN
	  ○ Systeemconcepten voor een mogelijke architectuur bepalen
	▪ Achterhalen requirements van informatiesysteem
	  ○ Definitief vaststellen van requirements
	▪ Bepalen definitieve architectuur
		▪ Bepalen van de incrementen en de volgorde in het uitvoeren van de incrementen.
		▪ Activiteiten van deze fase worden beschreven in het definitierapport.

Ontwikkelfase	Increment 1: (Wordt bepaald in Definitiefase) ▪ Ontwerpen topologie van omgeving ▪ Realiseren van topologie met encryptie-omgeving ▪ Testen van realisatie Increment 2: (Wordt bepaald in Definitiefase) ▪ Ontwerpen werking van fysieke sleutel ▪ Realiseren van ontwerp ▪ Testen van realisatie Increment 3: (Wordt bepaald in Definitiefase) ▪ Ontwerpen van het systeem ▪ Realiseren van het ontwerp ▪ Testen van realisatie	▪ Per increment wordt een gedeelte proof of concept en ontwikkelrapport opgeleverd. ▪ De incrementen zijn na de definitiefase bepaald.

Tabel 4.1 Fasebeschrijving van de methode

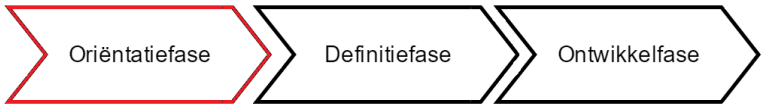
5 Oriëntatiefase

De oriëntatiefase wordt heeft als doel de opdracht te verhelderen. Het is duidelijk dat de opdrachtgever een zwaar beveiligde verbinding wil opzetten bij klanten van ICT Group. Het is nog niet precies duidelijk of er nog specifieke eisen zijn waaraan het project moet voldoen. In de oriëntatiefase worden ook risico's vastgesteld van het project en een planning gemaakt. De Oriëntatiefase is verdeeld als volgt:

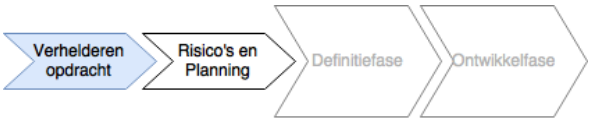
Oriëntatiefase:

- Verhelderen opdracht
- Risico's en Planning

In het plan van aanpak (Zie bijlage, "Plan van Aanpak") worden deze onderwerpen uitgebreid besproken.



5.1 Verhelderen opdracht



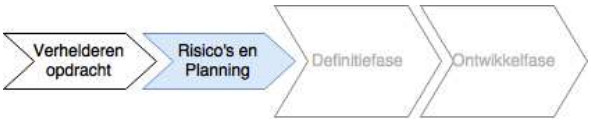
Vrij snel aan de start van het project zijn er met de opdrachtgever en begeleider meetings gehouden met als doel het verhelderen van de opdracht (Zie bijlage, "Interviewverslagen"). In deze meetings wordt er met behulp van interviews achterhaald wat de wensen zijn van de opdrachtgever bij het project. Deze wensen worden meegenomen in literatuuronderzoek en dan kan er worden bepaald welke wensen vastgesteld kunnen worden als eisen. In Tabel 5.1 worden citaten uit de gehouden interviews weergegeven die de specifieke wensen heeft op de opdracht.

Wens van ICT Group	Citaat uit interviewverslag
Niet de verbinding, maar de data	1. Wat is de huidige situatie bij klanten die gebruik willen gaan maken van deze extra beveiliging? "De huidige situatie is niet slecht. De klanten zijn er bang voor dat het kan gebeuren. Het idee is uitgewerkt door Bart, dat er een box komt dat de alleen medewerkers die via de box data verzenden naar de "cloud" alleen de mensen die ook in bezit zijn van dezelfde box de data kunnen zien. Mensen die van dezelfde dienst gebruik maken kunnen deze berichten niet zien en zien alleen de "normale" data waar zij alleen toegang toe hebben. ICT wil dit product op meerdere markten uitbrengen en niet alleen specifiek voor die havenbedrijven."
Raspberry PI is Generiek en vervangbaar	1. Is een Raspberry Pi wel een veilig apparaat, er zou makkelijk een MITM-aanval op worden gedaan. "De Raspberry Pi is het apparaat dat beschikbaar is voor het onderzoek. Als er uit een onderzoek uitkomt dat een ander apparaat beter is dan kunnen we kijken of het mogelijk is die aan te schaffen. Voor nu willen we graag een Raspberry Pi, omdat deze generiek en vervangbaar is. "

Multifactor-authenticatie	1. Is het bekend welke middel er gebruikt mag worden bij klanten, want een USB is niet de enige fysieke device waar de encryptie-key op zou kunnen staan. “Ik ben het met je eens dat USB verouderd is. Het fysieke apparaat mag je zelf bepalen met een juiste motivatie.”

Tabel 5.1 Wensenlijst

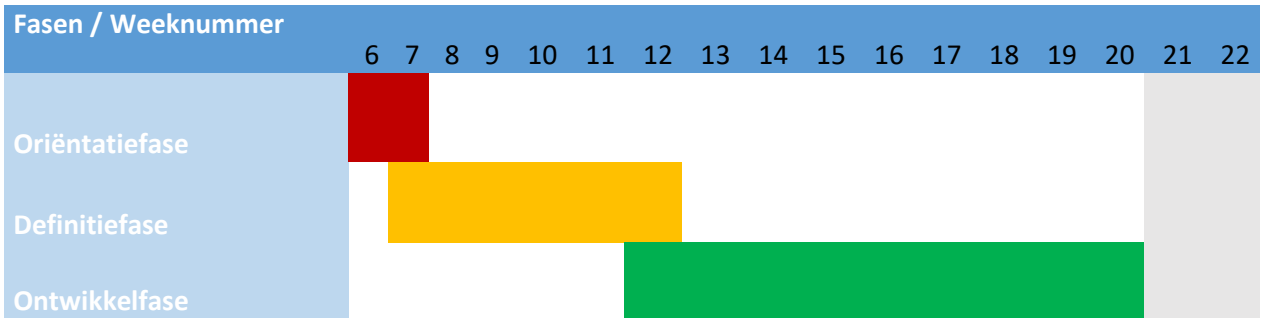
5.2 Risico’s en planning vaststellen



Bij een project komen er altijd risico’s kijken die een invloed kunnen hebben op het project. Door alle risico’s in kaart te brengen en daarbij de benodigde maatregelen op te stellen kan er een planning worden opgezet. Hieronder volgt een lijst van risico’s die zijn voorgekomen binnen het project. Bij elke risico wordt een vermelding van de kans dat een risico zal optreden en de impact van een risico en een omschrijving van de maatregelen die genomen kunnen worden om de risico’s te voorkomen. Ondanks de genomen maatregelen van een risico, zijn sommige toch voorgekomen en is Plan B uitgevoerd om het te verhelpen.

Risico	Kans	Impact	Maatregel	Plan B
Gebrek aan hardware	Klein	Groot	ICT zorgt voor de hardware die nodig is voor het uitvoeren van het project. De meeste componenten kunnen worden aangevraagd bij de helpdesk van ICT. Als er hardware nodig is die niet bij de helpdesk aan te vragen is kan er aan de begeleider worden gevraagd of het mogelijk is om de hardware te krijgen. Vraag dit tijdig aan bij de begeleider vanwege de kans dat het veel tijd gaat kosten om de hardware te regelen.	Het was snel duidelijk dat er een raspberry PI 3 nodig was voor het project. Deze moest nog besteld worden en heeft een aantal weken geduurd voordat deze er was. Er was wel een Raspberry Pi 2 aanwezig en deze is tijdelijk gebruikt om op te ontwikkelen.
Problemen met opzetten van testomgeving en werkomgeving	Groot	Middel	Het interne netwerk van ICT Group is goed beveiligd. Er mogen niet zomaar apparaten aangesloten worden op het netwerk. Er moet een aanvraag gedaan worden met het MAC-adres van het apparaat bij de helpdesk. Het risico is makkelijk te voorkomen door al vroeg aan te geven bij de helpdesk dat de student bezig is met een afstudeeropdracht en de MAC-adressen door te geven als die bekend zijn.	Voor de start van het project heeft de begeleider het MAC-adres van de student doorgegeven, maar dat leek bij aankomst op de eerste dag niet goed te zijn gegaan. Na zelf contact opnemen met de helpdesk hebben zij wel het MAC-adres kunnen toevoegen.

Nadat het duidelijk was wat de risico's waren binnen het project is er een globale planning opgezet. In Tabel 5.2 wordt de planning weergegeven. Deze planning geeft een verwachting van de tijd in weken aan in een hoeveelheid tijd een fase zal gaan duren. De laatste twee weken waren alvast ingepland om aan het afstudeerdocument te werken.



Tabel 5.2 Planning uit Plan van Aanpak

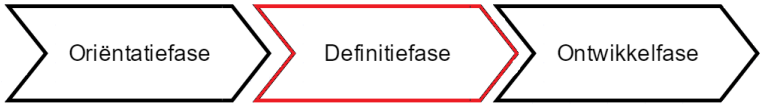
6 Definitiefase

In dit hoofdstuk worden werkzaamheden die in de definitiefase gedaan zijn beschreven. Er is onvoldoende kennis over de technieken, dus moest er worden ingelezen over het onderwerp. Met behulp van het onderzoek en de gehouden interviews zijn de requirements opgesteld. Daarna zijn er mogelijke systeem opstellingen opgesteld. Met behulp van het onderzoek en requirements is er een proof of concept gekozen dat ontwikkeld wordt. De definitiefase is verdeeld in de volgende punten:

Definitiefase:

- Onderzoek
- Requirements vaststellen
- Mogelijke opstellingen
- Definitieve architectuur
- Incrementen bepalen

De volledige beschrijving van deze fase wordt gedaan in het definitierapport (zie bijlage, “Definitierapport”).



6.1 Onderzoek

Tijdens de definitiefase is er onderzoek verricht naar mogelijke encryptiemethoden, verschillende MFA. Al de verzamelde informatie over de verschillende onderwerpen zijn gedocumenteerd in het definitierapport (Zie Bijlage, “Definitierapport”). In het rapport worden met behulp van de gevonden informatie verschillende systeemconcepten opgezet en daaruit een afweging gemaakt naar het meest geschikte systeem voor de opdrachtgever.

Versleutelde data onderscheiden

Een wens van de opdrachtgever was een veel gebruikt publiek systeem gebruiken om het proof of concept te bewijzen. Er is besloten om dit te onderzoeken en als het mogelijk is als eis in het project mee te nemen. Er zal eerst moeten worden onderzocht hoe data wordt weggeschreven en opgehaald bij de publieke dienst. Om dat te achterhalen moet netwerkverkeer bestudeerd worden. Het netwerkverkeer van de publieke dienst wordt over een beveiligde verbinding verstuurd en er zijn een aantal manieren geprobeerd om te achterhalen hoe het netwerkverkeer is opgebouwd.

- **Wireshark** – Het netwerkverkeer onderscheppen en bekijken met een Pre-master key of SSLstrip.[2]
- **MITMProxy** – Al het netwerkverkeer door een SSL-interceptie proxy op te vangen. Er wordt een eigen certificaat aangebracht op de cliënt-pc die een verbinding opzet naar de proxy. De proxy pakt dan het verkeer uit en opnieuw weer in. Vervolgens zorgt de proxy voor een verbinding met facebook.com.

[2] Shaver, j. (2015, February 11). *Decrypting TLS Browser Traffic With Wireshark – The Easy Way!*

- **Owasp ZAP** – Owasp als een browser-proxy gebruikt wordt en het netwerkverkeer op een applicatieniveau bekeken.

Het was deels mogelijk het netwerkverkeer te bekijken, maar niet genoeg om te achterhalen hoe het publieke systeem zijn data ophaalt of wegschrijft. Dit was vrij al snel duidelijk en is met de opdrachtgever besproken om zelf een publieke dienst op te bouwen. Met een eigen publieke dienst is er toegang tot alle gegevens van het systeem en kan de werking van het proof of concept worden bewezen.

Encryptiemethodes

Symmetrische Encryptie

Er zijn verschillende symmetrische encryptiemethodes onderzocht die mogelijk gebruikt kunnen worden in het project. Bij een symmetrische encryptie wordt één sleutel gebruikt voor het versleutelen en ontsleutelen van data. De volgende symmetrische encryptie algoritmes zijn gevonden en onderzocht:

- **3DES** – (Triple **D**ata **E**ncryption **S**tandard). Dit algoritme is eigenlijk het drie keer toepassen van een normale DES-encryptie achter elkaar. DES maakt gebruik van een 56-bit sleutel. Deze sleutel is zo klein dat hij makkelijk te kraken is. De uitbreiding naar 3DES zorgt voor een 168-bit sleutel. De sleutel kan niet meer zo makkelijk worden gekraakt.
- **Serpent** – Het Serpent algoritme is tweede geworden bij de AES-competitie. Het algoritme gebruikt datablokken van 128-bit en 128, 192 of 256-bit sleutels. Het algoritme moet 32 encryptierondes doorlopen.
- **AES** – (**A**dvanced **E**ncryption **S**tandard) van Rijndael is bekroond tot opvolger van het DES-algoritme. Het heeft een sleutelruimte van 128,192 of 256-bit en gebruikt datablokken van 128-bit. Het algoritme gebruikt 10,12 of 14 encryptie rondes. De grootte van de sleutel bepaalt het aantal encryptierondes
- **Twofish** – Dit algoritme is de opvolger van het eerdere Blowfish-algoritme dat een van de vijf finalisten was in de AES-competitie. Het gebruikt, net als AES, 128-bit datablokken en 128,192 of 256-bit sleutels. Het algoritme gebruikt 16 encryptierondes.

AES is gekozen op basis van de volgende drie factoren: veiligheid, snelheid en simpelheid. AES vergeleken met de andere symmetrische algoritmes gaan som vooruit op een van de drie factoren, maar gaan dan zwaar achteruit op een andere factor. Rijndael's AES zal daarom worden gebruikt is in dit project als meest geschikte symmetrische encryptiemethode.

AES heeft de volgende drie mogelijkheden voor een sleutelgrootte: 128-, 192- en 256-bit sleutel. AES is zo snel dat het geen invloed heeft in performance op de Raspberry Pi. Er zal dan ook de meest veilige sleutel van 256-bit worden gebruikt binnen het project.

Binnen AES worden 128-bit datablokken gebruikt. Als de data groter is dat 128-bit moeten er meerdere blokken aan elkaar gekoppeld worden. Hiervoor bestaan verschillende modus. In Figuur 6.1[3] wordt een voorbeeld van het gebruik van ECB en CBC weergegeven. Bij het gebruik van de ECB-modus kunnen de patronen van originele document afgelezen. Dit komt doordat ECB elke datablok op dezelfde manier wordt

Figuur 6.1 ECB vs CBC-modus

versleuteld. Bij CBC en alle andere modus kunnen de patronen niet achterhaald worden en zijn veilig in gebruik. De volgende gevonden verschillende modus zijn vergeleken: CBC en CTR

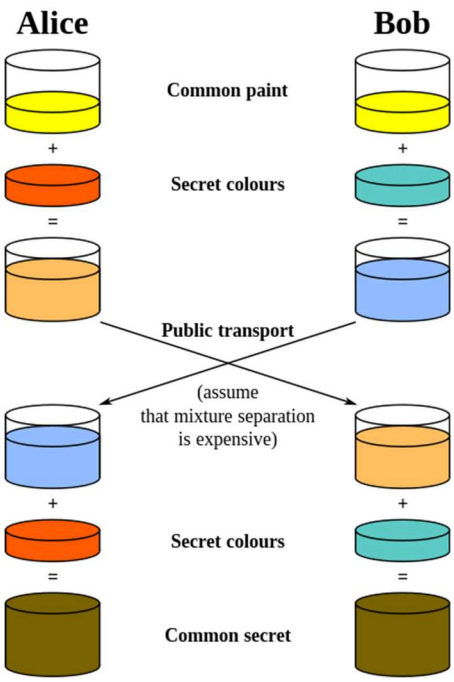
CBC is een veel gebruikte modus bij het gebruik van AES. Vergeleken met ECB is het langzamer, maar de patronen zijn niet zichtbaar. Bij CBC wordt er per datablok een encryptie gedaan met behulp van de vorige datablok. Het is bij CBC niet mogelijk om parallel te encrypten, omdat bij elke encryptie de vorige datablok nodig is. Decryptie kan wel parallel gedaan worden, omdat alle datablokken dan bekend zijn en gebruikt kunnen worden in het decryptie-proces.

CTR maakt gebruik van de waardes van een counter. CTR biedt gemakkelijk random access aan dit zorgt ervoor dat er een gedeelte van het bestand decrypt kan worden zonder dat het hele bestand hoeft te worden decrypt. CTR wordt afgeraden bij datablokken van minder dan 128-bit. CTR maakt parallel encryptie mogelijk waardoor een hoge snelheid bij het encrypten. Dit werkt alleen in zijn voordeel als het apparaat dat de encryptie doet een multi-core processor heeft. De Raspberry-Pi heeft een quad-core ARM-processor waardoor CTR een optie lijkt.

Beide modus bieden geen integriteit aan. Er wordt niet gecontroleerd of er met de data is gerommeld. Een oplossing daarvoor is het versturen van een Message Authentication Code (MAC). Er zou apart nog een HMAC meegestuurd kunnen worden, zodat er een zekerheid is dat de data versleutelde data niet is aangepast.

[3] https://www.slideshare.net/amiralisan/cryptography-in-python

Een nadeel van symmetrisch encryptie is dat er één sleutel is om de kunnen encrypten en decrypten. Deze sleutel moet dan gedeeld worden met een andere partij zodat zij ook de berichten kunnen lezen. Met het Diffie-Helman algoritme kunnen twee partijen zonder voorkennis over een onbeveiligde verbinding een sleutel uitwisselen om een beveiligde verbinding op te zetten. In Figuur 6.2[4] wordt het Diffie-Helman algoritme uitgelegd met behulp van kleuren. Alice en Bob hebben spreken beiden een kleur af dat geen geheim is. Beiden mengen deze kleur met een geheime kleur die alleen voor hunzelf bekend is. Hieruit komt een gemengde kleur die ze naar elkaar versturen over een onbeveiligde verbinding. Zij mengen dan de ontvangen gemengde kleur van de ander met hun eigen geheime kleur. Dan ontstaat er een kleur die alleen voor Alice en Bob bekend zijn en de sleutel is voor het opzetten van een beveiligde verbinding.



Figuur 6.2 Diffie-Helman uitgelegd in kleuren

Een derde afluisterende partij zou net zo lang verschillende kleuren moeten uitproberen totdat hij een kleur krijgt die gelijk is met een van de kleuren van "Public transport" in het bovenstaande figuur. Als de waardes worden vervangen door getallen met een kleine waarde kan dit nog wel gedaan worden. Als de getallen grote waardes hebben dan kan het jaren duren voordat de externe partij het juiste getal weet. Dit wordt ook wel het discrete logaritme probleem genoemd.

Asymmetrische encryptie

Naast symmetrische encryptie dat gebruik maakt van één sleutel bestaat ook asymmetrische encryptie dat gebruikt maakt van twee sleutels. Veel van de hedendaagse asymmetrische encryptiemethoden zijn gebaseerd op het Diffie-Helman algoritme. Om de werking van asymmetrische encryptie te begrijpen is het belangrijk het Diffie-Helman algoritme te begrijpen, uitgelegd in Figuur 6.2.

De sleutels binnen asymmetrische encryptie worden ook wel een publieke en privé-sleutel genoemd. De publieke sleutel is kenbaar voor iedereen, maar de privé sleutel niet. De publieke sleutel kan alleen voor versleuteling worden gebruikt, maar de privé sleutel kan versleutelen en ontsleutelen van de data.

De volgende twee asymmetrische encryptie algoritmes zijn gevonden en onderzocht:

- **RSA**[5] – Op dit moment het meest gebruikte asymmetrische encryptie algoritme. Het wordt wel vereist om grote getallen te gebruiken voor de sleutel. Bij het gebruik van een lage waarde voor de sleutel zouden computers de sleutel kunnen ontcijferen.
- **ECDH** (**E**lliptisch **C**urve **D**iffie-**H**ellman) – Het ECDH-protocol is een opvolger van het RSA-protocol. Het maakt gebruikt van het Elliptisch Curve Cryptografie (ECC) -algoritme, maar wordt alleen nog niet gebruikt als standaard. Dit is omdat deze manier van asymmetrische encryptie vrij nieuw is en het is een complex algoritme. Niet iedereen begrijpt de algoritme, waardoor men de voorkeur heeft nog RSA te gebruiken. Ondanks dat werkt het sneller werkt dan het RSA-protocol en dat maakt ECC-encryptiealgoritmes aantrekkelijk[6].

[4] https://simple.wikipedia.org/wiki/Diffie-Hellman_key_exchange
[5] A Method for obtaining digital signatures and public-key cryptosystems -- http://people.csail.mit.edu/rivest/Rsapaper.pdf
[6] *RSA vs ECC Comparison for Embedded Systems – Atmel - http://www.atmel.com/Images/Atmel-8951-CryptoAuth-RSA-ECC-Comparison-Embedded-Systems-WhitePaper.pdf*

Uit onderzoek is gebleken dat beide protocollen veilig zijn in het gebruik ervan. Het ECDH-protocol wordt niet als standaard gebruikt, maar is veel sneller dan het RSA-protocol. Het ECDH-protocol gebruikt een minder grote encryptie-sleutel waardoor deze zuiniger is in het gebruik van geheugen en processorkracht.[7] Ondanks de voordelen van ECDH vergeleken met RSA, blijft RSA een standaard. Het RSA-protocol bestaat veel langer dan het ECC-protocol en de kans dat er in de toekomst een fout in het RSA- protocol wordt ontdekt is veel kleiner. Dit maakt het RSA-protocol betrouwbaarder dan het ECDH-protocol. Het project wordt uitgevoerd echter op een Raspberry Pi waar niet veel geheugen en processorkracht beschikbaar is. Als er bespaard kan worden op het geheugen en processorkracht moet dat zeker worden gedaan. Er zal een op ECDH gebaseerd algoritme worden gebruikt als het gaat om een asymmetrische encryptie.

[7] Martijn Grooten - Elliptic Curve Cryptography for those who are afraid of maths

Infrastructuur

Na het onderzoeken van verschillende encryptiemethodes zijn er verschillende manieren onderzocht voor het opzetten van veilige infrastructuur. Vooral als er gebruik gemaakt wordt van asymmetrische encryptie is het belangrijk dat de ontvanger beschikt over de juiste public key. Dit wordt gedaan met behulp van een Public Key Infrastructure (PKI). De volgende manieren zijn gevonden en onderzocht als mogelijk infrastructuur.

- **Certificaten PKI** – Werkt op basis van certificaten en is op het moment het meest gebruikte PKI op het Internet. Een derde vertrouwde partij verifieert door middel van certificaten de public key van een instantie. Dit certificaat moet ook beveiligd aangeleverd worden met een public key encryptie. Er ontstaat een keten die gemanaged wordt.
- **Blockchain PKI** – Het gebruik van aaneen geketende hash-kettingen om certificaten te verifiëren. Er is dan geen derde vertrouwde partij meer nodig. Deze manier van certificaten verifiëren bevindt zich nog in een conceptfase en wordt nog niet veel gebruikt.[8]
- **Web of Trust (WOT)** – Iedereen kan certificaten verifiëren en is vertrouwd op basis van het vertrouwen van anderen. Het WOT wordt veel gebruikt binnen het PGP-protocol.
- **Simple PKI/ Simple Distributed Security Infrastructure (SPKI/SDSI)** – Het linken van namen en toegang met sleutels. Het idee is nooit verder gekomen dan een conceptfase.
- **Kerberos** – Authenticatie protocol dat werkt met een externe server dat tickets uitgeeft aan de gebruikers zodat ze hun identiteit kunnen aantonen.

Het onderzoek heeft aangetoond dat een Certificaten PKI het meest geschikt is binnen dit project. In het verleden is er voorgekomen verleden dat een CA beïnvloed was, maar de kans dat dit gebeurd is klein. In een bedrijfsomgeving is het van belang om te weten wie er in vertrouwen wordt genomen. Bij een WOT is dat niet altijd duidelijk, omdat bij een WOT-PKI individuelen in vertrouwen worden genomen in plaats van een CA. Dit neemt het risico weg dat een CA beïnvloed zou kunnen zijn. Alleen is het niet altijd duidelijk wie er in vertrouwen wordt genomen. De SPKI/SDSI -omgeving is een oud concept dat niet verder is gekomen dan een conceptfase. Een Blockchain PKI is een nieuwe ontwikkeling dat zich ook in een conceptfase bevindt. Beiden bieden het minste zekerheid in beveiliging. Een Blockchain PKI zou mogelijk nog in de toekomst gebruikt kunnen worden als het verder is ontwikkeld. Kerberos is meegenomen in het onderzoek, maar het is een authenticatieprotocol. Het protocol versleutelt het verkeer dat over het netwerk heen gaat niet. Dit maakt het protocol kwetsbaar voor een MITM -aanval. Het wordt dan ook vaak toegepast in combinatie met het certificaten PKI om zo de veiligheid van de data te waarborgen.

[8] André Clerc - Temet. (2017, February 9). *About & Beyond PKI - Blockchain and PKI.*

Multifactor-authenticatie mogelijkheden

Een wens van de opdrachtgever is een toevoeging van een multifactor-authenticatie (MFA). Er bestaan veel manieren om een MFA te implementeren. De opdrachtgever heeft hier als wens dat het MFA iets fysieks is. De volgende mogelijke MFA-mogelijkheden leken het meest te passen bij wat de opdrachtgever zoekt in een MFA.

- **Bluetooth authenticatie** – Het verbinden van een apparaat via Bluetooth. Het wachtwoord is het apparaat dat verbindt met de Raspberry Pi.
- **OTP** (One-Time Password) – Het genereren van een op tijd of hash gebaseerd wachtwoord.
- **U2F** (Universal 2nd Factor) – Een soort USB-sleutel die een private-key bewaart. Door het drukken op de knop van de USB-sleutel wordt er een bericht gesigneerd met de private-key.

Bluetooth

De opdrachtgever wilde graag als optie een smartphone in combinatie met Bluetooth als mogelijk MFA. Het protocol is veel in hedendaagse objecten te vinden en de Raspberry Pi 3 heeft er ook een module voor. Er bestaan twee aftakkingen van het Bluetooth-protocol:

- Bluetooth standaard (BR/EDR)
- Bluetooth SMART / Bluetooth Low-Energy (BLE)

Bluetooth standaard wordt veel gebruikt bij het streamen van een hoge data hoeveelheid. Het gebruik van Bluetooth standaard kost veel energie voor het apparaat dat het communicatieprotocol gebruikt. Een smartphone die op batterijen werkt zou daar snel last van gaan hebben. De Special Interest Group (SIG) heeft daarop vanaf Bluetooth 4.0 een aftakking van het Bluetooth-protocol ontwikkelt genaamd Bluetooth SMART. Bluetooth SMART wordt ook wel Bluetooth Low-Energie (BLE) genoemd. BLE kan geen hoge data hoeveelheden verzenden, daarentegen verbruikt het veel minder batterijvermogen en is vele malen sneller. Tijdens het onderzoek werd duidelijk dat het SIG Bluetooth als het ware had “uitgekleed” om zo het energiebesparende BLE te ontwikkelen. Door het “uitkleden” van het protocol was de veiligheid die het protocol had fors achteruitgegaan[9].

Begin 2017 heeft de SIG Bluetooth 5 uitgebracht. De SIG geeft aan dat deze versie van het protocol 2x sneller is, 4x meer bereik heeft en 8x meer broadcastruimte[10]. Ondanks de verbeterde beveiliging geven critici aan dat hackers altijd wel een mogelijkheid hebben ontdekt om bluetooth te misbruiken. Op een artikel over Bluetooth 5 staat het volgende beschreven “2x the speed + 4x the range = 8x the security risk”[11]. Zij bedoelen hiermee dat door het uitbreiden van het bereik in bluetooth 5 hackers vanaf een grotere afstand en sneller Bluetooth zouden kunnen misbruiken. De korte afstand die nodig was om een Bluetooth verbinding tot stand te brengen zorgde voor een vorm van beveiliging. Hackers kunnen met Bluetooth 5 al van ver het Bluetooth verkeer onderscheppen.

Vanaf de aftakking in Bluetooth 4.0 zijn opvolgende versies van het protocol uitgebracht met verbeteringen aan energiebesparing, grootte van dataverkeer en veiligheid. De volgende mogelijkheden bluetooth versies zijn onderzocht:

Klassieke Bluetooth: ▪ Bluetooth 4.1 ▪ Bluetooth 4.2 ▪ Bluetooth 5	**Bluetooth Low-Energy**: ▪ BLE Legacy (4.0 & 4.1) ▪ BLE Secure Connections (4.2) ▪ Bluetooth 5

[9] Bluetooth-low-energy-security.pdf – Bluetooth SIG
[10] Bluetooth-5-faq.pdf – Bluetooth SIG
[11] http://www.devicelock.com/blog/2842.html - DeviceLock DLP

De verschillen in beveiliging van elke aftakking en versie worden in Tabel 6.1 en Tabel 6.2 weergegeven.

Klassieke Bluetooth (BR/EDR)	Bluetooth 4.1	Bluetooth 4.2	Bluetooth 5
Voordelen	- Sterke sleuteluitwisseling - Sterke encryptie	- Sterke sleuteluitwisseling - Sterke encryptie	- Sterke sleuteluitwisseling - Sterke encryptie
Nadelen	- Energie verbruik	- Energie verbruik	- Energie verbruik

Tabel 6.1 Beveiliging voor- en nadelen Bluetooth standaard

Bluetooth Low-Energy	Bluetooth Legacy (4.0 & 4.1)	Bluetooth Secure Connections (4.2)	Bluetooth 5
Voordelen	- Sterke encryptie	- Sterke sleuteluitwisseling - Sterke encryptie	- Sterke sleuteluitwisseling - Sterke encryptie
Nadelen	- Zwakke sleuteluitwisseling	- Geen	- Groot bereik

Tabel 6.2 Beveiliging voor- en nadelen Bluetooth Low-Energy

Na het opstellen van de tabellen met de voor- en nadelen van Bluetooth is er gekeken naar de mogelijkheden van de Raspberry Pi 3. Deze heeft een Bluetooth 4.1 module geïmplementeerd. Bluetooth klassiek heeft in 4.1 al de “Secure Connections” update gehad. BLE Legacy daarentegen maakt in versie 4.1 nog gebruik van een zwakke sleuteluitwisseling en is daarom niet geschikt als een MFA voor het project. Bluetooth standaard 4.1 is wel veilig genoeg voor het gebruik als MFA in dit project, omdat er geen veiligheidsnadelen zijn gevonden met Bluetooth standaard als MFA.

One Time Password (OTP)

OTP is een bekend algoritme voor het genereren van een wachtwoord dat één keer geldig is voor een authenticatieproces. Er bestaan verschillende soorten OTP’s en volgende zijn gevonden en onderzocht:

- HMAC-based One Time Password (HOTP)
- Time-based One Time Password (TOTP)

HOTP genereert wachtwoorden gebaseerd op een HASH bericht. Het wachtwoord is dan voor een onbepaalde tijd geldig. Het TOTP-protocol genereert wachtwoorden gebaseerd op de tijd en de wachtwoorden zijn vaak een beperkte tijd geldig. Dit maakt het gebruik van TOTP veiliger vergeleken met het HOTP-protocol. In het onderzoek zijn de volgende voor- en nadelen gevonden voor dit project:

Voordelen OTP:

- **Makkelijke implementatie** – Het is niet moeilijk op te zetten. Er bestaan apps op de smartphone die deze dienst aanbieden.
- **Beperkte tijdsduur TOTP** – Door de beperkte geldigheidduur van de wachtwoorden van TOTP is het lastig om een aanval uit te voeren. Een aanval zou in real-time moeten gebeuren.

Nadelen OTP:

- **Werkt op batterij** – Apparaten waarop de token op wordt gegenereerd werken op batterijen. Als deze op is kan de gebruiker niet meer inloggen. Tenzij er een mogelijkheid is om het apparaat op te laden of batterijen te vervangen.
- **Verouderde software** – Als het protocol gebuikt wordt op een smartphone app dan kan het voorkomen dat de software verouderd is. Dit kan ervoor zorgen dat de codes die gegenereerd worden niet meer overeenkomen met elkaar.
- **Geen phishing-bescherming** – Het protocol heeft geen bescherming tegen phishing-aanvallen. Bij het protocol TOTP is door de beperkte geldigheidsduur van het wachtwoord lastiger te achterhalen wat het wachtwoord is op het moment.

Universal 2nd Factor (U2F)

Het U2F-protocol is speciaal ontwikkeld voor het uitvoeren van MFA door Google en Yubico. De standaard wordt op het moment ondersteund door de FIDO Alliance. U2F is open-source en wordt tegenwoordig veel toegepast in combinatie met USB-sticks. De browser communiceert met het U2F-apparaat over een USB-poort met challenges. De U2F-stick bevat een privé-sleutel die gebruikt wordt om de challenges te signeren. Het signeren wordt gedaan met knop op de USB-stick.[12] Uit het onderzoek zijn de volgende voor- en nadelen van het U2F-protocol gevonden voor dit project:

Voordelen U2F:

- **Usability –** Het is simpel te gebruiken. De gebruiker steekt de sleutel in de USB-poort van de computer en met een druk op de knop is er een complex authenticatieproces afgerond.
- **Phishing-bescherming** – Het U2F protocol heeft een ingebouwde phishing-bescherming. Door middel van het meesturen van een Origin kan worden achterhaald of er data is aangepast.
- **Privacybescherming** – Door het aanmaken van een verschillende private- en public-key, wordt voorkomen dat providers kunnen zien van welke diensten een gebruiker gebruik maakt

Nadelen U2F:

- **Zwakke cloning beveiliging –** Het optellen van een counter is geen sterke beveiliging tegen het kopiëren van de U2F-stick. Als de gekopieerde stick eerder gebruikt wordt dan de originele, dan zal de server de gekopieerde stick herkennen als de originele.

Biometrische beveiliging

Aan het begin van het project is besloten om de mogelijkheid voor een biometrische beveiliging mee te nemen in het onderzoek. Als er voldoende tijd over was dan zou er nog een increment worden toegevoegd met een biometrische MFA. De volgende biometrische MFA-mogelijkheden leken geschikt voor het project en zijn onderzocht:

- **Vingerafdrukherkenning** – Met behulp van een vingerafdrukscanner de vingerafdruk van de gebruiker vergelijken met een vingerafdruk in een database.
- **Stemherkenning** – Met behulp van een microfoon de stem van de gebruiker vergelijken met een opname in een database.
- **Gezichtsherkenning** – Met behulp van een camera het gezicht van de gebruiker vergelijken met een foto in een database.

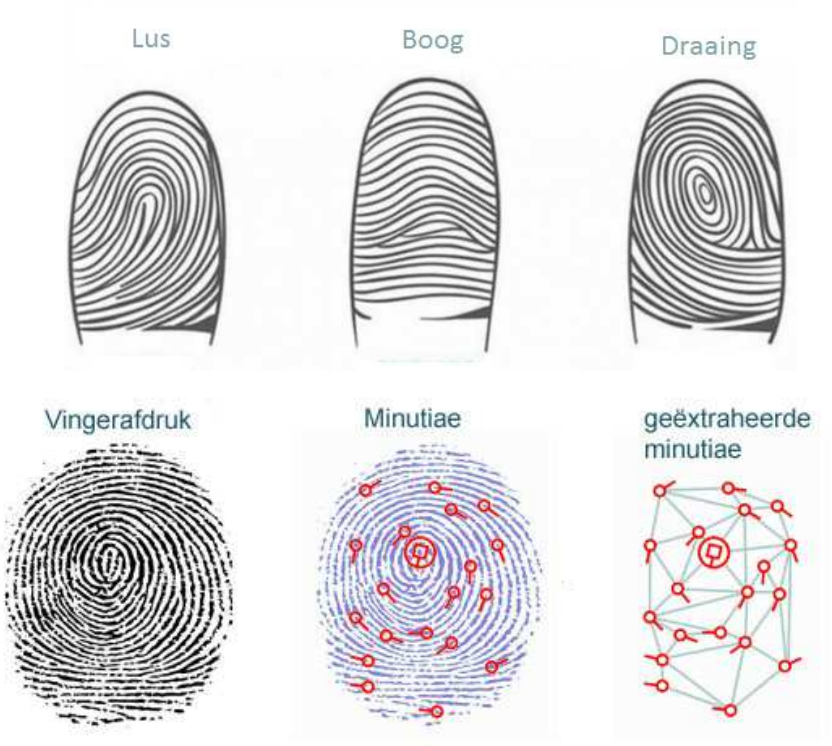
[12] https://robn.io/talks/u2f-lca-2017/U2F-notes.pdf - *U 2 can U2F - linux.conf.au 2017 – Hobart, Tasmania*

Vingerafdrukherkenning

Een van de meest bekende biometrische identificatie-methode is vingerafdrukherkenning. Een vingerafdruk is voor een persoon zijn gehele leven hetzelfde en kan niet makkelijk worden veranderd. Er zijn twee verschillende methodes gevonden die vingerafdrukken met elkaar vergelijken:

- **Afstandsmethode** – Het definiëren van verschillende punten in de vingerafdruk en het vaststellen van de afstand tussen die punten.
- **Huffmancodering** – De binaire code van de vingerafdrukafbeelding comprimeren met als uitkomst een unieke vector. De vector wordt dan vergeleken met andere vectoren in een database.

Van een vingerafdruk bestaan er drie basispatronen. De basispatronen zijn een lus-, boog- en draaiing patroon. Binnen de drie basispatronen kunnen afgeleide patronen bestaan. Een vingerafdruk bestaat uit verschillende unieke eigenschappen, ook wel minutiae genoemd. In Figuur 6.3 [13] wordt een voorbeeld van vingerafdrukpatronen en minutiae weergegeven.



Figuur 6.3 Vingerafdrukpatronen en minutiae

Er is besloten bij een vingerafdruk biometrie de afstandsmethode te gebruiken. Een vingerafdrukscanner bij elke Raspberry plaatsen wilde de opdrachtgever niet. Als het nodig is een biometrische beveiliging toe te voegen aan het proof of concept, zou een oplossing het gebruik van de vingerafdrukscanner die in smartphones zit.[14]

[13] http://vemichron.eu/be/nl/eur/nieuws/algemeen/biometrie -- http://dailyvibes.org/fingerprint-reveal-personality/
[14] Fingerprint Recognition.pdf - FBI

Stemherkenning

Met stemherkenning kan met de stem van een persoon zijn identiteit geverifieerd worden. De stem van ieder persoon is uniek, kan bewust of onbewust veranderen. Uit onderzoek is gebleken dat de stem weinig karakteristieken heeft om nauwkeurig vast te stellen wat de identiteit van een persoon is. De omgeving speelt ook een grote rol in bij stemherkenning. Als een persoon zich in een luidruchtige omgeving bevindt kan het lastig zijn de identiteit van de persoon vast te stellen. Dit neemt de betrouwbaarheid van stemherkenning weg.[15]

Een Raspberry Pi heeft geen audio-chip en er bestaan niet veel voorbeelden van de identificatie via stemherkenning. Er is besloten stemherkenning niet verder te onderzoeken als mogelijk MFA voor het project, omdat er betere biometrische authenticatie methodes bestaan die een hoog betrouwbaarheidspercentage hebben.

Gezichtsherkenning

Er bestaan meerdere manieren voor het identificeren van een persoon met de eigenschappen van het gezicht. Met behulp van een eerdere opname wordt er bepaald of het de juiste persoon is. De gevonden manieren zijn:

- **Verhoudingen in de gezicht** – Het herkennen van verhoudingen van de oren, ogen, neus en mond.
- **Temperatuur** – De temperatuur in het gezicht herkennen

Gezichtsherkenning die wordt gedaan met temperatuur wordt gedaan met infraroodcamera's. Deze camera's zijn nog te duur en zijn daarom geen optie in het gebruik binnen dit project. Gezichtsherkenning dat verhoudingen in het gezicht vergelijkt leek mogelijk te zijn. Er bestaan camera's die niet te duur zijn en tegenwoordig zitten camera's ook ingebouwd in smartphones.[16] Deze kunnen gebruikt worden voor het project.

[15] https://www.security.nl/posting/24772/Stemherkenning%3A+biometrie+op+afstand
[16] https://www.intechopen.com/books/reviews-refinements-and-new-ideas-in-face-recognition/thermal-infrared-face-recognition-a-biometric-identification-technique-for-robust-security-system

6.2 Requirements vaststellen

Met een combinatie van het onderzoek dat in de definitiefase is gedaan en interviewgesprekken die met de opdrachtgever zijn gehouden (zie bijlagen, “definitierapport - onderzoek” en “Interviewverslagen”) zijn de eisen van het project worden vastgesteld. In Tabel 6.3 worden de requirements van het project beschreven. Een eis bevat een ID, omschrijving, prioriteit en mogelijk commentaar. De prioriteit wordt toegekend met behulp van de MoSCoW-methode. Er is voor deze methode gekozen, omdat de student er ervaring mee heeft.

ID	Omschrijving	Prioriteit (MoSCoW)	Commentaar
REQ01	Het POC bestaat uit een Publieke dienst	M	
REQ02	Het POC heeft een Raspberry Pi die data versleutelt en ontsleutelt op de publieke dienst	M	Dit wordt op een RPI3 ontwikkeld en wordt de "Security-Pi" genoemd
REQ03	Het POC heeft een MFA	M	
REQ04	Er mogen geen aanpassingen worden gedaan aan de publieke dienst	M	
REQ05	Op de publieke dienst kan worden ingelogd met een gebruikersnaam en wachtwoord	M	
REQ06	De publieke dienst kan berichten plaatsen	M	
REQ07	De publieke dienst kan geplaatste berichten weergeven	M	
REQ08	De publieke dienst kan nieuwe gebruikers registreren	M	
REQ09	De Security-Pi gebruikt een AES-256 encryptie voor data	M	Data die opgeslagen wordt op publieke dienst
REQ10	De Security-Pi kan de versleutelde data ontsleutelen en weergeven	M	
REQ11	Op de Security-Pi kan worden ingelogd met gebruikersnaam en wachtwoord	M	
REQ12	Op de Security-Pi kan worden ingelogd met het MFA-apparaat	S	
REQ13	Op de Security-Pi kan een MFA-apparaat worden geregistreerd	S	
REQ14	De MFA heeft een krachtige bescherming tegen phishing-aanvallen	S	
REQ15	De MFA is een fysiek apparaat	S	USB of Smartphone
REQ16	De MFA kan niet makkelijk worden vergeten	S	"Een USB vergeet je makkelijker dan een Smartphone"
REQ17	De MFA heeft een laag energieverbruik	S	
REQ18	De MFA kan biometrisch identificeren	C	

Tabel 6.3 Requirements van het proof of concept

6.3 Mogelijke Opstellingen

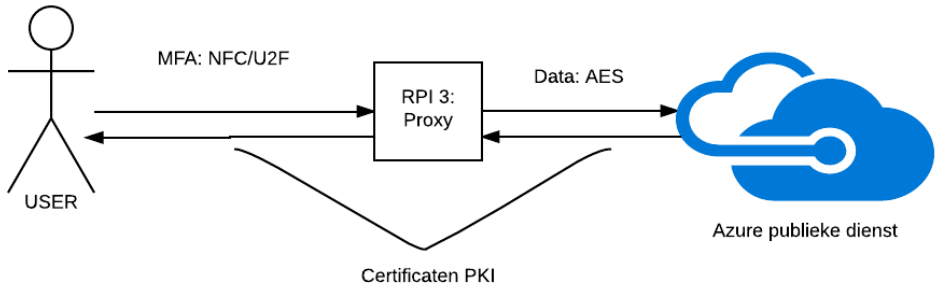
Met behulp van het onderzoek en de requirements kunnen er mogelijke opstellingen worden gemaakt. In deze paragraaf worden de mogelijke opstellingen besproken die zijn gekozen voor een mogelijke systeemarchitectuur voor het proof of concept.

De volgende opstellingen lijken mogelijk voor dit project:

- Opstelling 1: Certificaten PKI & U2F-Stick
- Opstelling 2: Certificaten PKI & U2F-BLE
- Opstelling 3: Certificaten PKI & Bluetooth Klassiek

Opstelling 1: Certificaten PKI & U2F-Stick

In de eerste opstelling wordt er een Certificaten PKI-omgeving opgezet waarop de proxy-dienst van de RPI3 wordt uitgevoerd. De data die verzonden wordt naar de publieke dienst wordt met AES versleuteld. De gebruiker authenticeert zich met een gebruikersnaam, wachtwoord en een U2F-stick.



Er is een lijst opgesteld die duidelijk de voor- en nadelen weergeeft van de eerste opstelling:

Voordelen opstelling 1:

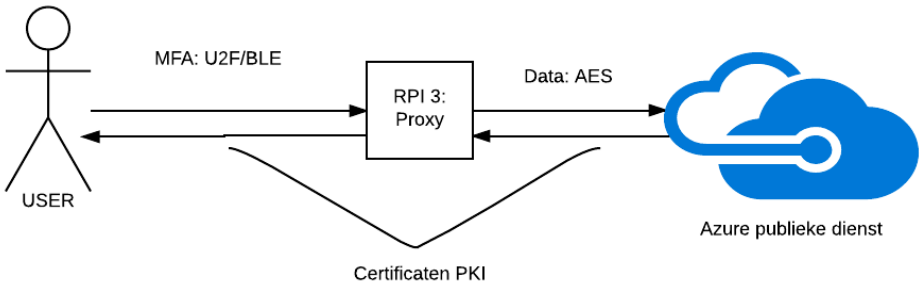
- Veiligheid van U2F
- Korte afstand (Apparaat moet fysiek in Raspberry Pi)

Nadelen opstelling 1:

- Fysiek apparaat dat makkelijk kan worden vergeten in de RPI

Opstelling 2: Certificaten PKI & U2F/BLE

De tweede opstelling wordt er een certificaten PKI gebruikt met een combinatie van BLE en U2F. BLE verreist weinig energie, maar is zo uitgekleed dat het een te onveilig protocol is om te gebruiken als een MFA. Door het toevoegen het U2F-protocol wordt er wel een sterke authenticatie toegevoegd.



Lijst van voor- en nadelen bij het gebruik van opstelling 2 in dit project:

Voordelen opstelling 2:

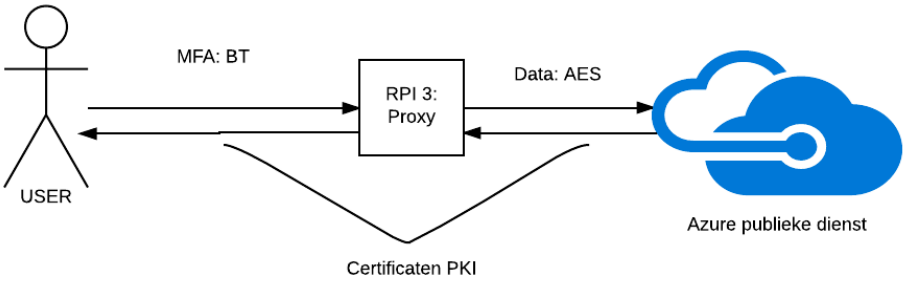
- Redelijk korte afstand. (Kan worden aangepast)
- De gebruiker heeft niet de mogelijkheid de sleutel te laten liggen bij het apparaat. (Bijv. usb-sleutel in de RPI3 laten)
- Een sterke MFA, maar de Usability blijft goed te gebruiken.
- Een telefoon is lastiger te klonen, stelen of vergeten.
- Low- energy, het vergt weinig stroom van authenticatie apparaat.

Nadelen opstelling 2:

- Er bestaat nog geen werkend product van op de markt

Opstelling 3: Certificaten PKI & Klassieke Bluetooth

In de derde opstelling wordt er een certificaten PKI gebruikt in combinatie met klassieke Bluetooth. Klassiek Bluetooth is een veilig te gebruiken protocol, maar verreist veel energie van het MFA-apparaat.



Een lijst van voor- en nadelen bij het gebruik van opstelling 3 binnen dit project.

Voordelen opstelling 3:

- Het protocol is op zichzelf een veilig protocol, er hoeven geen ingewikkelde methodes/protocollen worden toegevoegd.

- Usability – het protocol is makkelijk te gebruiken en veel hedendaagse apparaten hebben het.

Nadelen opstelling 3:

- Het vergt veel energie van het authenticatie apparaat.

6.4 Bepalen definitieve architectuur

In deze paragraaf worden de mogelijke opstellingen vergeleken met elkaar en een opstelling geselecteerd die ontwikkeld zal worden als proof of concept. De opstellingen zijn vooral gefocust op de MFA in de opstellingen. De eisen van de publieke dienst en Security-Pi zijn duidelijk. De requirements voor de afweging zijn geselecteerd met opdrachtgever en daaruit is een keuze gekomen. In Tabel 6.4 wordt de afweging van de verschillende opstellingen weergegeven.

Requirements / Opstelling	1: Certificaten PKI & U2F – Stick	2: Certificaten PKI & U2F/BLE - Smartphone	3: Certificaten PKI & Klassieke Bluetooth - Smartphone
REQ14	+++	+++	++
REQ15	+++	++	++
REQ16	+	++	++
REQ17	+++	++	+
REQ18	+	+++	+++
Totaal +	**11**	**12**	**10**

Tabel 6.4 Afweging van mogelijke opstellingen | + = Onvoldoende ++ = Voldoende +++ = Goed

Uit de afweging is opstelling 2 als het meest geschikt voor het proof of concept. Van dit product bestaan niet veel voorbeelden en is nog in ontwikkeling bij een bedrijven[17]. De MFA in opstelling 2 is een samenvoeging het U2F protocol in combinatie met BLE. De Raspberry-Pi 3 ondersteunt Bluetooth 4.1 wat niet zo veilig bleek te zijn. Door het toevoegen van het U2F-protocol wordt het juist een sterk MFA.

Om dit werkend te krijgen is besloten het product op te bouwen in delen. Als eerste wordt het BLE-gedeelte ontwikkeld. Vervolgens wordt het U2F gedeelte ontwikkeld en als laatst worden deze twee samengevoegd tot een MFA voor het Security-Pi systeem.

[17] Yubico, BLE/U2F -- https://www.yubico.com/2016/06/yubikey-u2f-tracking-bluetooth-maturity/

6.5 Bepalen incrementen

Incrementen konden niet meteen bepaald worden. Er moest eerst onderzoek gedaan worden voordat er bepaald kon worden hoe de incrementen gevuld zouden worden. Aan het eind van de definitiefase zijn de incrementen opgesteld die worden beschreven in Tabel 6.5.

Increment	Beschrijving	Activiteiten
Increment 1: Publieke dienst opzetten	In deze increment wordt een publieke dienst opgezet, zodat de werking van het proof of concept kan worden weergegeven.	- Ontwerpen omgeving - Realiseren omgeving - Testen van realisatie omgeving
Increment 2: Ontwikkelen Security-Pi	Er wordt een security-pi ontwikkeld die data van de gebruiker kan versleutelen en ontsleutelen op de publieke dienst.	- Ontwerpen werking van Security- Pi - Realisatie van de Security-Pi - Testen Security - Pi
Increment 3: Opzetten MFA	Deze increment wordt het ontwikkelen van de MFA die uit het onderzoek als meest geschikte is gekomen.	- Ontwerpen werking MFA - Realisatie van ontwerp MFA - Testen realisatie MFA

Tabel 6.5 Incrementen beschrijving

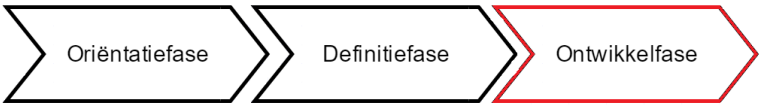
7 Ontwikkelfase

In dit hoofdstuk wordt de laatste fase van project beschreven, de ontwikkelfase. De activiteiten die verricht worden zijn ontwerpen, realiseren en testen van het proof of concept. Deze werkzaamheden zijn in incrementen opgeleverd. Met de ontwerpen wordt de werking van het systeem duidelijk gemaakt. Vanuit de ontwerpen kan er code worden gerealiseerd. Dit wordt gedaan in de realisatie activiteit. Als laatste wordt de implementatie getest met behulp van scenario's die zijn opgesteld in de ontwerpactiviteit. De ontwikkelfase verdeeld als volgt:

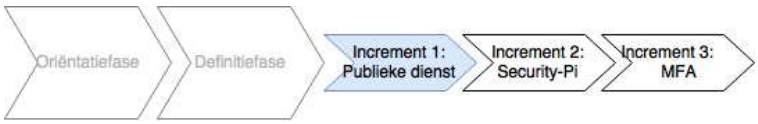
Ontwikkelfase:

- Increment 1: Publieke dienst
- Increment 2: Security-Pi
- Increment 3: MFA

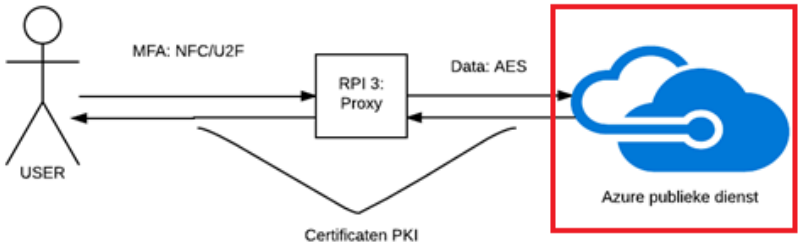
Het volledige verslag van de ontwikkelfase is beschreven in het ontwikkelrapport (zie bijlage, "Ontwikkelrapport").



7.1 Increment 1: Publieke Dienst opzetten



In het eerste increment wordt er een publieke dienst opgezet in Azure om de werking van het proof of concept te bewijzen. In Figuur 7.1 wordt weergegeven welk gedeelte van het proof of concept in het eerste increment wordt ontwikkeld. De opdrachtgever wilde in eerste instantie een bestaande publieke dienst gebruikten zoals Facebook of Twitter. Er is onderzoek gedaan naar de mogelijkheid ervan en er kwam toen uit dat dit niet mogelijk is. Er is toen een beslissing genomen om zelf een publieke dienst op te zetten op Azure van ICT Group. Er wordt als eerste een ontwerp gemaakt van de werking van de publieke dienst. Als het ontwerp duidelijk is kan er worden gezocht naar een bestaande open-source applicatie die voldoet aan het ontwerp. Als er geen applicatie gevonden is die voldoet aan deze eisen zal er zelf een publieke dienst moeten worden ontwikkeld.



Figuur 7.1 Proof of concept - increment 1: Azure publieke dienst

Ontwerp publieke dienst

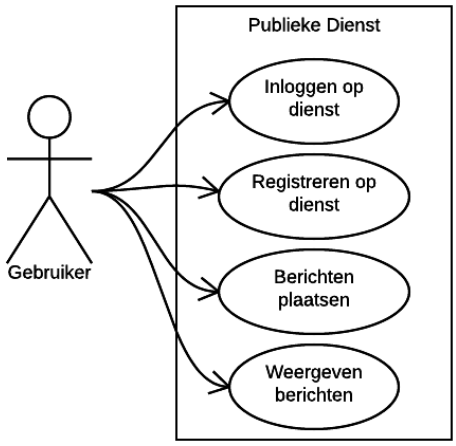
Het eerste activiteit in het opzetten van de publieke dienst is het zo goed mogelijk in kaart brengen van de eisen van de publieke dienst. In Tabel 7.1 worden de eisen beschreven die gaan over hoe de publieke dienst is opgebouwd. Deze eisen worden omgezet naar een ontwerp van de publieke dienst.

ID	Omschrijving
REQ01	Het POC bestaat uit een Publieke Dienst
REQ04	Er mogen geen aanpassingen worden gedaan aan de publieke dienst
REQ05	De publieke dienst kan worden ingelogd met een gebruikersnaam en wachtwoord
REQ06	De publieke dienst kan berichten plaatsen
REQ07	De publieke dienst kan geplaatste berichten weergeven
REQ08	De publieke dienst kan nieuwe gebruikers registreren

Tabel 7.1 requirements publieke dienst

Use-Case Diagram Publieke Dienst:

Met een Use-Case diagram wordt werking van het systeem globaal gevisualiseerd. Met behulp van de opgestelde requirements is er een Use-Case diagram gemaakt. In Figuur 7.2 wordt de Use-Case diagram van de publieke dienst weergegeven. De gebruiker kan zich laten registreren en inloggen op de publieke dienst. Een gebruiker kan ook de geplaatste berichten weergeven en berichten plaatsen.



Figuur 7.2 Use-Case Diagram - Publieke Dienst

Scenario's Publieke Dienst

Met behulp van de Use-Case diagram in Figuur 7.2 zijn er scenario's voor elke case gemaakt. Een scenario beschrijft het doel, pre-conditie, activiteiten en alternatieven. Voor dit verslag worden er twee scenario's beschreven, voor een volledige beschrijving van alle scenario's wordt er verwezen naar het ontwikkelrapport (zie bijlage, "Ontwikkelrapport").

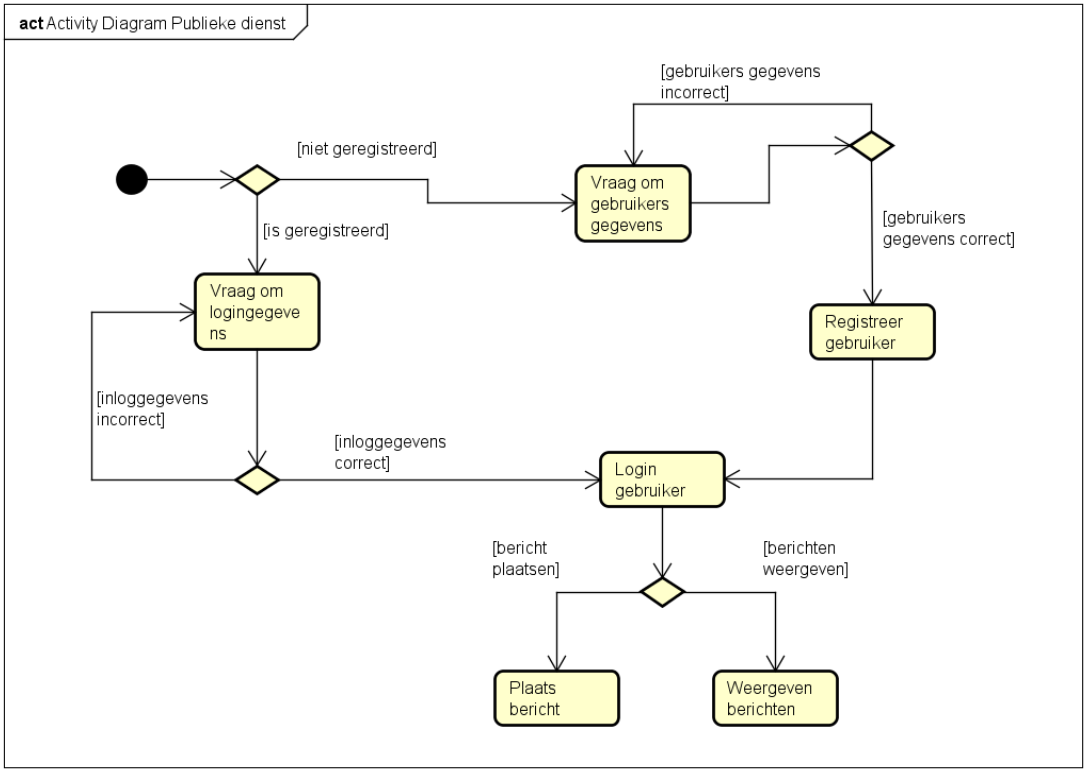
IC1 – SC1	Inloggen gebruiker
Doel	Het publieke systeem logt de gebruiker in
Pre-Condities	De gebruiker is geregistreerd
Activiteiten	1. De gebruiker voert inloggegevens in 2. Het systeem controleert de gegevens

	3. Het systeem logt de gebruiker in
Alternatieven	1. De ingevoerde gegevens van de gebruiker kloppen niet 2. De gebruiker wordt teruggestuurd naar het inlogscherm

IC1 – SC3	Bericht plaatsen
Doel	De gebruiker plaatst een bericht op het publieke systeem
Pre-Condities	De gebruiker is ingelogd
Activiteiten	1. De gebruiker voert een bericht in 2. De gebruiker geeft aan dat het bericht gedeeld mag worden 3. Het systeem bewaart het bericht in een database
Alternatieven	Geen

Activiteitsdiagram Publieke Dienst

Met een activiteitsdiagram worden de mogelijke activiteiten van het systeem weergegeven. Door een proces van keuzes kan een systeem in zo een toestand terecht komen. Met behulp van de scenario's die eerder zijn opgezet is er een activiteitsdiagram gemaakt. In Figuur 7.3 wordt het activiteitsdiagram weergegeven van de publieke dienst. Als de gebruiker het systeem opstart kan hij een keuze maken om in te in te loggen met inloggegevens of zich laten registreren als gebruiker. Beide routes leiden tot een ingelogde gebruiker. Waarna de gebruiker kan kiezen om een bericht te plaatsen of het weergeven van berichten.



Figuur 7.3 Activiteitsdiagram - Publieke dienst

Realisatie publieke dienst

Deze activiteit beschrijft de implementatie van de publiek dienst. Doormiddel van de ontwerpen uit de vorige activiteit zal er gezocht worden naar een mogelijk systeem dat kan functioneren als publieke dienst in dit project. Als tweede zal de werking van de gekozen publieke dienst worden uitgelegd. Het derde gedeelte gaat over de deployment van het publieke systeem.

Mogelijke Publieke diensten

Het is voor de opdrachtgever niet interessant wat voor een publieke dienst wordt opgezet. Er is gekeken naar bestaande mogelijke publieke diensten. Als de publieke dienst maar voldoet aan de voorwaarden die in de ontwerpactiviteit is besproken.

- Registreren
- Inloggen
- Berichten plaatsen
- Berichten weergeven

De volgende mogelijke publieke diensten zijn gevonden.

Dienst	Beschrijving
Minitwit	Een mini twitter kloon ontwikkelt in Python Flask
Firefeed	Een open-source twitter kloon ontwikkelt in Firebase
Scaffeinate Socify	Een social-network platform ontwikkelt in Ruby on Rails

Minitwit is gekozen als meest geschikt voor het project. Het Minitwit project is ontwikkeld in Python, daar heeft de student heeft een voorkeur voor vanwege ervaring met de programmeertaal. Met deze ervaring hoeft er geen onderzoek worden gedaan naar de werking van de programmeertaal. Dit bespaart tijd in het project. Binnen het project wordt Python 2.7 gebruikt, vanwege de grote compatibiliteit met externe modules en library's. Het komt soms voor dat deze niet compatibel zijn met Python 3.x.

Werking Minitwit

Minitwit is een open-source Twitter kloon die is ontwikkeld in Python. Het framework dat Minitwit gebruikt is Flask. Flask is simpele python framework die een uitgebreide documentatie heeft met uitleg in het gebruik van Flask. Voordat Minitwit geïmplementeerd kon worden op Azure is er een lokale Virtuele Machine met een Ubuntu server 16.04 opgezet. Er is wel alvast toegang tot Azure van ICT Group opgevraagd, zodat als het lokaal is gelukt om Minitwit te implementeren het minder tijd kost deze implementatie over te zetten naar Azure.

De werking van Minitwit is eigenlijk de werking van Flask uitleggen. In plaats van code uit Minitwit wordt een klein Flask applicatie opgezet en beschreven. In Kader 7.1 wordt een voorbeeld van een Python Flask applicatie weergegeven. Bij elke applicatie is het belangrijk dat Flask geïmporteerd wordt in het project, zodat er met het Flask framework gewerkt kan worden. Als tweede wordt er aan de Flask applicatie een naam toegekend. Als de applicatie aangeroepen moet worden kan dat met "app" gedaan worden. Het is belangrijk dat elke html pagina een python functie krijgt, ook al gebeurt er niks aan de server zijde dan nog moet deze gedefinieerd zijn. Deze functie geeft een html pagina terug. Boven elke functie staat een @app.route('pagina') dat de URL van de betreffende pagina weergeeft.

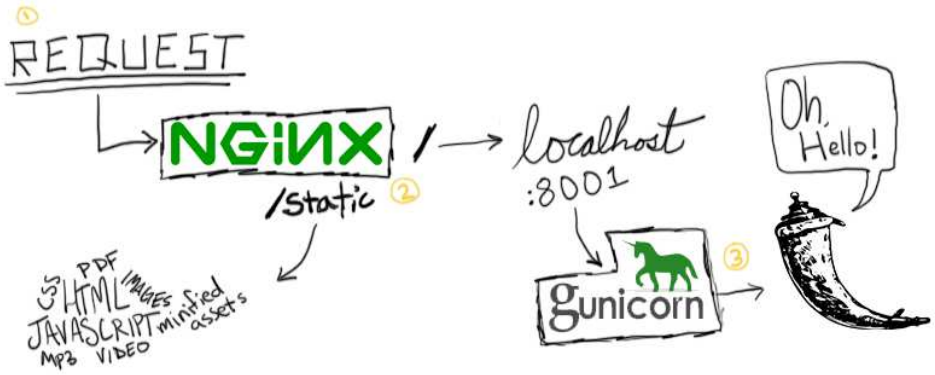
```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def index():
#[pythoncode dat de pagina kan uitvoeren]
   return render_template ('Index.html')
```

Kader 7.1 Hello World! Flask applicatie

Deployment Minitwit

De opdrachtgever wil graag de publieke dienst in een Azure omgeving laten implementeren voor het proof of concept. Er is toegang verleent aan de student tot de Azure omgeving van ICT Group. Op Azure is er een virtuele machine (VM) opgezet met Ubuntu Server 16.04 als operating systeem. Hierop zijn alleen de benodigde python packages geïnstalleerd en de Minitwit applicatie overgezet. Als eerst om de applicatie werkend te krijgen is de Flask package geïnstalleerd. Nu was het mogelijk de applicatie op te starten op de VM, maar als de terminal gesloten werd werkte ook de applicatie niet meer. De applicatie moet constant blijven werken. Hiervoor is een manier gevonden met behulp van Nginx en Gunicorn.

In Figuur 7.4 [18] wordt de werking van Nginx, Gunicorn en Flask weergegeven. Nginx is een http-webserver en reverse proxy dat gebruikt wordt om websites op de hosten. De Nginx webserver kan niet communiceren met Python applicaties en Minitwit is een python geschreven Flask applicatie. Flask ondersteund het Web Server Gateway Interface (WSGI). WSGI zorgt voor de communicatie tussen Nginx en Flask. Er was een medium nodig die voor de communicatie tussen Nginx en de Flask-applicatie zorgt. Gunicorn is gevonden als oplossing. Gunicorn is een Python WSGI-webserver en is gebruikt als medium tussen Nginx en Flask.



Figuur 7.4 Werking Nginx, Gunicorn en Flask

[18] https://realpython.com/blog/python/kickstarting-flask-on-ubuntu-setup-and-deployment/

Testen publieke dienst

In dit hoofdstuk worden de uitgevoerde testen beschreven die in dit increment zijn uitgevoerd. Voor het testen van de publieke dienst zijn de scenario's genomen die zijn opgesteld in de ontwerpactiviteit. De scenario's zijn aangepast naar testscenario's die horen te werken op de publieke dienst. Er zullen twee beschreven worden voor dit verslag. Een volledige beschrijving van alle testscenario's wordt er verwezen naar het Testrapport (zie bijlage, "Testrapport").

In Tabel 7.2 wordt de test beschreven die betrekking hebben tot het inloggen van de gebruiker op de publieke dienst. Tabel 7.3 beschrijft het plaatsen van een bericht op de publieke dienst. Beide testen die worden weergegeven zijn geslaagd.

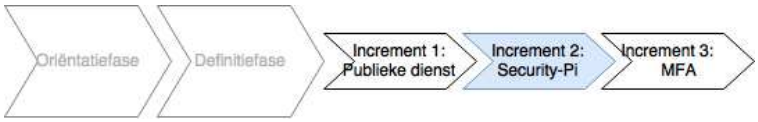
IC1 – TC1	Inloggen gebruiker
Doel	Het publieke systeem logt de gebruiker in
Pre-Condities	De gebruiker is geregistreerd
Activiteiten	4. De gebruiker voert inloggegevens in 5. Het systeem controleert de gegevens 6. Het systeem logt de gebruiker in
Alternatieven	3. De ingevoerde gegevens van de gebruiker kloppen niet 4. De gebruiker wordt teruggestuurd naar het inlogscherm
Verwachte Resultaat	De gebruiker is ingelogd op publieke dienst Minitwit
Resultaat	De gebruiker is ingelogd op publieke dienst Minitwit
Geslaagd	JA

Tabel 7.2 IC1-TC1 Inloggen gebruiker

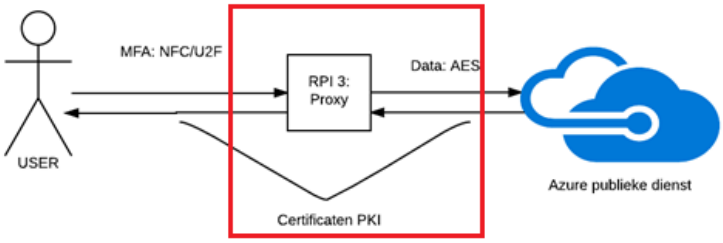
IC1 – TC3	Bericht plaatsen
Doel	De gebruiker plaatst een bericht op het publieke systeem
Pre-Condities	De gebruiker is ingelogd
Activiteiten	4. De gebruiker voert een bericht in 5. De gebruiker geeft aan dat het bericht gedeeld mag worden 6. Het systeem bewaard het bericht in een database
Alternatieven	Geen
Verwachte resultaat	Bericht is geplaatst op publieke dienst Minitwit
Resultaat	Bericht is geplaatst op publieke dienst Minitwit
Geslaagd	JA

Tabel 7.3 IC1-TC3 Bericht plaatsen

7.2 Increment 2: Ontwikkelen Security-Pi



In het tweede increment wordt de Security-Pi ontwikkeld op een Raspberry Pi 3. In Figuur 7.5wordt weergegeven welk gedeelte van het proof of concept deze increment wordt ontwikkeld. Dit is het product dat de opdrachtgever wil hebben. De security-Pi zorgt ervoor dat de data die opgeslagen wordt op de publieke dienst versleuteld is. Om deze data te bekijken heeft gebruiker een Security-Pi nodig. Er is eerst een ontwerp gemaakt van de Security-Pi zodat er een duidelijk beeld ontstaat voor het ontwikkelen van de Security-Pi. Wanneer het ontwerp af is wordt de security-pi ontwikkeld in het realisatiegedeelte. Als laatste zal de security-pi worden getest.



Figuur 7.5 Proof of concept - increment 2: Security-Pi

Ontwerp Security-Pi

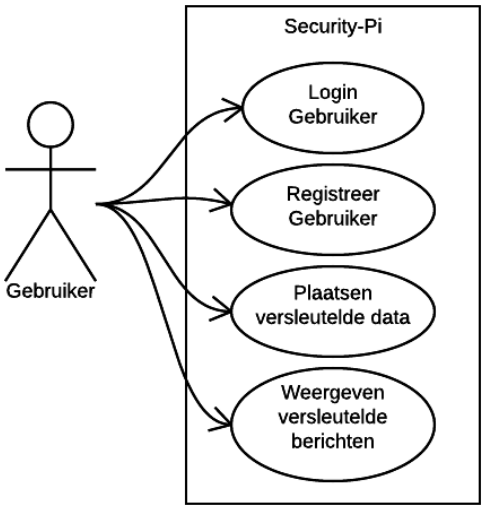
Om een ontwerp te maken van de Security-Pi zijn de requirements die in de vorige fase zijn vastgesteld erbij gehaald. In tabel worden deze eisen beschreven en daaruit is er een Use-case diagram gemaakt. Vanuit de Use-Case zijn er mogelijke scenario's opgesteld die bij de Security-Pi kunnen voorkomen. Met behulp van de Use-case en scenario's is er een activiteitsdiagram gemaakt die processen van de Security-Pi weergeeft. Als laatst zijn er sequentiediagrammen gemaakt die een beter beeld geven van de scenario's en het proces dat het systeem moet uitvoeren. De volledige ontwerpactiviteit van increment 2 staat beschreven in het ontwikkelrapport (zie bijlage, "Ontwikkelrapport – increment 2: Security – Pi")

ID	Omschrijving Requirements
REQ09	De Security-Pi gebruikt een AES-256 encryptie voor data
REQ10	De Security-Pi kan de versleutelde data ontsleutelen en weergeven
REQ11	Op de Security-Pi kan worden ingelogd met gebruikersnaam en wachtwoord
REQ12	Op de Security-Pi kan worden ingelogd met het MFA-apparaat (Wordt ontwikkeld in increment 3)
REQ13	Op de Security-Pi kan een MFA-apparaat worden geregistreerd (Wordt ontwikkeld in increment 3)

Tabel 7.4 Requirements Security-Pi

Use-Case Diagram Security-Pi

In Figuur 7.6 wordt een Use-Case diagram weergegeven van het Security-Pi systeem. Het systeem bestaat uit soortgelijke cases als de publieke dienst. De cases zijn inloggen en het registreren van een gebruiker. De andere cases zijn het plaatsen en weergeven van versleutelde diensten.



Figuur 7.6 Use-Case Diagram - Security-Pi

Scenario's Security-Pi

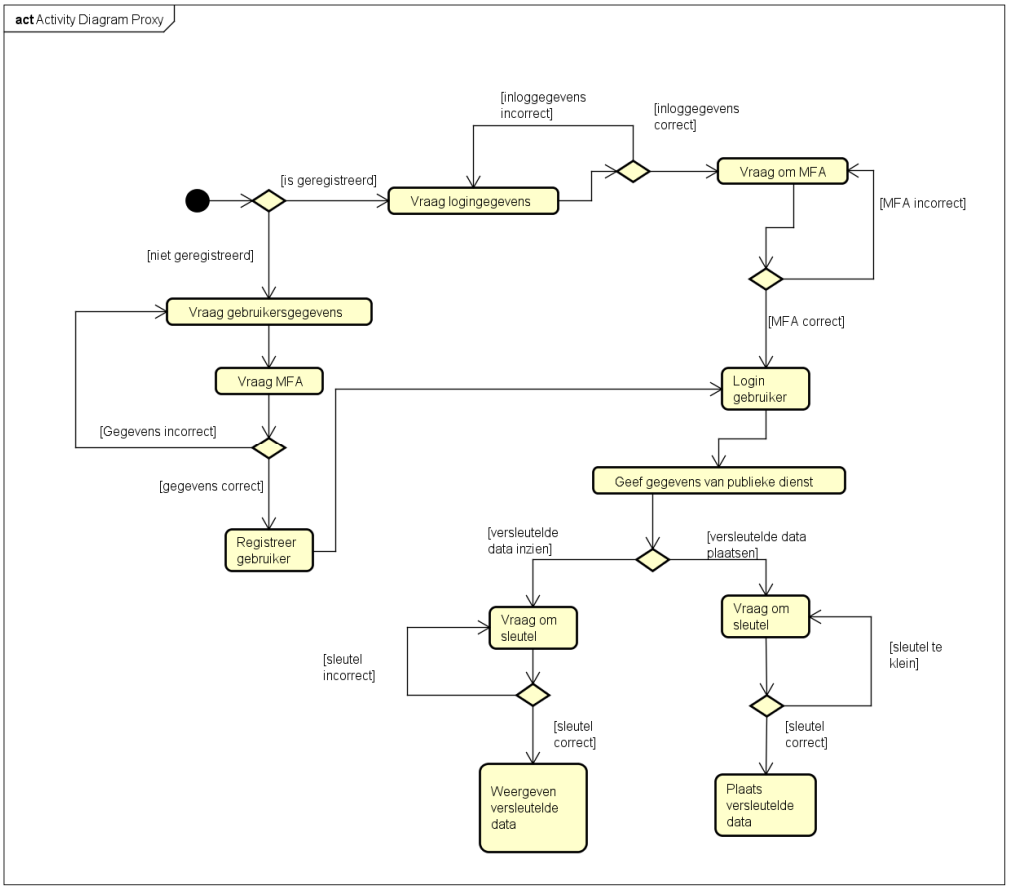
Ook in deze iteratie zijn er scenario's opgesteld die het doel, pre-conditie, activiteiten en alternatieven beschrijven. Er worden twee scenario's beschreven, voor een volledige beschrijving van alle scenario's wordt verwezen naar de bijlage "Ontwikkelrapport".

IC2 – SC1	Inloggen gebruiker op security-pi
Doel	De gebruiker logt in op het Security-Pi systeem
Pre-Condities	De gebruiker is geregistreerd op het Security-Pi systeem
Activiteiten	1. De gebruiker voert inloggegevens in 2. De gebruiker voert MFA uit (Wordt in increment 3 ontwikkeld) 3. Het systeem logt de gebruiker in
Alternatieven	1. De ingevoerde inloggegevens kloppen niet 2. De uitgevoerde MFA klopt niet

IC2 – SC3	Versleutelde bericht plaatsen
Doel	De gebruiker plaatst een versleuteld bericht op het Security – Pi systeem
Pre-Condities	De gebruiker is ingelogd op het Security-Pi Systeem
Activiteiten	1. De gebruiker geeft aan dat er een versleuteld bericht geplaatst moet worden 2. De gebruiker voert bericht in 3. Het systeem versleutelt het bericht 4. Het systeem plaatst bericht op publieke dienst
Alternatieven	Geen

Activiteitsdiagram Security-Pi

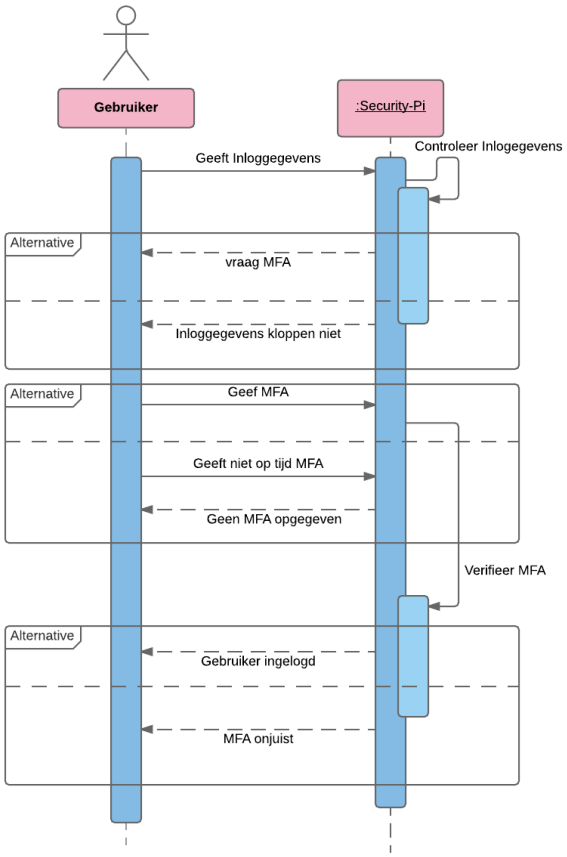
In Figuur 7.7 wordt een activiteitsdiagram weergegeven van de Security-Pi. Als een gebruiker dit systeem wil gebruiken dan kan er een keuze worden gemaakt om in te loggen of te registreren. Bij het inloggen wordt er gevraagd om inloggegevens en een MFA. Als beide ingevoerde gegevens kloppen dan wordt de gebruiker ingelogd. Bij het registreren wordt er om registratiegegevens gevraagd en MFA zodat beide geregistreerd worden. Als het registratieproces goed is verlopen is de gebruiker ook ingelogd. Een ingelogde gebruiker kan versleutelde data plaatsen en weergeven.



Figuur 7.7 Activiteitsdiagram - Security-Pi

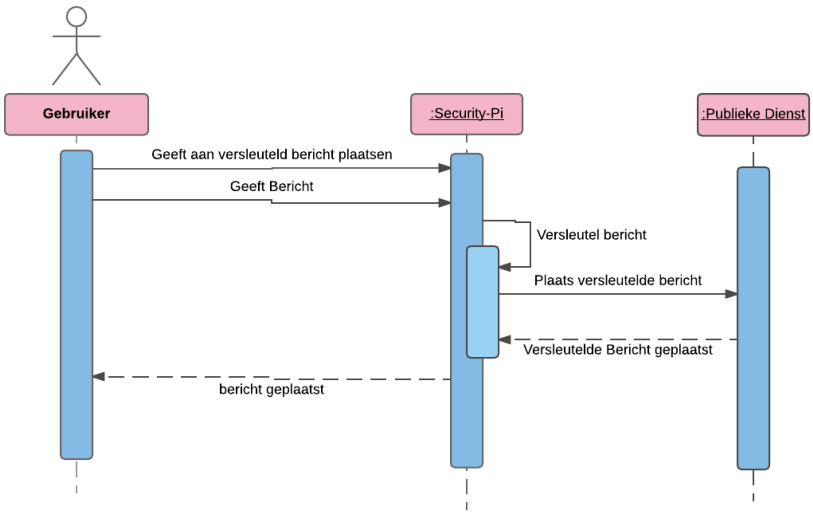
Sequentie Diagrammen Security-Pi

In Figuur 7.8 wordt een sequentiediagram weergegeven van het inloggen van een gebruiker op de Security-Pi. Een gebruiker geeft zijn inloggegevens op en het Security-Pi systeem controleert deze gegevens. Als de inloggegevens niet kloppen dan geeft het Security-Pi systeem aan de gebruiker een melding dat de opgegeven inloggegevens niet kloppen. Als de inloggegevens wel kloppen vraagt het Security-Pi systeem aan de gebruiker om een MFA uit te voeren. Wanneer er niet op tijd een verificatie wordt opgegeven geeft de Security-Pi een melding aan de gebruiker dat er niet op tijd een MFA is opgegeven. Als de gebruiker wel een verificatie via het MFA-apparaat heeft opgegeven wordt deze gecontroleerd. Als het MFA klopt logt het systeem de gebruiker in en stuurt een bericht naar de gebruiker dat het inloggen is gelukt. Wanneer authenticatie via het MFA-apparaat niet klopt geeft het Security-Pi systeem dit aan bij de gebruiker.



Figuur 7.8 Sequentie Diagram - Security-Pi - IC2-SC1 Inloggen gebruiker

In Figuur 7.9 wordt een sequentiediagram weergegeven dat het plaatsen van een versleutelde bericht weergeeft. De gebruiker geeft als eerst aan dat hij een versleutelde bericht wilt plaatsen en geeft dan het bericht door dat versleuteld moet worden. Versleutelt de Security-Pi het bericht en vervolgens plaatst de Security-Pi het versleutelde bericht op de publieke dienst en die geeft aan dat het bericht is geplaatst. De Security-Pi geeft aan de gebruiker door dat het versleutelde bericht geplaatst is.



Figuur 7.9 Sequentie Diagram - Security-Pi – IC2-SC3 Versleuteld bericht plaatsen

Realisatie Security-Pi

In deze activiteit wordt het realisatiegedeelte van de Security-Pi beschreven. In de vorige activiteit zijn er ontwerpen van het Security-Pi systeem gemaakt, deze worden nu ontwikkelt tot een werkend systeem. Als eerst zal de implementatie het versleutelingsscript worden beschreven, daarna de implementatie van de Raspberry-Pi als proxy.

Implementatie versleuteling script

Er is als eerst gekeken naar het versleutelen van data met een AES-encryptie, daarna naar mogelijke manieren om de versleutelde data te plaatsen op de publieke dienst. Er zijn verschillende python cryptografie library's gevonden die mogelijk gebruikt kunnen worden in dit project. De gevonden cryptografie library's zijn:

Cryptografie Library's	Beschrijving
PyCrypto	Een Python cryptografie toolkit
M2Crypto	Python wrapper voor OpenSSL
Cryptography	Python library voor cryptografische algoritmes

Tabel 7.5 Gevonden cryptografie library's

PyCrypto

PyCrypto is de oudste en een nog veel gebruikte cryptografie toolkit als het gaat om cryptografie met Python. Veel van de hedendaagse Python library's en toolkits zijn gebaseerd op PyCrypto. Voorbeeld van een op PyCrypto gebaseerde python cryptografie library's is Google's Keyczar. PyCrypto is alleen niet zo actief en up-to-date de originele ontwikkelaar is gestopt met support van de toolkit. Een voordeel van PyCrypto is doordat het langer bestaat veel voorbeelden bestaan in het gebruik ervan. PyCrypto is dan ook gebruikt als cryptografie library in het project.

M2Crypto

M2Crypto module is een Python wrapper om de originele c-gebaseerde OpenSSL library. M2Crypto gebruikt SWIG om Python te koppelen aan OpenSSL. Het nadeel van M2Crypto is dat het niet goed gedocumenteerd is. Vanwege onvoldoende documentatie en voorbeelden is er niet verder gekeken naar het gebruik van M2Crypto.

Cryptography

Cryptography is een nieuwer, maar een veel actiever protocol. Er komt regelmatig een nieuwe release van uit en de documentatie is heel uitgebreid. Dit maakt Cryptography het meest aantrekkelijk in gebruik. De library maakt gebruikt van CFFI, een nieuwe manier van C-code aanroepen vanaf Python.

De Cryptography module kwam pas later tijdens het project bekend, er was nog met de opdrachtgever gesproken over het aanpassen van het versleutelingsscript, maar er werd aangeraden om dat niet te doen. Het is werkend en als er voldoende tijd overblijft kan dit aangepast worden als het nodig is. PyCrypto is geen slechte keuze, omdat deze iets minder actief en ouder is. PyCrypto bestaat veel langer en dat geeft juist meer betrouwbaarheid in het gebruik ervan. In het ontwikkelrapport wordt de implementatie van het versleutelingsscript weergegeven (zie bijlage, "Ontwikkelrapport").

Implementatie Certificaat

Voor het proof of concept is er een certificaat gegenereerd, zodat data dat over de verbinding tussen de gebruiker en Security-Pi heengaat niet onderschept kan worden. Dit is gedaan met behulp van OpenSSL. Met OpenSSL kan makkelijk een certificaat gegenereerd. In Figuur 7.10 wordt het proces van het genereren van het certificaat weergegeven. Het is een simpel proces door de gegevens van het type certificaat in te vullen die nodig is. In dit geval is er een ECDSA-certificaat gegenereerd met de daarbij behorende sleutel.

```
pi@raspberrypi:~ $ openssl ecparam -genkey -out eckey.pem -name prime256v1
pi@raspberrypi:~ $ openssl req -x509 -new -key eckey.pem -out cert.pem
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:NL
State or Province Name (full name) [Some-State]:Zuid-Holland
Locality Name (eg, city) []:Barendrecht
Organization Name (eg, company) [Internet Widgits Pty Ltd]:ICT Group
Organizational Unit Name (eg, section) []:T&L
Common Name (e.g. server FQDN or YOUR name) []:Ramiro Jagai
Email Address []:ramiro.jagai@ict.nl
pi@raspberrypi:~ $
```

Figuur 7.10 Genereren van certificaat

Webparsing en Webscraping

Het versleutelen en ontsleutelen is werkend. Er gezocht naar manieren om de versleutelde data te plaatsen en op de publieke dienst en ook weer te verzamelen. De volgende tools zijn gevonden die mogelijk lijken voor dit project:

Tools	Beschrijving
Scrapy	Een framework voor webcrawling, scraping en parsing
BeautifulSoup	Een python library alleen voor webparsing
Selenium	Een open-source automated testing suite voor webapplicaties

Tabel 7.6 Gevonden tools voor verzamelen en verzenden van data van/naar publieke dienst

Scrapy

Scrapy is een open source python framework voor webcrawling en webscraping. Webcrawler worden ook wel Spiders genoemd. Deze Spiders worden gebruikt om automatisch specifieke data van het internet te halen. Er wordt een lijst van URL's meegegeven en de Spider gaat ze allemaal af, maar bewaard alle links die het kan vinden op een website en voegt deze toe aan de lijst die het af moet gaan. Webscraping is het verzamelen van data die de Spider vindt op de websites. Het framework Scrapy kan mogelijk worden gebruikt voor dit project. Er zou een Spider gecreëerd kunnen worden die de versleutelde data onderscheid van de normale data.

Beautiful Soup

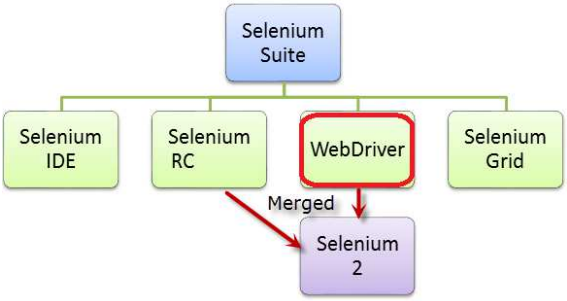
Beautiful Soup is een webparsing python library dat zich richt op een bepaalde website. Beautiful Soup "crawlt" niet naar andere websites die het tegenkomt en gaat alleen de websites af die expliciet zijn aangegeven. Het is dan alleen mogelijk data te verzamelen van de opgegeven websites. Het voordeel bij het gebruik binnen dit project is er hoeft niet een groot framework worden gebruikt dat grotendeels alleen gebruikt wordt voor één website. Het probleem van het extracten van data zou mogelijk Beautiful Soup worden toegepast, maar het was nog altijd niet duidelijk hoe data op de publieke dienst zou worden geplaatst.

Selenium

Selenium Suite is een groot applicatiepakket dat zich richt op het automatisch testen van applicaties. Selenium Suite bestaat uit 4 componenten: Selenium IDE, Selenium RC, WebDriver en Selenium Grid. De componenten Selenium RC en WebDriver zijn samengevoegd en vormen Selenium 2. Selenium is een open-source en heeft verschillende ontwikkelaars.

Met de WebDriver is het mogelijk om vanuit verschillende programmeertalen automatisch handelingen uit te voeren via een webbrowser informatie te verzamelen van een bepaalde website. Dit maakt Scraping mogelijk, maar ook het verzenden van data naar de website. Hiermee zijn twee handelingen die de Security-Pi moet uit voeren opgelost. Het is mogelijk alleen de WebDriver component apart van Selenium Suite te gebruiken. Er hoeft ook niet een groot framework worden geïnstalleerd terwijl er maar een klein gedeelte in gebruik wordt genomen. In Figuur 7.11[19] wordt de opbouw van Selenium Suite weergegeven en de keuze van WebDriver.

[19] https://seleniumwithjavapython.wordpress.com/selenium/selenium-suite/



Figuur 7.11 Opbouw van Selenium Suite met keuze WebDriver

In Kader 7.2 wordt een gedeelte van het ophalen van versleutelde data weergegeven. Er wordt een publieke dienst opgegeven dat is in dit geval publieke dienst die het IP-adres http://192.168.137.150:8000/ heeft. De versleutelde berichten worden opgeslagen in de variabele "messages". De decryptie van de berichten wordt gedaan met het versleutelingsscript.

```
driver.set_page_load_timeout(30)
driver.get("http://192.168.137.150:8000/")
driver.maximize_window()
driver.implicitly_wait(20)

messages = driver.find_elements_by_id("berichten")
```

Kader 7.2 Ophalen van versleutelde data met Selenium WebDriver

Implementatie Security-Pi applicatie

Het samenvoegen van het versleutelingsscript en het webparsing/scraping-script is uiteindelijk de Security-Pi werkend. Het script heeft alleen nog geen UI. Het script is nu alleen werkend met een terminal. De klanten van de opdrachtgever kunnen niet allemaal omgaan met een terminal en het is nodig dat er een UI waarmee de klanten gemakkelijk hun data kunnen opgeven en dat de applicatie het verder afhandelt. In de vorige iteratie is er gewerkt met het Python framework Flask. Er is voldoende kennis nu om met Flask een webapplicatie op te zetten. Op de Raspberry-Pi is in Python Flask een webapplicatie opgebouwd geïmplementeerd zoals dat ook met Minitwit is gedaan.

Implementatie proxy

Het tweede gedeelte bestaat uit het opzetten van de Raspberry-Pi als een proxy, zodat een gebruiker bij het aansluiten van zijn PC aan de Raspberry gebruik maakt van de Raspberry als proxy om op het internet te komen. Als de gebruiker naar de betreffende publieke dienst wil gaan dan wordt de gebruiker doorverwezen naar de versleuteling applicatie op de Security-Pi. De volgende mogelijke manieren zijn onderzocht als mogelijk proxy voor de Security-Pi.

Proxy's	Beschrijving
Nginx	Webserver dat opgezet kan worden als proxyserver
Python Proxy	Een eigen in python ontwikkelde proxy
Squid	Speciaal bedoeld voor verschillende proxyconfiguraties

Tabel 7.7 Gevonden proxymogelijkheden

Nginx is al eerder gebruikt bij het opzetten van de publieke dienst, maar kan ook geconfigureerd worden als proxyserver. Op de Raspberry-Pi is om de Python Flask webapplicatie werkend te krijgen Nginx geïnstalleerd. Er

is gekeken naar een manier om Nginx ook te configureren als een proxy. Het voordeel om Nginx ook als proxy te configureren is de besparing van geheugen op de Raspberry-Pi. Het nadeel is dat er geen duidelijk gescheiden implementatie is van Security-Pi applicatie en de proxy. Als er iets gebeurd met het applicatiegedeelte binnen Nginx en het programma opnieuw geïmplementeerd moet worden. Dan zal onnodig ook het proxygedeelte opnieuw moeten worden geïmplementeerd

De mogelijkheid is het creëren van een eigen in Python ontwikkelde proxy. Dit heeft als voordeel dat er geen onnodig grote proxyprogramma hoeft worden geïnstalleerd. Hiermee kan ruimte worden bespaard op de Raspberry-Pi, maar dan is het niet zeker of de proxy voldoet aan veiligheid voorwaarden. Als de Python proxy ontwikkeld wordt moet daar ook op worden gelet. Voor dit project is het niet handig om daar veel tijd in te stoppen, omdat er een tijd te kort kan ontstaan voor het implementeren van het laatste increment.

Squid is een softwarepakket gericht op het opzetten van verschillende soorten proxy's. Squid wordt wereldwijd veel gebruikt en is een goed ontwikkeld proxy met een uitgebreide documentatie. Door Squid juist te configureren kunnen kan er niet makkelijk misbruik van de proxy gemaakt kunnen worden. Een andere reden om Squid te gebruiken is het scheiden van de webserver met de proxy. Nginx wordt al gebruikt als webserver voor de Flask applicatie. Als er iets misgaat met de applicatie of proxy is het lastig te achterhalen waar dat precies gebeurd. Squid is dan ook gekozen voor dit project om als proxy te functioneren.

Deployment proxy Squid

Squid is geïnstalleerd om te werken als proxy in het Security-Pi systeem. Al het verkeer van de gebruiker gaat eerst door de proxy daarna pas het internet op. Als gedetecteerd wordt dat de gebruiker naar de publieke dienst wil gaan dan wordt de gebruiker omgeleid naar de versleutelingsapplicatie op de Security-Pi. Daar krijgt de gebruiker de mogelijkheid om data in te voeren of te bekijken. Om dit duidelijk te maken aan Squid is SquidGuard gevonden. Met de SquidGuard is het mogelijk om bepaalde domeinen of IP-adressen te blokkeren of juist niet. Binnen SquidGuard was het mogelijk om het adres van de publieke dienst op te geven en de gebruiker dan meteen door te verwijzen naar de Security-Pi applicatie. Als de gebruiker nu naar het domein van de publieke dienst probeerde te gaan dan werd de applicatie van Security-Pi weergegeven, maar SquidGuard veranderde niet de naam van in de adresbalk. SquidGuard zorgt ervoor dat intern de verwijzing wordt gedaan. Deze werking van SquidGuard werkte nadelig op het proof of concept, want door deze interne verwijzing naar de Security-Pi pagina kon er niet verder worden geklikt naar andere pagina's van de Security-Pi applicatie. Het was een oneindige loop naar alleen de homepagina van de Security-Pi. Door SquidGuard weer te de-installeren en handmatig in Squid te de verwijzing aan te geven werkte de proxy wel goed. De gebruiker werd nu daadwerkelijk naar de pagina zelf gebracht en niet alleen een weergave van de homepagina van de Security-Pi applicatie.

Op dit moment werkt de proxy, maar de gebruiker moet zelf aangeven in de webbrowser dat hij gebruik wil maken van de proxy. De opdrachtgever verwacht een oplossing waar de klant zelf niks voor hoeft te doen. Een "plug&play" oplossing voor de klant. Dit houdt in dat de klant het alleen maar hoeft aan te sluiten en ermee aan de slag kan. Om dit op te lossen zal de Squid proxy transparant worden geconfigureerd. Met een transparante proxy hoeft de gebruiker niet zelf aan te geven en zijn browser dat hij gebruikt wil maken van een proxy. Er was helaas op dit moment geen tijd genoeg voor deze increment om aan te werken. Er is besproken met de opdrachtgever dit gedeelte later als er voldoende tijd is te implementeren.

Testen Security-Pi

Dit hoofdstuk beschrijft de testen die betrekking hebben tot increment 2 de Security-Pi. De scenario's die opgesteld zijn in de ontwerpactiviteit zijn aangepast naar testscenario's. Er zullen ook weer voor het verslag twee testscenario's worden beschreven. Voor een volledige beschrijving van alle testscenario's wordt er verwezen naar het Testrapport (zie bijlage, "Testrapport").

In Tabel 7.8 en Tabel 7.9 worden de testen weergegeven van het inloggen op de Security-Pi en het plaatsen van een versleutelde bericht. Beide testen zijn werkend en dus ook geslaagd.

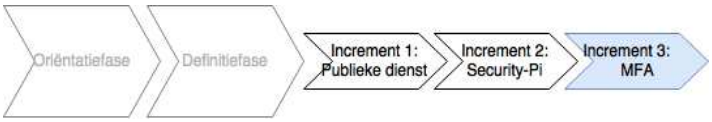
IC2 – TC1	Inloggen gebruiker op security-pi
Doel	De gebruiker logt in op het Security-Pi systeem
Pre-Condities	De gebruiker is geregistreerd op het Security-Pi systeem
Activiteiten	1. De gebruiker voert inloggegevens in 2. De gebruiker voert MFA uit 3. Het systeem logt de gebruiker in
Alternatieven	1. De ingevoerde inloggegevens kloppen niet 2. De uitgevoerde MFA klopt niet
Verwachte Resultaat	De gebruiker is ingelogd op de Security-Pi
Resultaat	De gebruiker is ingelogd op de Security-Pi
Geslaagd	JA

Tabel 7.8 IC2-TC1 Inloggen gebruiker op Security-Pi

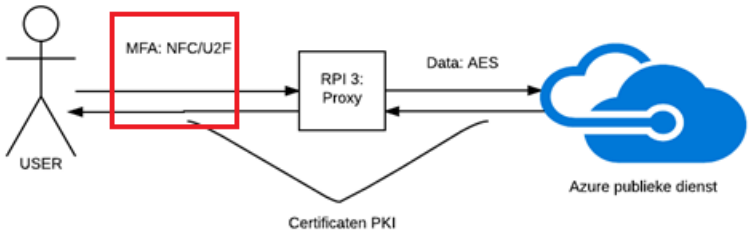
IC2 – TC3	Versleutelde bericht plaatsen
Doel	De gebruiker plaatst een versleuteld bericht op het Security – Pi systeem
Pre-Condities	De gebruiker is ingelogd op het Security-Pi Systeem
Activiteiten	1. De gebruiker geeft aan dat er een versleuteld bericht geplaatst moet worden 2. De gebruiker voert bericht in 3. Het systeem versleuteld het bericht 4. Het systeem plaatst bericht op publieke dienst
Alternatieven	Geen
Verwachte Resultaat	Er is een versleutelde bericht geplaatst via de Security-Pi op de publieke dienst Minitwit.
Resultaat	Er is een versleutelde bericht geplaatst via de Security-Pi op de publieke dienst Minitwit.
Geslaagd	JA

Tabel 7.9 IC2-TC3 Versleutelde bericht plaatsen

7.3 Increment 3: Opzetten MFA



Het derde increment voegt een MFA (multifactor authenticatie) toe aan de Security-Pi. In Figuur 7.12 wordt weergegeven welk deel van het proof of concept in deze increment wordt ontwikkeld. Er is een U2F-stick aangeschaft om de werking van het MFA te ontwikkelen en testen. Als eerst worden de eisen omgezet tot een ontwerp. De tweede activiteit is met het ontwerp een realisatie van het product ontwikkelen. Als laatste zal de werking van de realisatie worden getest. Een volledige beschrijving van de ontwikkeling van het MFA staat beschreven in het ontwikkelrapport (zie bijlage, “Ontwikkelrapport – increment 3: MFA”)



Figuur 7.12 Proof of concept - increment 3: MFA

Ontwerp MFA

In deze activiteit wordt het ontwerp van het MFA-gedeelte van de Security-Pi gemaakt. Er worden vastgestelde eisen genomen die gaan over de MFA en omgezet naar een Use-Case diagram en daarvan weer mogelijke scenario’s opgesteld. Met behulp van de Use-Case diagram en scenario’s is er een activiteitsdiagram gemaakt.

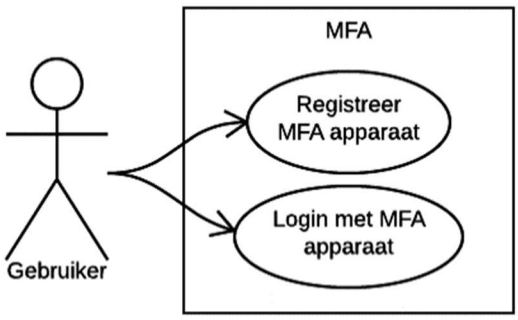
In Tabel 7.10 worden de requirements weergegeven die gaan over de MFA. De eisen REQ12 en REQ13 zijn eisen die iets zeggen over de werking ervan. De andere eisen zijn meer voorwaarden waaraan de MFA moet voldoen. Hier kan rekening mee gehouden worden tijdens het ontwikkelen, maar niet worden verwerkt in het ontwerp.

ID	Omschrijving Requirements
REQ12	Op de Security-Pi kan worden ingelogd met het MFA-apparaat
REQ13	Op de Security-Pi kan een MFA-apparaat worden geregistreerd
REQ14	De MFA heeft een krachtige bescherming tegen phishing-aanvallen
REQ15	De MFA is een fysiek apparaat
REQ16	De MFA kan niet makkelijk worden vergeten
REQ17	De MFA heeft een laag energieverbruik
REQ18	De MFA kan biometrisch identificeren

Tabel 7.10 Requirements MFA

Use-Case diagram MFA

In Figuur 7.13 wordt een Use-Case diagram van het MFA-systeem weergegeven. De werking van het MFA is simpel. Een gebruiker kan het MFA-systeem laten registreren of ermee inloggen.



Figuur 7.13 Use-Case Diagram – MFA – U2F

Scenario's MFA

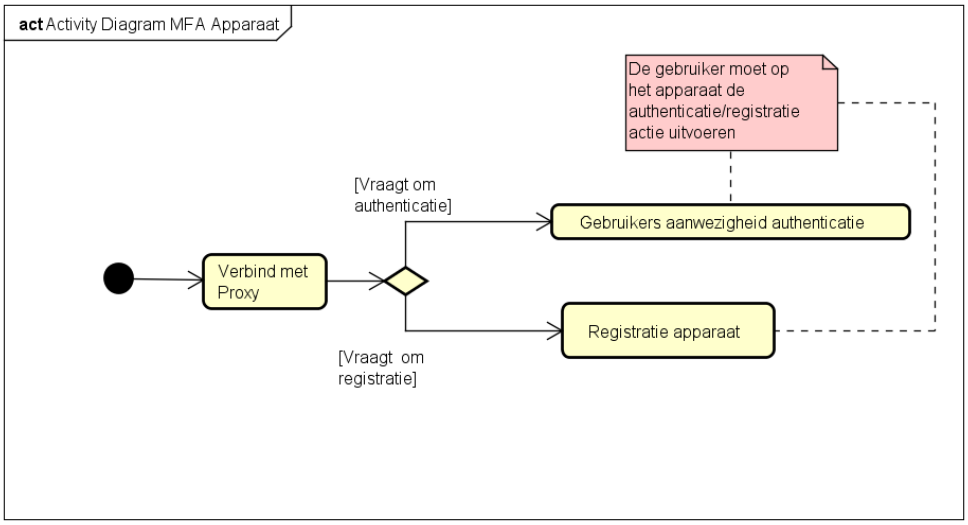
Voor elke case is er een scenario opgesteld dat de werking van elke case weergeeft. Er zijn in totaal twee mogelijke scenario's voor het MFA-systeem. De opgestelde scenario's zijn:

IC3 – SC1	Gebruikers Login Authenticatie
Doel	De gebruiker is ingelogd met MFA-apparaat
Pre-Condities	De gebruiker heeft de correcte gebruikersnaam en wachtwoord ingevoerd
Activiteiten	1. Het systeem vraagt om verificatie op het MFA-apparaat 2. De gebruiker geeft verificatie op MFA-apparaat 3. Het systeem controleert de ontvangen verificatie 4. De gebruiker is geverifieerd met MFA-apparaat
Alternatieven	1. Controle geeft aan dat de ontvangen verificatie van MFA-apparaat niet klopt

IC3 – SC2	Registratie MFA-apparaat
Doel	De gebruiker heeft een MFA-apparaat toegevoegd aan zijn account
Pre-Condities	1. De gebruiker is ingelogd op het Security-Pi Systeem
Activiteiten	1. De gebruiker geeft dat er een MFA-apparaat toegevoegd moet worden aan zijn account 2. Het systeem vraagt om bevestiging van MFA-apparaat dat toegevoegd moet worden 3. De gebruiker geeft registratie vanaf MFA-apparaat 4. Het systeem controleert de ontvangen registratie bericht 5. Het systeem registreert MFA-apparaat aan account
Alternatieven	Controle geeft aan dat MFA-apparaat al geregistreerd is op een ander account.

Activiteitsdiagram

In Figuur 7.14 wordt een activiteitsdiagram van het MFA-apparaat weergegeven. Het MFA-apparaat heeft een BLE-connectie met de Security-Pi. Over deze connectie kan de Security-Pi vragen om een registratie of authenticatie met het MFA-systeem.



Figuur 7.14 Activiteitsdiagram - MFA

Realisatie MFA

In deze activiteit wordt het realisatiegedeelte van het MFA-systeem beschreven. In de vorige activiteit zijn er ontwerpen van het MFA-systeem gemaakt. De ontwerpen worden gebruikt voor het implementeren van het MFA-systeem in het proof of concept. Er is eerder besloten om een BLE/U2F – MFA op te zetten op een smartphone. Als eerst wordt er een beschrijving van een connectie met BLE gegeven. Ten tweede zal worden beschreven hoe het U2F protocol is geïmplementeerd. Als laatste zal er een beschrijving volgen van samenvoegen van het BLE met het U2F-protocol.

Implementeren BLE-connectie

Als eerste is het mogelijk gemaakt om een BLE-connectie met de Security-Pi te maken. De raspberry moet BLE-connecties kunnen opzetten. Om een connectie met de Raspberry te kunnen opzetten moet de raspberry vindbaar gemaakt worden. De Raspberry is hiervoor opgezet als BLE Beacon, zodat deze gevonden kan worden. Als de Beacon zichtbaar is moet het ook connecties toestaan. In Kader 7.3 wordt weergegeven hoe de raspberry is ingesteld als Beacon:

```
sudo hciconfig hci0 up

sudo hciconfig hci0 leadv 0

sudo hcitool -i hci0 cmd 0x08 0x0008 1E 02 01 1A 1A FF 4C
00 02 15 E2 0A 39 F4 73 F5 4B C4 A1 2F 17 D1 AD 07 A9 61
00 00 00 00 C8 00
```

Kader 7.3 Implementatie BLE Beacon

Hieronder volgt een uitleg van de implementatie in Kader 7.3:

- Rood = Hiermee wordt aangegeven dat de Raspberry moet adverteren en connecties toelaat.
- Groen = Dit gedeelte bepaalt het type Beacon en het bericht dat de raspberry adverteert. Het type Beacon is in dit geval een Eddystone Beacon, omdat deze makkelijk in gebruik zijn met Android en Apple smartphones. Een alternatief was IBeacon, maar die type Beacon werkt officieel alleen met Apple smartphones.

Implementeren U2F

In dit gedeelte wordt de implementatie van het U2F-protocol beschreven. Om het U2F-protocol te implementeren over een BLE-connectie is het nodig om de werking van het U2F-protocol te bestuderen. Er is een werkende U2F registratie en authenticatie systeem opgezet op de Security-Pi applicatie. Hiervoor is een U2F-stick "YubiKey NEO" van Yubico gebruikt. Yubico biedt ontwikkelaars library's aan die het mogelijk maken om een U2F MFA te implementeren in applicatie. De library die hiervoor is gebruikt is de Python library "u2fval-client". De library maakt het mogelijk om met de U2F validatie server te communiceren. De U2F validatie server verzorgt de registratie en validatie van U2F-apparaten.

U2F Validatie Client

De U2Fval-Client library is als eerste opgezet, zodat er op de Security-Pi ingelogd kan worden met een MFA. De werking kan getest worden met een server van Yubico. Yubico heeft een U2F validatie server opgezet voor ontwikkelaars, zodat zij deze kunnen gebruiken voor hun eigen applicaties. De server kan worden gebruikt via https://u2fval.appspot.com/. De gebruiker kan kiezen om deze applicatie blijven te gebruiken voor de validaties, maar dan wordt de informatie van gebruikers opgeslagen op de Yubico server. De opdrachtgever wil graag een veilige dienst zonder dat daar nog een derde partij een rol in speelt. In Kader 7.4 wordt een voorbeeld van het registreren van een U2F-sleutel weergegeven.

```
@app.route('/u2f_register', methods=['POST'])
def u2f_register():
  """Register a U2F device"""
  reg_req = u2fval.register_begin(get_current_user())
  return render_template('u2f_add.html',
                name=request.form['name'],
                reg_req=reg_req)
```

Kader 7.4 Registreren U2F-Sleutel

U2F Validatie Server

Op dit moment wordt de validatie van een U2F-sleutel gedaan door de validatie server van Yubico. Het bedrijf Yubico heeft een Python library uitgegeven om zelf de validatie lokaal op te zetten. Deze library heet "python-u2flib-server " en zou geïmplementeerd moeten worden op de Security-Pi, zodat validatie lokaal gedaan kan worden. Voor de implementatie van de validatieserver was er onvoldoende tijd dit is met de opdrachtgever besproken. Er is besloten dit niet verder te implementeren, maar zou meegenomen kunnen worden in een vervolgend project.

Implementeren BLE/U2F – MFA

Uiteindelijk worden het BLE-connectie gedeelte en U2F protocol samengevoegd tot een MFA. De MFA-applicatie is ontwikkeld in Java en er is hiervoor een Samsung Galaxy S6 gebuikt met Android 6. Deze smartphone is gekozen, omdat dit de enige smartphone was die de student beschikbaar had. De smartphone heeft een vingerafdrukscanner die mogelijk gebruikt kan worden als de opdrachtgever alsnog een biometrische beveiliging erbij zou willen hebben, mits er voldoende tijd beschikbaar was. Bij de vorige increment was er al sprake van een tijdstekort. Het is niet gelukt dit gedeelte van het proof of concept te voltooien. Het proof of concept heeft op dit moment wel een werkend U2F MFA waar de opdrachtgever genoeg aan heeft.

Testen MFA

In dit hoofdstuk worden de testen beschreven die uitgevoerd zijn in het derde increment de MFA. In dit increment zijn ook hier de scenario's genomen die zijn opgesteld in de ontwerpactiviteit en aangepast naar testscenario's. Er worden dit keer twee testscenario's genomen, maar er wordt ook weergegeven dat er aanpassingen zijn gedaan aan de testscenario's. Voor het ontwikkelen van het originele idee van een MFA was geen voldoende tijd meer. Er is een aanpassing gedaan aan het proof of concept, zodat er toch een sterk en werkend MFA aanwezig is. Voor een volledige beschrijving wordt er verwezen naar het Testrapport (zie bijlage, "Testrapport").

In Tabel 7.11 en Tabel 7.12 worden de testen weergegeven die het registreren en authentiseren beschrijven. De testen zijn beide niet geslaagd, omdat er geen tijd meer was om het te ontwikkelen. Er is met de opdrachtgever gesproken en er is toen uitgekomen om niet meer het U2F/BLE MFA te ontwikkelen. De opdrachtgever heeft toen aangegeven dat er een ander MFA mag worden ontwikkeld dat wel haalbaar is, maar ook voldoende veiligheid aanbiedt.

IC3 – TC1	Registreer U2F/BLE-apparaat
Doel	Het MFA-apparaat is geregistreerd
Pre-Condities	Geen
Activiteiten	1. Het systeem vraagt om MFA 2. De gebruiker maakt BLE-connectie met systeem via MFA-apparaat 3. De gebruiker drukt op registratie knop via MFA-apparaat 4. Het systeem registreert het MFA-apparaat als MFA
Alternatieven	1. Er wordt niet op registratieknop van MFA-apparaat gedrukt 2. MFA-apparaat is al geregistreerd
Verwachte Resultaat	De gebruiker heeft zijn MFA-apparaat geregistreerd op de Security-Pi via BLE-connectie
Resultaat	BLE-connectie mogelijk, maar nog geen U2F registratie proces
Geslaagd	NEE, Niet voldoende tijd voor ontwikkeling

Tabel 7.11 IC3-TC1 Registreer U2F/BLE-apparaat

IC3 – TC2	Authenticatie met U2F/BLE-apparaat
Doel	Verifieer gebruiker met MFA-apparaat
Pre-Condities	Geen
Activiteiten	1. Het systeem vraagt om verificatie van het U2F/BLE MFA-apparaat 2. De gebruiker maakt BLE-connectie met Security-Pi via U2F/BLE MFA-apparaat 3. De gebruiker drukt op authenticatie knop via MFA-apparaat 4. Het Security-Pi systeem heeft de gebruiker geverifieerd

Alternatieven	1. Er wordt een verkeerde U2F-verificatie verstuurd van U2F/MFA-apparaat 2. Er wordt niet op de verificatieknop van U2F/BLE MFA-apparaat gedrukt
Verwachte Resultaat	De gebruiker is geverifieerd met zijn U2F/BLE MFA-apparaat
Resultaat	BLE-connectie mogelijk, maar U2F authenticatie mogelijk
Geslaagd	NEE, niet voldoende tijd voor ontwikkeling

Tabel 7.12 IC3-TC2 Authenticatie met U2F/BLE-apparaat

In Tabel 7.13 en Tabel 7.14 worden de aangepaste testscenario's weergegeven die het testen met een U2F-stick als MFA beschrijft. De vorige testen waren niet geslaagd, omdat er gewoonweg geen tijd meer was om het te ontwikkelen. De testen zijn aangepast naar het testen met een U2F-stick. Op deze manier is er een sterk en toch nog een werkend MFA voor het proof of concept.

IC3 – TC1.1	Registreer U2F-stick
Doel	De U2F-stick is geregistreerd
Pre-Condities	De U2F-stick is niet geregistreerd
Activiteiten	1. Het Security-Pi systeem vraagt om op U2F knop te drukken 2. De gebruiker drukt op knop op U2F-stick 3. U2F signeert de aanvraag met registratiegegevens 4. Het Security-Pi systeem registreert de U2F-stick als MFA
Alternatieven	1. Er wordt niet op knop van U2F-stick gedrukt 2. U2F-stick is al geregistreerd
Verwachte Resultaat	De gebruiker heeft zijn U2F-Stick geregistreerd op de Security-Pi
Resultaat	De gebruiker heeft zijn U2F-Stick geregistreerd op de Security-Pi
Geslaagd	JA

Tabel 7.13 IC3-TC1.1 Registreer U2F-Stick

IC3 – TC2.1	Authenticatie met U2F-stick
Doel	Verifieer gebruiker met U2F-stick
Pre-Condities	De U2F-stick is geregistreerd
Activiteiten	1. Het Security-Pi systeem vraagt om verificatie van de U2F-stick 2. De gebruiker drukt op knop van U2F-stick 3. U2F stick signeert de aanvraag met authenticatiegegevens 4. Het Security-Pi systeem het de gebruiker geverifieerd
Alternatieven	1. Er wordt een verkeerde U2F-verificatie verstuurd 2. Er wordt niet op de knop van U2F-stick gedrukt
Verwachte Resultaat	De gebruiker is geverifieerd met een U2F-stick
Resultaat	De gebruiker is geverifieerd met een U2F-stick
Geslaagd	JA

Tabel 7.14 IC3-TC2.1 Authenticatie met U2F-Stick

8 Conclusie

Aan de start van het project is er een aanpak gekozen die de student door het project heeft geleid. Deze aanpak is een aangepaste vorm van de waterval-methode die gebruik maakt incrementen om het proof of concept te ontwikkelen.

De eerste fase van het project is de oriëntatiefase. Aan het begin van de opdracht wordt aangegeven dat ICT Group als doel graag een zwaar beveiligde verbinding wilt hebben. Er zijn in de oriëntatiefase interviews geweest om de opdracht te verhelderen. De opdracht is toen aangepast naar het beveiligen van data op een externe dienst. Hiermee wilt de opdrachtgever aantonen aan zijn klanten dat het mogelijk is eigen data versleuteld op te slaan bij een externe publieke opslagdienst. Het resultaat van deze werkzaamheden heeft geleid tot een duidelijk beeld van de opdracht.

In de tweede fase is er literatuuronderzoek uitgevoerd om meer kennis over het onderwerp op te doen. Na het onderzoek zijn de eisen van de opdrachtgever zijn vastgesteld met behulp van gehouden interviews en meetings. Daarna zijn er mogelijke systeemopstellingen opgesteld, waaruit een afweging is gemaakt en als resultaat opstelling 2: certificaten PKI, AES 256-bit in CBC-modus en als MFA het U2F-protocol in combinatie met BLE gebruikt wordt. Toen er een opstelling was geselecteerd zijn er incrementen opgesteld die worden ontwikkeld in de volgende fase.

De laatste fase is de ontwikkelfase. In deze fase zijn de opgestelde incrementen ontwikkeld tot een werkend proof of concept. Aan het einde van elke increment zijn er validatietesten uitgevoerd. De testen zijn uiteindelijk allemaal geslaagd en daarmee wordt aangetoond dat de doelstelling mogelijk te behalen is met het proof of concept.

Het proof of concept kan worden aangeboden aan klanten van ICT Group die een behoefte hebben aan extra beveiliging van hun data dat bij ICT Group of een andere organisatie wordt opgeslagen. In bepaalde gevallen kan deze extra beveiliging voor klanten een bepalingspunt zijn of zijn met ICT Group een contract willen aangaan. Dit proof of concept is een manier om het probleem van de opdrachtgever te verhelpen. Er zullen vast andere manieren bestaan om het doel in het project te realiseren. In dit geval heeft het onderzoek de student naar dit resultaat geleid en het resultaat is werkend. Het resultaat is gepresenteerd aan de opdrachtgever en is er tevreden mee.

9 Evaluatie Afstudeeropdracht

In dit hoofdstuk wordt een evaluatie van de visie die de student heeft op de afstudeeropdracht.

Het was aan de start niet duidelijk wat de opdrachtgever wilde hebben er is toen bij het afstudeerplan genoteerd wat op dat moment de gedachte was bij de wens van de opdrachtgever. Om dit duidelijk te krijgen was het duidelijk dat dit gedeelte achterhaald moest worden en er een plan opgesteld om dit te achterhalen. Tijdens het opstellen van een plan van aanpak is er veel contact geweest met de begeleider en andere collega's van ICT Group. Zij hadden allemaal hun eigen ideeën over het selecteren van een aanpak. Na veel overleg leek het RAD-algoritme voor ICT Group meest bruikbaar, maar na zelf onderzoek naar ontwikkelalgoritmes is er uiteindelijk gekozen voor een watervalmethode. Deze keuze blijkt een juiste keuze zijn geweest, omdat het project goed is verlopen.

In de oriëntatiefase is er een planning opgesteld met wat op dat moment de verwachte tijdsduur zou zijn van een bepaalde activiteit. Dat is uiteindelijk toch anders dan verwacht. Er is meer tijd in de definitiefase vereist geweest en waardoor er minder tijd overbleef in de ontwikkelfase.

Tijdens de definitiefase is er veel onderzoek geweest naar mogelijke encryptiemethoden en MFA dat bruikbaar zou zijn in het proof of concept. Daarbij is er veel input geweest van verschillende medewerkers van ICT Group wat een mogelijkheid zou kunnen bieden bij het proof of concept. Als deze medewerking van meerdere medewerkers er niet was geweest dan zou het onderzoek ook niet zo uitgebreid geweest. Het heeft mij geholpen om op een goede richting te blijven. Dit is bij een afstudeertraject lastig, omdat je vooral op jezelf bent aangewezen om alles te doen.

In de ontwikkelfase is er veel gekeken naar mogelijkheden die gebruikt kunnen worden om het product te realiseren. Het komt bij sommige incrementen voor dat er pas later tijdens de ontwikkeling van het product een mogelijk betere keuze naar voren kwam. Aangezien het project een deadline heeft is er alleen in overleg met de opdrachtgever een keuze gemaakt om een overstap te maken naar de juiste keuze.

Over het algemeen is mijn ervaring bij ICT Group zeer bevallen. Ik heb al eens eerder het afstudeertraject doorgelopen en dat was niks vergeleken met de begeleiding die ik hier vanuit ICT Group heb meegekregen. De medewerkers en begeleider waren oprecht geïnteresseerd in het helpen van mij met mijn opdracht. Iets wat ik de vorige keer miste bij het afstuderen.

Bronnenlijst

André Clerc - Temet. (2017, February 9). *About & Beyond PKI - Blockchain and PKI.* Opgehaald van http://www.sig-switzerland.ch: http://www.sig-switzerland.ch/wp-content/uploads/2015/06/SIGS_Feb2017_TEMET_Does_Blockchain_secure_PKIs_in_the_long-term.pdf

Atmel. (2015). *RSA vs ECC Comparison for Embedded Systems.*

Bluetooth SIG. (2016). *Bluetooth Core Specification 5.0 FAQ.* SIG.

Bluetooth SIG. (2017). *Bluetooth Core Specification v5.0.* Bleutooth SIG.

Bluetooth SIG. (sd). *Bluetooth low energy security.*

Bluetooth SIG Security Expert Group. (2002, April 19). *Bluetooth Securtiy White Paper.* Retrieved from http://grouper.ieee.org: http://grouper.ieee.org/groups/1451/5/Comparison%20of%20PHY/Bluetooth_24Security_Paper.pdf

Cobb, C. (2004). *Cryptography For Dummies.* John Wiley & Sons.

Dams, J. (2012, october 12). *An introduction to elliptic curve cryptography*. Opgehaald van http://www.embedded.com: http://www.embedded.com/design/safety-and-security/4396040/An-Introduction-to-Elliptic-Curve-Cryptography

Developers, S. B. (Regisseur). (2015). *SF Bitcoin Devs Seminar: RevokeSSL: An Independent Revocation Service using the Bitcoin Blockchain* [Film].

DLP, D. (2016, june 16). *Bluetooth 5: 2x the speed + 4x the range = 8x the security risk*. Opgehaald van http://www.devicelock.com/: http://www.devicelock.com/blog/2842.html

Earle, J. (2015, May 16). *Elliptic Curve Cryptography & Diffie-Hellman.* Opgehaald van http://www.csbreakdown.com/: https://www.youtube.com/embed/yDXiDOJgxmg?rel=0

Facebook. (2013, July 31). *Secure browsing by default*. Opgehaald van https://www.facebook.com: https://www.facebook.com/notes/facebook-engineering/secure-browsing-by-default/10151590414803920/

Knafo, J. (2016, October 21). *Most Popular 2-Factor Authentication (2FA) Compared*. Opgehaald van https://blog.devolutions.net: https://blog.devolutions.net/2016/10/most-popular-2-factor-authentication-2fa-compared.html

London, S. B.-S. (Regisseur). (2015). *Martijn Grooten - Elliptic Curve Cryptography for those who are afraid of maths* [Film].

N, R. (2017). *U 2 can U2F.* Opgehaald van https://robn.io/: https://robn.io/talks/u2f-lca-2017/U2F-notes.pdf

National Institute of Standards and Technology. (2012, June). *Guide to Bluetooth Security.* Opgehaald van http://nvlpubs.nist.gov: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-121r1.pdf

National Science and Technology Council (NSTC). (2013). *Fingerprint Recognition.* FBI.

Open Universiteit Nederland. (sd). *OU.* Opgehaald van Portal.ou.nl: http://portal.ou.nl/documents/informatica/snapshots/T07341_01.pdf

Robles, P. (2015, March 17). *Can the blockchain replace SSL?* Opgehaald van https://www.programmableweb.com/: https://www.programmableweb.com/news/can-blockchain-replace-ssl/analysis/2015/03/17

Schneider, P. F. (2005). *Something You Know, Have, or Are*. Opgehaald van https://www.cs.cornell.edu: https://www.cs.cornell.edu/courses/cs513/2005fa/NNLauthPeople.html

Shaver, j. (2015, February 11). *Decrypting TLS Browser Traffic With Wireshark – The Easy Way!* Opgehaald van https://jimshaver.net/: https://jimshaver.net/2015/02/11/decrypting-tls-browser-traffic-with-wireshark-the-easy-way/

Tel, G. (2006). *Cryptografie - Beveiliging van de digitale maatschappij.* Universiteit Utrecht.

Tripura University, I. (2011, July 27). *https://www.intechopen.com.* Opgehaald van Thermal Infrared Face Recognition – a Biometric Identification Technique for Robust Security System: https://www.intechopen.com/books/statistics/reviews-refinements-and-new-ideas-in-face-recognition/thermal-infrared-face-recognition-a-biometric-identification-technique-for-robust-security-system

inhoud

1 Afstudeerplan

Informatie afstudeerder en gastbedrijf (*structuur niet wijzigen*)

Afstudeerblok: 2017-1.1 (start uiterlijk 6 februari 2017)
Startdatum uitvoering afstudeeropdracht: 1februari 2017
Inleverdatum afstudeerdossier volgens jaarrooster: 2 juni 2017

Studentnummer: 10065954
Achternaam: dhr. Jagai (*) *weghalen niet van toepassing*
Voorletters: RR
Roepnaam: Ramiro
Adres: Dr.J.Presserstraat 25
Postcode: 2552 LR
Woonplaats: Den Haag
Telefoonnummer:
Mobiel nummer: 06-24 83 28 76
Privé emailadres: ramirojag@live.nl

Opleiding: TI
Locatie:Delft (*) *weghalen niet van toepassing*
Variant: voltijd

Naam studieloopbaanbegeleider: dhr. J. van Peski
Naam begeleidend examinator: dhr. J. P. M. de Vreught
Naam tweede examinator: dhr. M. G. Maris

Naam bedrijf: ICT Group NV
Afdeling bedrijf:
Bezoekadres bedrijf: Kopenhagen 9
Postcode bezoekadres: 2993 LL
Postbusnummer: nvt
Postcode postbusnummer: nvt
Plaats: Barendrecht
Telefoon bedrijf: 088 908 2000
Telefax bedrijf: 088 908 2500
Internetsite bedrijf: www.ict.eu

Achternaam opdrachtgever: dhr. Lamot (*) *weghalen niet van toepassing*
Voorletters opdrachtgever: B.
Titulatuur opdrachtgever:
Functie opdrachtgever: innovation manager
Doorkiesnummer opdrachtgever:
Email opdrachtgever: Bart.Lamot@ict.nl

Achternaam bedrijfsmentor: dhr. van 't Hof (*) *weghalen niet van toepassing*
Voorletters bedrijfsmentor: A
Titulatuur bedrijfsmentor:
Functie bedrijfsmentor: Project manager
Doorkiesnummer bedrijfsmentor:
Email bedrijfsmentor: Andre.van.t.Hof@ict.nl

NB: bedrijfsmentor mag dezelfde zijn als de opdrachtgever

Doorkiesnummer afstudeerder:
Functie afstudeerder (deeltijd/duaal):

Titel afstudeeropdracht:

Het ontwikkelen van extra beveiliging voor communicatie van klanten naar ICT Group

Opdrachtomschrijving

1. Bedrijf

ICT Group heeft in haar 40 jaar bestaan een groot en divers portfolio van klanten opgebouwd. De klanten bevinden zich in een groot aantal uiteenlopende markten zoals bijvoorbeeld energie, water & infrastructuur, healthcare en logistiek. De overkoepelende factor is technologie die versterkt wordt door software.

Door een achtergrond in embedded software ontwikkeling is Internet of Things een dagelijkse praktijk voor ICT Group geworden. Hierin zien zij een ontwikkeling dat bestaande "domme" voorwerpen steeds "slimmer" worden gemaakt.

Een aantal klanten van ICT Group zijn logistieke bedrijven in het havengebied van Rotterdam. Deze bedrijven bewaren de logistieke container informatie bij ICT Group. ICT Group gebruikt nu een direct onveilige verbinding met de hun eigen cloud. ICT Group wil dat software die bij klanten staat zo min mogelijk aangepast moet worden.

2. Probleemstelling

Het probleem is dat deze klanten momenteel hun logistieke data niet veilig opslaan bij ICT Group.

3. Doelstelling van de afstudeeropdracht

Het doel is dat klanten met behulp van een extra beveiligde verbinding logistieke data in de Cloud van ICT Group kunnen opslaan. Op deze wijze is de data van de klant beter beveiligd.

4. Resultaat

Een zwaar beveiligde verbinding tussen de klant en ICT Group

5. Uit te voeren werkzaamheden, inclusief een globale fasering, mijlpalen en bijbehorende activiteiten

- Oriëntatiefase (1 week)
  - Verhelderen opdracht
  - Risico's vaststellen
  - Project plannen

- Definitiefase (5 weken)
  - Analyse over onderwerp
    - literatuuronderzoek
    - Gebruik van USB, NFC, Bluetooth (als fysieke sleutel)
    - Uitzoeken encryptiemethoden(PKI, AES, 3DES)
    - Mogelijkheden in VPN
    - Systeemconcepten voor een mogelijke architectuur bepalen
  - Achterhalen requirements van informatiesysteem
    - Definitief vaststellen van requirements
  - Bepalen definitieve architectuur
  - Bepalen van de incrementen voor het vervolg van het project

- Ontwikkelfase met later te bepalen incrementen (9 weken totaal)
  - Een ontwerp maken
  - Realiseren
  - Testen

6. Op te leveren (tussen)producten

- Plan van aanpak
- Definitie-rapport
- Ontwikkelrapport per increment
- POC, proof of concept
- Testrapport

7. Te demonstreren competenties en wijze waarop

- G1 Praktische aspecten hanteren in (internationale) projecten
  Dit wordt aangetoond aan de hand van de risicoanalyse in het plan van aanpak, bijwonen van vergaderingen. ICTGroup is een internationaal georiënteerd bedrijf waar ik mij moet zien te handhaven, twee periodes lang.

- A1 - Analyseren van het probleemdomein
  Dit wordt aangetoond door het doen van onderzoek en analyses op de huidige situatie bij de klanten van ICT Group. Door het houden van interviews met medewerkers van ICT Group kan er worden achterhaald wat de huidige situatie is bij klanten.

- A3 – Achterhalen van behoeften van belanghebbenden
  Dit kan worden aangetoond door het maken van interviewverslagen met medewerkers van ICT Group.

- C9 - Ontwerpen van een technisch infrastructuur
  Dit wordt aangetoond in het ontwerp van het ontwikkelrapport

- D18 – Testen van een infrastructuur
  Dit wordt aangetoond aan de hand van de testrapportage

ICT Group N.V.

HET ONTWIKKELEN VAN EEN EXTRA BEVEILIGING VOOR NETWERKCOMMUNICATIE VAN KLANTEN NAAR DE CLOUD VAN ICT GROUP

PLAN VAN AANPAK

Technische Informatica
Versie 1.0
13 februari 2017

Inhoudsopgave

1 Introductie

ICT Group (kortweg ICT) bestaat al sinds 1978 en is opgericht onder de naam ICT-automatisering. De naam van ICT is in de loop van de jaren vaak veranderd en sinds 2016 is de officiële naam van het bedrijf ICT Group N.V. Het hoofdkantoor van ICT is gevestigd in Barendrecht en het bedrijf heeft meerdere vestigingen door heel het land. Plaatsen van de vestigingen van ICT zijn Bergen op Zoom, Eindhoven, Deventer, Gorinchem, Groningen, Maastricht, Oosterhout. ICT kan worden verdeeld in verschillende units de gerelateerd zijn aan de branches waarin zij werkzaam zijn. De units zijn Automotive & Mobility, Water & Infrastructure, Healthcare, Industry, Transport & Logistics, High Tech, Energy en Manufacturing.

Dit project wordt uitgevoerd op het hoofdkantoor in Barendrecht in de unit Transport & Logistics. Klanten van Transport & Logistics zijn deels logistieke bedrijven in het havengebied van Rotterdam. Zij zijn voor deze opdracht de klant. Binnen deze havenbedrijven leeft de vrees dat er weleens logistieke data wordt aangepast door hackers; deze passen dan bijvoorbeeld de inhoud aan van een container die op een schip aankomt. Op deze manier wordt het nog moeilijker voor de bedrijven om containers te achterhalen die illegale handelswaar bevatten. De klanten van ICT hebben ICT gevraagd om met een mogelijke oplossing te komen. ICT heeft een onderzoek gestart naar het gebruik van three-factor authentication (3FA) bij de systemen van klanten. De eerste factor staat voor “something you know”, de tweede factor staat voor “something you have” en de derde factor staat voor “something you are”. Aan de student de taak om voor ICT te onderzoeken wat de mogelijkheden hierin zijn.

Als eerste wordt er in dit plan van aanpak een opdrachtomschrijving gegeven die een aanleiding, probleemstelling, doelstelling en de verwachte resultaten weergeeft. Daarna worden de projectgrenzen besproken. Als derde worden de op te leveren producten besproken. Vervolgens wordt beschreven hoe het project wordt aangepakt door het selecteren van een ontwikkelmethode en het maken van een fasebeschrijving met daarbij een opsomming van de activiteiten. Het vijfde hoofdstuk bevat informatie over de organisatie en werkplek binnen ICT. Daarna volgt er nog een planning van het project en daarbij behorende mijlpalen. Als laatste worden de risico’s beschreven die invloed kunnen hebben op het project en daarbij de maatregelen ter beperking van die risico’s.

2 Afstudeeropdracht

In dit hoofdstuk wordt de opdracht omschreven die zal worden uitgevoerd bij ICT. De omschrijving van de afstudeeropdracht kan worden verdeeld in een aanleiding, probleemstelling, doelstelling en resultaten.

2.1 Aanleiding

Een aantal klanten van ICT zijn logistieke bedrijven in het havengebied van Rotterdam. Deze bedrijven bewaren logistieke containerinformatie bij ICT. ICT is verantwoordelijk voor de veiligheid van deze logistieke data. Op het moment blijkt dat de verbinding die nu in gebruik is zou kunnen worden misbruikt. De klanten van ICT hebben gevraagd naar een betere beveiliging in deze verbinding.

2.2 Probleemstelling

ICT gebruikt nu bij deze klanten een directe verbinding naar hun eigen Azure cloud. Deze verbinding is nog kwetsbaar voor aanvallen van buitenaf waarbij de logistieke data van de klant zou kunnen worden aangepast.

2.3 Doelstelling

ICT is op zoek naar een oplossing voor haar klanten die zo generiek en vervangbaar is dat de software die bij klanten staat niet aangepast hoeft te worden. Medewerkers die gebruik willen maken van de verbinding naar de Cloud hebben naast een wachtwoord ook een fysieke sleutel “something you have” en een biometrische sleutel “something you are” nodig om de verbinding tot stand te brengen.

2.4 Resultaten

Het resultaat is een rapport en een proof of concept met behulp van een Raspberry Pi die ICT een inzicht geven in het gebruik van een extra beveiliging in de verbinding tussen haar klanten en de Azure Cloud.

3 Projectgrenzen & Randvoorwaarden

Dit hoofdstuk worden de grenzen en voorwaarden bepaald die horen bij deze opdracht. Door meteen aan het begin van een project de grenzen vast te leggen wordt er een afbakening gedaan van het resultaat. Zo ontstaat er geen onduidelijkheid aan het eind van het project over wat de student heeft uitgevoerd.

De volgende punten vallen binnen de projectgrenzen:

- Het project wordt ontwikkeld op een Raspberry Pi met als OS Linux.
- Er wordt een externe fysieke sleutel gebruikt die de verbinding met de Cloud tot stand brengt. Er wordt onderzoek gedaan naar de beste, niet – kopieerbare oplossing.
- De data moet ook worden beveiligd, daarvoor zal een encryptiemethode worden gekozen.
- Er mogen geen aanpassingen worden gedaan in bestaande software van klanten.

Tijdens het opstellen van het document “plan van aanpak” is samen met de opdrachtgever en begeleider besloten dat de volgende punt wordt toegevoegd aan de opdracht.

- Toevoegen van nog een factor aan de beveiliging: “biometrische beveiliging”. De opties kunnen zijn vingerafdrukscan, irisscan, gezichtsherkenning en spraakherkenning. Later in het project wordt bepaald welke biometrische beveiliging wordt gebruikt.

Naast een afbakening van de projectgrenzen zijn er ook een aantal randvoorwaarden om het onderzoek uit te kunnen voeren.

- Er wordt gezorgd voor een Raspberry PI waarop de student kan werken.
- De student heeft een werkplek voor zijn testomgeving en documentatie.
- Er wordt gezorgd voor een mogelijke externe fysieke sleutel.
- Als een biometrische beveiliging wordt toegevoegd aan de opdracht moet er een apparaat beschikbaar zijn die de biometrische eigenschappen van een persoon kan vaststellen.

4 Producten

In dit hoofdstuk wordt een inzicht gegeven in de producten die worden opgeleverd gedurende en aan het eind van deze afstudeeropdracht.

Plan van aanpak
In het document "*plan van aanpak*" wordt de opdracht verder uitgewerkt. Er wordt een onderzoek naar verschillende methodes gedaan die mogelijk toegepast kunnen worden in dit project. Nadat er een keuze is gemaakt in het gebruik van een methode wordt er een fasebeschrijving gemaakt die de activiteiten per fase weergeeft. Daarna kan er een globale planning worden gemaakt van de tijdsduur van elke fase.

Definitierapport
Dit document beschrijft een analyse van het probleemdomein die verricht is. Er wordt vastgesteld wat de eisen en wensen van het informatiesysteem zijn en er wordt bepaald hoe de systeemarchitectuur eruit komt te zien.

Ontwikkelrapport
Per increment wordt er een ontwikkelrapport opgeleverd. Het ontwikkelrapport beschrijft meerdere ontwerpen die gemaakt zijn en het implementeren van het ontwerp.

Testrapport
Het testrapport bevat verschillende testen die zijn gedaan ter controle van de implementatie. De testen worden uitgevoerd door de begeleider en opdrachtgever.

Proof of concept (POC)
De POC is een demonstratie van het onderzoek dat is verricht. Hiermee wordt aangetoond dat de gekozen oplossing is ook daadwerkelijk bruikbaar is.

Afstudeerverslag
Het afstudeerverslag beschrijft het gehele proces en belangrijke keuzes die gemaakt zijn tijdens het project. Dit is tevens ook het verslag dat moet worden ingeleverd op de Haagse Hogeschool.

5 Aanpak

In dit hoofdstuk wordt er gekeken naar de mogelijke methodes die gebruikt kunnen worden in dit project. Er wordt als eerst een ontwikkelstrategie gekozen. Als het duidelijk is welke strategie er zal worden gebruikt kan er een ontwikkelmethode worden geselecteerd.

5.1 Ontwikkelstrategie

Om een juiste keuze te maken bij het selecteren van diverse ontwikkelmethoden is er als eerste een aantal bekende ontwikkelstrategieën geselecteerd. Deze zijn vervolgens vergeleken met de kenmerken van het project. Er dient gekeken te worden in hoeverre de kenmerken van het project het best aansluiten binnen een ontwikkelstrategie. Hieronder volgen de categorieën waar het project op aan kan sluiten.

- **Waterval** – Bij het toepassen van de watervalmethode wordt er in fasen gewerkt en is het alleen mogelijk naar de volgende fase te gaan als alles af is van de voorgaande fase. Als de ontwikkelaar begonnen is met een bepaalde fase mag hij niet meer teruggaan naar een vorige fase.

- **Iteratief** – Bij alleen iteratief software ontwikkelen wordt het ontwikkelproces steeds herhaald en elke herhaling levert een voorlopige versie van de software op, waarna er om feedback wordt gevraagd om de software aan te passen. Dit proces wordt steeds herhaald totdat de klant tevreden is of er geen tijd meer over is of het project buiten het budget valt.

- **Incrementeel** – Bij incrementeel software ontwikkelen wordt de software in delen opgebouwd. Elke iteratie levert een increment op. Een increment is het doorlopen van de opgestelde disciplines dat een deel werkende code van de complete software oplevert. Dit proces herhaalt zich steeds totdat er een compleet werkend systeem uitkomt.

Hieronder volgen de voor- en nadelen bij het toepassen van een van de bovenstaande ontwikkelstrategieën.

Watervalstrategie

De watervalmethode is een lineaire ontwikkelmethode die ontwikkelaars helpt inzicht en duidelijkheid te verschaffen bij grote softwareprojecten. De watervalmethode verdeelt het project in fasen en gaat als het ware één voor één de fasen af als een soort waterval.
Het voordeel van de watervalmethode is dat het een rechttoe-rechtaan ontwikkelmethode is. De fasen en mijlpalen van het project staan vast met deze strategie en is het gewoonweg op volgorde de fasen afgaan.
Een nadeel van de watervalmethode is dat er niet gemakkelijk terug kan worden gegaan naar een fase, zonder dat alle andere fasen eerst opnieuw worden doorlopen. Dit kan tegenwerken als de wensen van de opdrachtgever tijdens het project veranderen.

Iteratieve strategie

Alleen iteratief ontwikkelen is een herhalend proces dat elke iteratie een voorlopige versie van het volledige systeem oplevert. Er wordt na elke iteratie feedback gegeven op het opgeleverde systeem en bij de volgende iteratie worden de aanpassingen meegenomen in het ontwikkelproces.
Het voordeel van iteratief ontwikkelen is dat na de eerste iteratie al een volledig systeem wordt opgeleverd. Als de wensen van de opdrachtgever weinig veranderen. Hebben de opvolgende iteraties minder tijd nodig om het systeem naar wens van de opdrachtgever te maken.
Een nadeel is dat één iteratie veel tijd kost om een volledig systeem op te leveren. Als de wensen van de opdrachtgever veel veranderen kan het systeem dat is opgeleverd na de eerste iteratie niet meer worden gebruikt voor de opvolgende iteraties. De opvolgende iteraties kunnen dan veel tijd in beslag nemen, maar stukken van het systeem dat is opgeleverd in de eerste iteratie kunnen worden meegenomen als deze nog wel overeenkomen met de wensen van de opdrachtgever.

Incrementele strategie

Bij incrementeel ontwikkelen wordt er een herhaling van fasen doorlopen. Na elke iteratie wordt er één increment opgeleverd dat een deel van het volledige systeem bevat en die wordt na elke iteratie aangevuld

totdat er een volledig systeem ontstaat. Het voordeel van een incrementele strategie is dat er steeds alleen een deel van het volledige systeem wordt opgeleverd na een iteratie. Na een iteratie wordt er opnieuw vastgesteld met de opdrachtgever of de eisen van het systeem nog overeen komen met wat er ontwikkeld gaat worden. Dit maakt de tijdsduur van een iteratie korter en geeft minder kans dat er een systeem ontwikkeld wordt dat niet voldoet aan de eisen.

Besluit ontwikkelstrategie
De wensen en behoeften van de gebruikers kunnen tijdens het verloop van het project wijzigen. Het is belangrijk voor dit project om vooraf rekening te houden met eisen die kunnen veranderen. Om deze reden valt de watervalmethode af.
Er is besloten incrementeel te ontwikkelen, zodat er rekening gehouden kan worden met veranderende eisen en wensen van de opdrachtgever. Bij iteratief ontwikkelen kan veel tijd verloren gaan om een volledig systeem op te leveren in iedere iteratie. Door gebruik van een incrementele strategie en elke iteratie opnieuw de eisen van het systeem vast te stellen met de opdrachtgever kan er na verloop van tijd een volledig systeem ontstaan.

5.2 Ontwikkelmethode

Om uit verschillende incrementele methodes een keuze te maken zijn er een aantal voorbeelden genomen van methodes die incrementeel gebruikt kunnen worden. Deze methodes worden dan vergeleken met wat de opdrachtgever precies wil hebben. De opdrachtgever wil in dit geval een volledig onderzoek, ze hebben niks aan een voorlopige versie of een project dat nog niet af is. Dit kunnen ze niet presenteren aan klanten van ICT die de opstelling zouden kunnen gebruiken binnen hun bedrijf.

Om een afweging te maken in verschillende methoden worden een aantal belangrijke kenmerken van de opdracht genomen waaraan de ontwikkelmethode moet voldoen.

- **Projectgrootte** – De grootte van het project is middelmatig en wordt uitgevoerd door één student. De methode moet beheersbaar en simpel blijven zodat één student de methode juist toe kan passen.
- **Mate van ceremonie** – Dit kenmerk staat voor de mate waarin het proces wordt gecontroleerd, formeel is en resultaten worden gemodelleerd en gedocumenteerd. Voor ICT is het belangrijk dat er een product wordt opgeleverd dat werkt. Voor de Haagse Hogeschool is het belangrijk dat het proces wordt beschreven door middel van documentatie en modellen.
- **Gestructureerdheid van het probleem** – Het probleemdomein in dit project is gestructureerd. Het is duidelijk wat het probleem is en de oplossing kan met behulp van het uitvoeren van onderzoek, programmeren en een proof of concept worden aangetoond.

Binnen de incrementele strategie zijn de volgende methodes geselecteerd die populair zijn in gebruik bij projecten: RUP, SCRUM en RAD. Hieronder volgt een beschrijving van de voor- en nadelen bij het gebruik van de methodes in dit project.

Rational Unified Process (RUP)
De ontwikkelmethode RUP is een grote en complexe ontwikkelmethode. RUP bestaat uit verschillende fases die weer bestaan uit diverse disciplines. Alle disciplines leveren artefacten op die weer gedocumenteerd moeten worden. Bij RUP wordt er gevraagd om veel artefacten op te leveren wat voor de opdrachtgever in dit project niet nodig is en ook veel kostbare tijd vereist. De methode RUP wordt het meest toegepast bij grote en ingewikkelde projecten. De complexiteit en grootte van dit project ligt op een gemiddeld niveau.

SCRUM
SCRUM is een veel gebruikte methode binnen ICT. De kracht van de methode ligt in het gebruik van teams en verschillende rollen te verdelen binnen het team. Volgens SCRUM worden er ook dagelijks met het team SCRUM–meetings gehouden om de voortgang van het project te bespreken. Ondanks dat SCRUM iteratief en incrementeel kan worden gebruikt, wordt er binnen SCRUM een maximale tijdsduur van 30 dagen aangegeven

per iteratie. Bij deze afstudeeropdracht is het nog niet duidelijk wat precies de tijdsduur is van een iteratie. Het houden van dagelijkse meetings met de opdrachtgever heeft geen toegevoegde waarde voor dit project en de verschillende rollen kunnen niet worden verdeeld als er maar één ontwikkelaar aan het project werkt. Op veel kenmerken van dit project voldoet SCRUM niet. SCRUM verliest zijn kracht als de methode wordt aangepast naar de kenmerken van het project.

Rapid Application Development (RAD)
RAD is een ontwikkelproces dat in opeenvolgende versies gebouwd en getest wordt. Er wordt een pilot gemaakt en daarop getest en schaduw gedraaid. Als er iets mis gaat, dan valt men op het oude prototype terug. Het gebruik van RAD wordt aangeraden bij kleine tot middelgrote projecten, waar snel een werkend product uit moet komen. Een specifiek nadeel van RAD is dat de systeemeisen in één keer moeten worden vastgesteld. In dit project is dit niet meteen een groot nadeel, omdat de opdrachtgever duidelijk weet wat het probleem is. Bij RAD ligt de mate van ceremonie laag. Het is in feite de eisen van het project vaststellen en dan incrementeel ontwikkelen. Dit zorgt voor een snelle oplevering van incrementen en zo kan ook vroeg de haalbaarheid van het project worden aangegeven. Een ander voordeel van RAD bij dit project is dat het een heel flexibele methode is. Het kan gebruikt worden bij software- en hardware-ontwikkelprojecten.

Besluit ontwikkelmethode
In Tabel 5.1 wordt weergegeven hoe sterk een methode aan een bepaald kenmerk voldoet. De afweging wordt gedaan met de letters A t/m E. De letter A staat voor een hoog en de letter E voor een laag compatibiliteit.

Kenmerken	RUP	SCRUM	RAD
Projectgrootte	D	C	A
Lage mate van ceremonie	D	C	A
Hoge gestructureerdheid van het probleem	B	B	A

Tabel 5.1 Afweging ontwikkelmethodes - A = Hoog, E = Laag

Het gebruik van RAD komt in de bovenstaande tabel uit als de meest optimale methode voor dit project. Aan het eind van deze afstudeeropdracht wordt er een proof of concept opgeleverd als oplossing. RAD biedt de ontwikkelaar vroeg een inzicht in de haalbaarheid van deze proof of concept. RUP bestaat uit meerdere fases en vaak komt het voor dat in de beginfases niet eens aan een proof of concept gewerkt wordt, maar er wordt juist meer focus gelegd op de documentatie. Dit kan ertoe leiden dat het laat in het project duidelijk is of het project haalbaar is. Als blijkt dat het project niet haalbaar is wordt het veel lastiger de opdracht aan te passen naar een wel haalbaar project. Het besluit is genomen om de methode RAD te gebruiken.

5.3 Fasebeschrijving

In de vorige paragraaf is er een besluit genomen om de methode van RAD te gebruiken. De methode wordt in deze paragraaf uitgewerkt tot een fasebeschrijving. Naast de fases worden de activiteiten en conclusies die uit een fase komen uitgelegd.

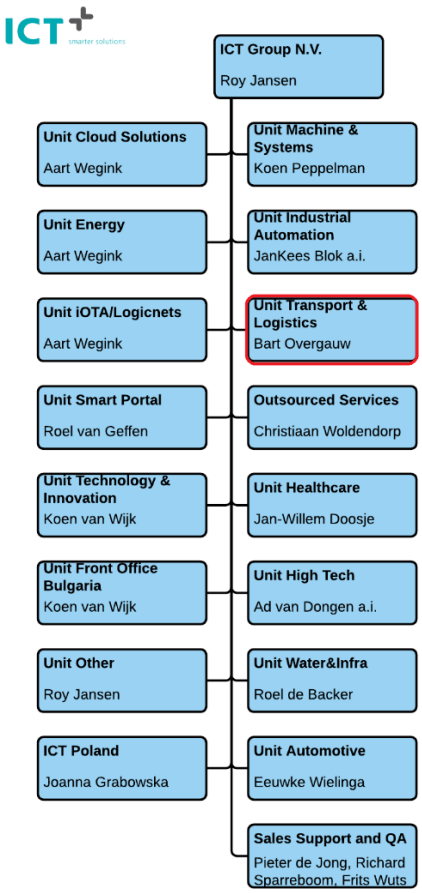
Fasen	Activiteiten	Conclusies & doel
Oriëntatiefase	▪ Verhelderen van de opdracht ▪ Risico's vaststellen ▪ Planning maken van project	Pas als de opdracht duidelijk is kan er goed onderzoek worden gedaan naar het onderwerp. Activiteiten van deze fase worden beschreven in het PVA.
Definitiefase	▪ Analyse probleemdomein ○ Literatuuronderzoek	▪ Bepalen van de incrementen en de

	○ Gebruik van USB, NFC, Bluetooth (als fysieke sleutel) ○ Uitzoeken encryptiemethoden (PKI, AES, 3DES) ○ Mogelijkheden in VPN ○ Systeemconcepten voor een mogelijke architectuur bepalen ▪ Achterhalen requirements van informatiesysteem ○ Definitief vaststellen van requirements ▪ Bepalen definitieve architectuur ▪ Bepalen definitieve biometrische beveiliging	volgorde in het uitvoeren van de incrementen. ▪ Activiteiten van deze fase worden beschreven in het definitierapport.
Ontwikkelfase	Increment 1: Encryptiemethode ▪ Specifieke analyse over het onderwerp ▪ Ontwerpen topologie van omgeving ▪ Realiseren van topologie met encryptie-omgeving ▪ Testen van realisatie Increment 2: gebruik van “fysieke” sleutel – “Something you have” ▪ Specifieke analyse over het onderwerp ▪ Ontwerpen werking van fysieke sleutel ▪ Realiseren van ontwerp ▪ Testen van realisatie Increment 3: Biometrische beveiliging – “Something you are” ▪ Specifieke analyse naar keuze biometrische beveiliging ▪ Ontwerpen van het systeem ▪ Realiseren van het ontwerp ▪ Testen van realisatie	▪ De volgorde van de incrementen kan altijd worden aangepast. ▪ Per increment wordt een gedeelte proof of concept en ontwikkelrapport opgeleverd.

Tabel 5.2 RAD-fasebeschrijving

6 Organisatie

In dit hoofdstuk wordt een beschrijving van de organisatie gegeven en de plek binnen het bedrijf waar de stagiair werkzaam is. ICT is sinds haar oprichting in 1978 werkzaam in een groot aantal uiteenlopende markten zoals bijvoorbeeld energie, water & infrastructuur, healthcare en logistiek. Voor al haar diverse markten heeft ICT verschillende afdelingen opgericht die zich met hun eigen werkzaamheden bezighoud. In figuur 1 wordt een organogram van de ICT Group N.V. weergegeven. De rood gemarkeerde afdeling geeft aan waar de stagiair zijn opdracht uitvoert in het bedrijf.



Figuur 6.1 Organogram ICT Group N.V.

Belanghebbenden

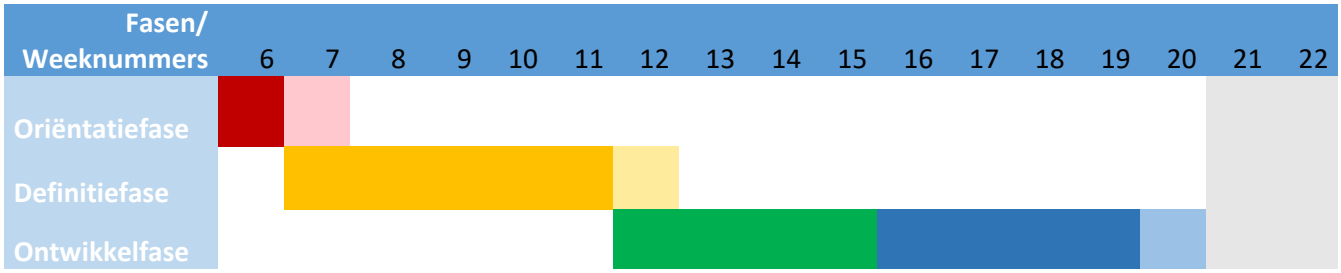
De volgende tabel laat zien wie er allemaal betrokken zijn bij deze opdracht vanuit de organisatie ICT Group.

Naam	Functie
Ramiro Jagai	Stagiair
André van ‘t Hof	Projectmanager
Bart Lamot	Innovation manager

Tabel 6.1 Belanghebbenden in het project

7 Planning

Dit hoofdstuk beschrijft de planning van de fases die eerder in hoofdstuk 5.3 zijn opgesteld. Het is lastig in te schatten hoeveel tijd er nodig is voor bepaalde onderdelen in fases. In Tabel 7.1 wordt er een globale planning weergegeven. Er wordt met lichtere kleuren een maximale uitlooptijd aangegeven van een fase. De twee kleuren in de ontwikkelfasen geven de verschillende incrementen aan. In de planning staan aan het eind twee weken alvast gereserveerd voor het afronden van het afstudeerverslag.



Tabel 7.1 Activiteitenplanning

Mijlpalen

In de volgende tabel worden de mijlpalen van dit project weergegeven.

Product	Opleverdatum
Plan van aanpak	**13 – 2 – 2017**
Definitierapport	**17 – 3 – 2017**
Ontwikkelrapport increment 1	**14 – 4 – 2017**
Testrapport increment 1	**14 – 4 – 2017**
Ontwikkelrapport increment 2	**12 – 5 – 2017**
Testrapport increment 2	**12 – 5 – 2017**
Proof of concept (POC)	**19 – 5 - 2017**
Afstudeerverslag	**2 – 6 – 2017**

Tabel 7.2 Mijlpalen

8 Risico's

Bij een project komen er altijd risico's kijken die een invloed kunnen hebben op het project. Hieronder volgt een lijst van verschillende risico's met een vermelding van de kans is dat een risico zal optreden. Met daarbij een omschrijving van de maatregelen die genomen kunnen worden om de risico's te voorkomen.

Risico	Kans	Maatregel
Deadline niet haalbaar	Middel	Door middel van het maken van een ruime planning vooraf wordt tegengegaan dat de deadline niet behaald wordt. Ook door het op tijd aangeven door de student aan de begeleider dat hij vastloopt op een probleem, kan er al vroeg hulp worden ingeschakeld.
Gebrek aan hardware	Klein	ICT zorgt voor de hardware die nodig is voor het uitvoeren van het project. De meeste componenten kunnen worden aangevraagd bij de helpdesk van ICT. Als er hardware nodig is die niet bij de helpdesk aan te vragen is kan er aan de begeleider worden gevraagd of het mogelijk is om de hardware te krijgen. Vraag dit tijdig aan bij de begeleider vanwege de kans dat het veel tijd gaat kosten om de hardware te regelen.
Afwezigheid begeleider	Middel	Het kan altijd zo zijn dat de begeleider afwezig is van of gewoonweg heel druk heeft. Als maatregel kan de stagiair aan andere collega's om hulp vragen die wel aanwezig zijn.
Verlies van documentatie of code	Klein	Het is altijd een mogelijkheid dat een systeem crasht en documentatie of code daarmee verloren gaat. Om dit te voorkomen wordt er gewerkt op OneDrive van Microsoft.
Problemen met opzetten van testomgeving	Groot	Het interne netwerk van ICT is goed beveiligd. Er mogen niet zomaar apparaten aangesloten worden op het netwerk. Er moet een aanvraag gedaan worden met het MAC-adres van het apparaat bij de helpdesk. Het risico is makkelijk te voorkomen door al vroeg aan te geven bij de helpdesk dat de student bezig is met een afstudeeropdracht en de MAC-adressen door te geven als die bekend zijn.

Tabel 8.1 Risico's en Maatregelen

ICT Group N.V.

HET ONTWIKKELEN VAN EEN EXTRA BEVEILIGING VOOR NETWERKCOMMUNICATIE VAN KLANTEN NAAR DE CLOUD VAN ICT GROUP

Definitierapport

Technische Informatica
De Haagse Hogeschool
Versie 0.4

Verklarende woordenlijst

In de volgende lijst worden veel voorkomende afkortingen en woorden verklaard op alfabetische volgorde.

Woord	Betekenis
(X)OTP	(Time- of HMAC-gebaseerd) One-Time Password
Azure Cloud	Een service van Microsoft voor het hosten van websites, databases enz.
Biometrie	Unieke kenmerken van een persoon die gebruikt kunnen worden bij identificatie
BLE	Bluetooth Low-Energy
BR/EDR	Basic Rate/Enhanced Data Rate
CA	Certificaatautoriteit
ECC	Elliptische-Curve Cryptografie
ECDH	Elliptische-Curve Diffie-Hellman
FHSS	Frequency-Hopping Spread Spectrum
HTTPS	HyperText Transfer Protocol Secure
MFA	Multi-Factor authenticatie
OOB	Out-Of-Band
PKI	Public Key Infrastructuur
Raspberry Pi	Een compact programmeerbaar computer dat werkt op een enkele printplaat
RSA	Encryptie algoritme afkorting gebaseerd op de namen van de ontwikkelaars
Single Sign-On	Eenmalig inloggen op een bepaalde dienst
SPKI/SDSI	Simple Public-Key Infrastructure/Simple Distributed Security Infrastructure
SSL/TLS	Secure Sockets Layer/Transport Layer Security
U2F	Universal Second Factor
WOT	Web Of Trust

Inhoudsopgave

1 Inleiding

In de oriëntatiefase zijn gesprekken gehouden met de opdrachtgever en begeleider zodat er een helder beeld is van de opdracht. Er zal een start worden gemaakt aan de tweede fase van dit project, dit wordt de definitiefase genoemd. In de definitiefase wordt er een onderzoek gedaan naar het probleem en kunnen de systeemeisen worden vastgesteld. Volgens het document “Plan van aanpak” staan hier ongeveer vijf weken voor ingepland.

Dit rapport bevat de documentatie van de definitiefase. In dit document worden eerst hoofd- en deelvragen opgesteld. Daarna wordt er onderzoek naar het probleem gedocumenteerd en aan de hand van het onderzoek kunnen deel- en hoofdvragen worden beantwoord. Met behulp van de beantwoording van de hoofd- en deelvragen worden de systeemeisen samen met de opdrachtgever vastgesteld. Als laatste volgt er een conclusie van de definitiefase.

2 Onderzoeksvragen

In de definitiefase zal er onderzoek worden gedaan naar het probleem. Uit dit onderzoek moet duidelijk worden welke opties er mogelijk zijn voor het uitvoeren voor de opdracht. De opdrachtgever zal een keuze moeten maken in de verschillende opties. Om voor de opdrachtgever de keuze makkelijker te maken zullen er onderzoeksvragen worden opgesteld. De onderzoeksvragen bestaan uit hoofd- en deelvragen. De hoofdvragen kunnen alleen beantwoord worden aan de hand van de conclusies uit de deelvragen. De volgende vragen zijn opgesteld.

2.1 Hoofdvragen

- "Wat is de meest optimale extra beveiliging die met behulp van een Raspberry Pi 3 aan applicaties van klanten toegevoegd kan worden zonder de applicaties aan te passen?"
- "Op welke manier zou dit project bestaande cybersecurityoplossingen kunnen vervangen?"

2.2 Deelvragen

Versleutelde data onderscheiden

- "Hoe kan versleutelde data van een publieke dienst zoals Facebook worden onderscheiden?"

Uitzoeken Cryptografie

- "Welke cryptografiemogelijkheden zijn er en wat zijn de voor- en nadelen?"

Uitzoeken fysieke sleutel & Biometrische beveiliging

- "Wat voor soorten multifactor-authenticatiemogelijkheden zijn er en welke zijn er nodig voor dit project?"

Bepalen Systeemarchitectuur

- "Wat is de meest optimale systeemarchitectuur voor dit project en wat zijn de voor- en nadelen hiervan?"

De hoofdvragen en deelvragen zullen aan het eind van dit onderzoek worden beantwoord. Met behulp van de conclusies die zijn getrokken uit de hoofdvragen en deelvragen kunnen de eisen van de proof of concept samen met de opdrachtgever definitief worden vastgesteld.

3 Versleutelde data onderscheiden

Al het verkeer dat door de Raspberry Pi heen gaat moet worden versleuteld. Het is daarna de bedoeling dat alleen gebruikers die zijn ingelogd zijn op een raspberry Pi de versleutelde en niet-versleutelde data kunnen uitlezen. Gebruikers die niet via een Raspberry gebruik maken van “de dienst” kunnen de beveiligde data niet bekijken.

Het is niet mogelijk het systeem van de klant te gebruiken voor dit project. De opdrachtgever heeft toen besloten om op een andere publieke dienst een extra beveiliging op te bouwen. Er is besloten om Facebook te gebruiken om het concept te testen. Om een extra beveiliging op Facebook te ontwikkelen moet er eerst worden onderzocht hoe het netwerkverkeer met de server van Facebook eruitziet. Facebook maakt gebruik van een https-only standaard, dat betekent dat al het netwerkverkeer van Facebook beveiligd is met een encryptie.

Er is een aantal manieren geprobeerd om het https-netwerkverkeer uit te lezen:

- Met Wireshark het netwerkverkeer onderscheppen en bekijken met een Pre-master key of SSLstrip
- Met MITMProxy al het netwerkverkeer door een SSL-interceptie proxy op te vangen. Er wordt een eigen certificaat aangebracht op de cliënt-pc die een verbinding opzet naar de proxy. De proxy pakt dan het verkeer uit en opnieuw weer in. Vervolgens zorgt de proxy voor een verbinding met facebook.com.
- Met Owasp ZAP, dat als een browser-proxy werkt, het netwerkverkeer op een applicatieniveau de berichten bekijken.

Al de bovenstaande methodes zijn gebruikt om het netwerkverkeer te kunnen bekijken van facebook.com. Met geen van de bovenstaande methodes is het echt gelukt om precies te achterhalen hoe facebook zijn berichten verwerkt. Er zal veel tijd verloren gaan met het analyseren van facebookverkeer en dat valt niet binnen de afbakening van de afstudeeropdracht. Er is samen met de opdrachtgever en begeleider besloten om een andere testomgeving te zoeken. De testomgeving kan een andere publieke dienst zijn of er wordt zelf een omgeving opgezet. De kans is hoog dat dezelfde problemen kunnen opkomen bij het selecteren van een andere publieke dienst. Er is afgezien van het gebruik van een andere publieke dienst en samen met de begeleider is de keuze gemaakt om zelf een dienst op te zetten waar de student alle rechten op heeft. De begeleider heeft de student toegang tot de Azure cloud van ICT gegeven, zodat er op Azure een eigen webapp ontwikkeld kan worden. De testomgeving kan dan altijd nog worden uitgebreid of vervangen.

4 Uitzoeken gebruik cryptografie

Encryptie betekent het versleutelen van het bericht dat verzonden wordt. Alleen de persoon met een “sleutel” kan het bericht lezen. Als derden het bericht onderscheppen kunnen zij het bericht niet uitlezen zonder dat ze de sleutel bezitten. Het gebruik van encryptie wordt heden ten dage als een hoge prioriteit gezien. Op het moment wordt er vaak met gevoelige data gewerkt over het internet. Bedrijven en banken zijn afhankelijk van een “goede” beveiliging van hun diensten. Moderne digitale cryptografie kan worden verdeeld in symmetrische- en asymmetrische encryptie. In dit hoofdstuk wordt er analyse gedaan van verschillende encryptiemethoden en een keuze gemaakt voor het gebruik van een encryptie in de proof of concept.

4.1 Symmetrische encryptie

Een symmetrische encryptie is het versleutelen van berichten waarvoor bij het ontcijferen van het bericht één sleutel nodig is. Een voordeel van het gebruik van een symmetrische cryptografie is dat het “encrypten” en “decrypten” sneller is dan bij asymmetrische cryptografie. Een nadeel is dat beide partijen in bezit moeten zijn van de sleutel. De sleutel moet dan op een veilige manier naar de andere partij worden verzonden.

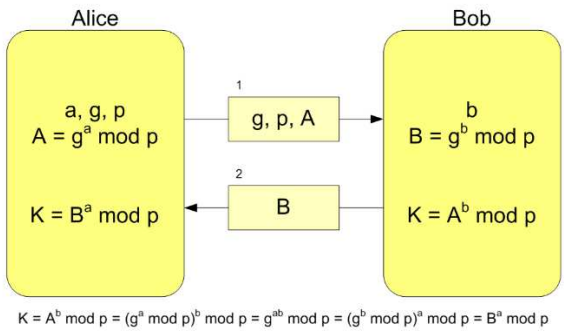
Een aantal hedendaagse cryptografieën hebben meegedaan aan de AES-competitie in het jaar 2000. Deelnemers moesten een oplossing bedenken voor het verouderde encryptie algoritme DES. In de volgende lijst wordt een aantal veel gebruikte symmetrische cryptografieën weergegeven:

- **3DES** – (Triple **D**ata **E**ncryption **S**tandard). Dit algoritme is eigenlijk het drie keer toepassen van een normale DES-encryptie achter elkaar. DES maakt gebruik van een 64-bit sleutel. Deze sleutel is zo klein dat hij makkelijk te kraken is. De uitbreiding naar 3DES zorgt voor een 168-bit sleutel. De sleutel kan niet meer zo makkelijk worden gekraakt.
- **Serpent** – Het Serpent algoritme is tweede geworden bij de AES-competitie. Het algoritme gebruikt datablokken van 128-bit en 128, 192 of 256-bit sleutels. Het algoritme moet 32 encryptierondes doorlopen.
- **AES** – (**A**dvanced **E**ncryption **S**tandard) van Rijndael is bekroond tot opvolger van het DES-algoritme. Het heeft een sleutelruimte van 128,192 of 256-bit en gebruikt datablokken van 128-bit. Het algoritme gebruikt 10,12 of 14 encryptie rondes. De grootte van de sleutel bepaalt het aantal encryptierondes
- **Twofish** – Dit algoritme is de opvolger van het eerdere Blowfish-algoritme dat een van de vijf finalisten was in de AES-competitie. Het gebruikt, net als AES, 128-bit datablokken en 128,192 of 256-bit sleutels. Het algoritme gebruikt 16 encryptierondes.

Als we kijken naar de bovenstaande algoritmes is duidelijk te zien dat Rijndael’s AES het meest veilig, maar ook het snelst in gebruik is. Om deze reden is het Rijndael algoritme ook als AES benoemd. AES zal worden gebruikt om symmetrisch data te encrypten.

4.2 Asymmetrische encryptie

De meeste hedendaagse asymmetrische cryptografieën zijn gebaseerd op het Diffie-Helman algoritme. Het Diffie-Hellman algoritme zorgt ervoor dat twee partijen zonder voorkennis over een onbeveiligde verbinding een sleutel kunnen uitwisselen om een beveiligde verbinding op te zetten



Figuur 4.1 Diffie-Helman algoritme

In Figuur 4.1 wordt een de werking van het Diffie-Hellman algoritme weergegeven. Hieronder wordt in korte stappen beschreven wat er precies gebeurt in Figuur 4.1.

1. Alice bedenkt een priemgetal p en een willekeurig getal a en g.
2. Door middel van modulair machtsverheffen wordt het getal A berekend.
3. De waardes van A, p en g worden verzonden naar Bob.
4. Bob verzint een willekeurig getal b en berekent daarmee de waarde van B.
5. Bob berekent ook de waarde van sessiekey K met behulp van b.
6. Bob verzendt de waarde van B naar Alice.
7. Alice berekent de waarde van sessiekey K met behulp van B en a.

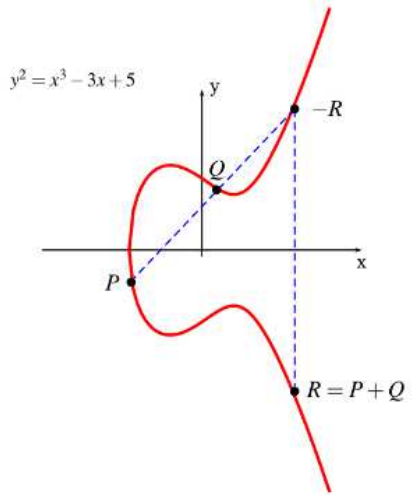
De waardes a en b zijn hierin de privé-sleutels en zonder deze sleutels kan de sessiekey K niet snel worden berekend. Een derde afluisterende partij zou net zo lang met g moeten vermenigvuldigen totdat de waarde A of B verschijnt. Dit is misschien nog wel te doen met kleine getallen, maar als het gaat om grote getallen kan dit jaren duren. Dit wordt ook wel het discrete logaritme probleem genoemd.

Bij moderne asymmetrische encryptie worden er twee sleutels gebruikt voor de encryptie en decryptie. Er zijn publieke sleutels en privé-sleutels. De publieke sleutel is voor de wereld bekend en wordt gebruikt voor het encrypten van data, maar alleen de privé-sleutel kan worden gebruikt voor het decrypten van de data. De privé-sleutel is alleen aan de eigenaar bekend. Hieronder volgen een aantal voorbeelden van populaire asymmetrische encryptie algoritmes:

- RSA – Op dit moment het meest gebruikte asymmetrische encryptie algoritme. Het wordt wel vereist om grote getallen te gebruiken voor de sleutel. Bij het gebruik van een lage waarde voor de sleutel zouden computers de sleutel kunnen ontcijferen.
- ECC (**E**lliptisch **C**urve **C**ryptografie) - Het ECC is een opvolger van het RSA-protocol. Het ECC-algoritme wordt alleen nog niet gebruikt als standaard. Ondanks dat het sneller werkt dan het RSA-protocol.

RSA

Het RSA-protocol is ontwikkeld door Ron Rivest, Adi Shamir en Len Adleman. De eerste letters van hun achternamen zijn gebruikt om het protocol te benoemen (**R**ivest, **S**hamir, **A**dleman). Het RSA-protocol gebruikt een public key die met iedereen gedeeld kan worden en één private key die alleen aan de eigenaar bekend is. De public key wordt gebruikt om berichten te encrypten, maar alleen een private key kan de berichten decrypten. De kracht van het van het RSA-protocol ligt in de sleutelgrootte. Door gebruik te maken van grote waardes voor de sleutelgrootte zorgt ervoor dat een afluisterende partij er jaren over doet om de juiste sleutel te vinden. Terwijl de eigenaar van de private key het bericht in enkele seconden kan uitlezen, ook wel het discrete logaritme probleem.



Figuur 4.2 voorbeeld elliptische curve

Elliptisch Curve Cryptografie

Het ECC-algoritme is nog jong en wordt nog niet veel gebruikt. Terwijl het protocol wel wordt gezien als een opvolger van het RSA-protocol. Het ECC-protocol maakt gebruik van een elliptische curve ($y^2 = x^3 - ax + b$).

In Figuur 4.2[20] wordt een curve weergegeven met de volgende elliptische formule $y^2 = x^3 - 3x + 5$. Op de curve wordt een lijn getrokken vanaf een basis punt (P) en alle punten waar de lijn de curve kruist (Q, -R) worden gebruikt om R te berekenen. Er wordt daarna nog een lijn getrokken van P naar R en waar de lijn weer de curve kruist wordt opnieuw een punt berekent. Dit proces wordt een bepaalde hoeveelheid keren uitgevoerd totdat er een waarde wordt bereikt dat voldoende beveiliging biedt. De kracht van ECC is dat met P en Q de waarde van R makkelijk en snel kan worden berekend, maar met R is het heel lastig om P en Q te berekenen. Dit wordt ook wel het elliptische curve logaritme probleem genoemd.[21]

Het is bewezen dat het protocol sneller en zuiniger met geheugen en processorkracht is vergeleken met RSA[22]. Een RSA-sleutel met een 3072-bit beveiliging heeft ECC daarentegen een 256-bit sleutel nodig om hetzelfde niveau van beveiliging te behalen.

Hieronder wordt in stappen algemeen beschreven hoe ECC wordt gebruikt in combinatie met Diffie-Helman. Dit protocol wordt ook wel Elliptisch Curve Diffie-Hellman (ECDH) genoemd en wordt gebruikt voor het uitwisselen van sleutels.[23]

1. Alice en Bob komen overeen op welke elliptische curve wordt gebruikt en een “basis punt” P op de curve.
2. Alice selecteert een groot geheim nummer a.
3. Bob selecteert een groot geheim nummer b.
4. Alice berekent A=a*P en stuurt alleen het antwoord naar Bob.
5. Bob berekent B=b*P en stuurt alleen het antwoord naar Alice.
6. Alice berekent a*(B*P).
7. Bob berekent b*(A*P).
8. Sessiesleutel = b*(A*P) = a*(B*P).

Beide partijen zijn uiteindelijk in bezit van de sessiesleutel. Een afluisterende partij kan de sessiesleutel niet berekenen zonder de waarde a of b. Om de sessiesleutel te berekenen zou de afluisterende partij een voor een waardes van a of b moeten proberen totdat het antwoord van A of B is bereikt. Als dit getal groot genoeg is kan dit proces ook jaren duren. Het ECDH-protocol heeft als voordeel dat de waarden a of b minder groot hoeven te zijn vergeleken met het RSA-protocol.

Beide protocollen zijn veilig te gebruiken voor het encrypten van data. Als er wordt gekeken naar de snelheid van het encryptie protocol dan is ECC het beste te gebruiken. ECC heeft een minder grote encryptie-sleutel nodig voor het encrypten van data om hetzelfde niveau van beveiliging te krijgen als RSA. Hierdoor is ECC ook

[20] *http://www.embedded.com/design/safety-and-security/4396040/An-Introduction-to-Elliptic-Curve-Cryptography*
[21] James Earle - CSBreakdown – Brock university - Elliptic Curve Cryptography & Diffie-Hellman
[22] *RSA vs ECC Comparison for Embedded Systems – Atmel - http://www.atmel.com/Images/Atmel-8951-CryptoAuth-RSA-ECC-Comparison-Embedded-Systems-WhitePaper.pdf*
[23] Martijn Grooten - Elliptic Curve Cryptography for those who are afraid of maths

zuiniger in het gebruik van geheugen en processorkracht. Ondanks al deze voordelen bij het gebruik van ECC biedt RSA meer betrouwbaarheid. Het protocol RSA bestaat veel langer dan het ECC-protocol en de kans dat er in de toekomst een fout in het algoritme ECC wordt ontdekt is veel groter dan bij het RSA-protocol. Ondanks RSA meer betrouwbaarder is, biedt ECC veel meer snelheid. Als er een fout in de toekomst wordt ontdekt in het ECC protocol, dan is het simpel een andere certificaat te creëren met een RSA encryptie.

4.3 Infrastructuur

Bij een asymmetrische encryptie is het belangrijk dat de ontvanger beschikt over de juiste public key. Dit wordt gedaan door middel van het opzetten van een public key infrastructuur (PKI). De volgende punten geven een aantal manieren weer om een PKI-omgeving op te zetten.

- **Certificaten PKI** – Werkt op basis van certificaten en is op het moment het meest gebruikte PKI op het Internet. Een derde vertrouwde partij verifieert door middel van certificaten de public key van een instantie. Dit certificaat moet ook beveiligd aangeleverd worden met een public key encryptie. Er ontstaat een keten die gemanaged wordt.
- **Blockchain PKI** – Het gebruik van aaneen geketende hash-kettingen om certificaten te verifiëren. Er is dan geen derde vertrouwde partij meer nodig. Deze manier van certificaten verifiëren bevindt zich nog in een conceptfase en wordt nog niet veel gebruikt.
- **Web of Trust (WOT)** – Iedereen kan certificaten verifiëren en is vertrouwd op basis van het vertrouwen van anderen. Het WOT wordt veel gebruikt binnen het PGP-protocol.
- **Simple PKI/ Simple Distributed Security Infrastructure (SPKI/SDSI)** – Het linken van namen en toegang met sleutels. Het idee is nooit verder gekomen dan een conceptfase.
- **Kerberos** – Authenticatie protocol dat werkt met een externe server dat tickets uitgeeft aan de gebruikers zodat ze hun identiteit kunnen aantonen.

De meest gebruikte manier van het opzetten van een PKI gebeurt met gesigneerde certificaten van een extern vertrouwd bedrijf. Als een opdrachtgever een infrastructuur wil die met een certificaatautoriteit (CA) werkt dan kan een certificaten PKI worden gebruikt.

Kerberos is een authenticatieprotocol dat met behulp van een vertrouwde derde partij een Ticket Granting Ticket (TGT) uitdeelt aan gebruikers. De vertrouwde derde partij wordt ook wel een Key Distribution Center (KDC) genoemd. Met een TGT kan de gebruiker zich identificeren bij de KDC en kan een Session Ticket worden aangevraagd. Een Session Ticket verleent toegang tot een service die aangesloten is op het Kerberos-netwerk. Het Session Ticket is dan een beperkte tijd geldig en maakt Single Sign-in mogelijk.

Besluit Infrastructuur

Om een selectie te maken uit de mogelijke PKI's wordt er gekeken wat nodig is voor deze opdracht. De opdrachtgever wil graag aan klanten een product presenteren dat het meeste veiligheid biedt. De Blockchain PKI en SPKI/SDSI bieden op dit moment het minst zekerheid aan als het gaat om veiligheid. De Blockchain PKI bevindt zich op dit moment in een conceptfase, maar kan zich in de toekomst ontwikkelen tot een bruikbaar PKI. Het protocol Blockchain is nog vrij nieuw en het blockchain-protocol in combinatie met een PKI is nog niet grondig getest. De SPKI/SDSI-omgeving is een oud idee dat nooit verder gekomen is dan een conceptfase en het is mogelijk om met het huidige certificaten PKI een SPKI/SDSI-omgeving op te zetten.

Een Web of Trust is een meer gebruikt PKI dan de vorige twee. Het WOT wordt veel gebruikt met het PGP-protocol. Alleen is het vertrouwen van een WOT-PKI gebaseerd op individuelen in plaats van een centraal bedrijf dat certificaten signeert. Dit model is goed te gebruiken als de opdrachtgever een infrastructuur zou willen die bouwt op het vertrouwen van anderen. Dit neemt het risico weg dat een CA beïnvloed zou zijn.

Kerberos is een authenticatieprotocol dat alleen de identiteit van een gebruiker kan aantonen. Het protocol versleutelt het verkeer dat over het netwerk heen gaat niet. Dit maakt het protocol kwetsbaar voor een MITM -

aanval. Het wordt dan ook vaak toegepast in combinatie met het certificaten PKI om zo de veiligheid van de data te waarborgen.

De kans dat een CA is beïnvloed is klein en in een bedrijfsomgeving is het van belang wie er in vertrouwen wordt genomen. Binnen een WOT is dat niet altijd duidelijk, om deze reden valt het WOT-PKI af. Blockchain PKI is nog in ontwikkeling en is nog niet goed getest. Het is belangrijk een veilige en een vertrouwde omgeving op te leveren. Er is daarom gekozen voor een Certificaten PKI-omgeving, omdat deze het meest gebruikt is op het moment. Een SPKI/SDSI is een oud concept dat ook kan worden gerealiseerd met een certificaten PKI. Een certificaten PKI is op het moment de meest gebruikte infrastructuur en geeft voldoende bescherming voor de omgeving waarop de Proof of Concept moet draaien.

5 Onderzoek fysieke sleutel

Het toegang verlenen met alleen een wachtwoord wordt tegenwoordig als niet-veilig beschouwd. De wachtwoorden kunnen te zwak zijn of iemand zou met behulp van bepaalde aanvalstechnieken het wachtwoord kunnen achterhalen. Een oplossing hiervoor is multifactor-authenticatie (MFA). MFA is het gebruik van een wachtwoord in combinatie met iets dat je bezit of bent waarmee toegang verleend kan worden tot een bepaalde dienst.

De opdrachtgever heeft aangegeven dat er een MFA gebruikt moet worden in de proof of concept. Het concept zou bedrijven aan moeten tonen dat applicaties of diensten extra beveiligd kunnen worden zonder dat de applicatie zelf aangepast wordt. In dit hoofdstuk wordt vooral de MFA “What you have” uitgewerkt. De volgende mogelijke manieren in de factor “What you have” zijn gevonden:

- **Bluetooth authenticatie** – Het verbinden van een apparaat via Bluetooth. Het wachtwoord is het apparaat dat verbindt met de Raspberry Pi.
- **OTP** (**O**ne-**T**ime **P**assword) – Het genereren van een op tijd of hash gebaseerd wachtwoord.
- **U2F** (**U**niversal **2**nd **F**actor) – Een soort USB-sleutel die een private-key bewaart. Door het drukken op de knop van de USB-sleutel wordt er een bericht gesigneerd met de private-key.

Er bestaan andere manieren om een MFA te implementeren. Deze drie methodes zijn gekozen omdat de opdrachtgever graag MFA als een soort fysieke sleutel wil hebben. De opdrachtgever gaf het volgende aan: “Het moet werken net als bij de bank, je hebt een pincode (Something you know) en je bankpas (Something you have)”. Deze methodes sluiten het meest aan bij wat de opdrachtgever wil hebben. Bij het eerste interview wordt aangegeven dat de fysieke sleutel een USB-stick mag zijn of een smartphone.

5.1 Bluetooth authenticatie

Bluetooth is tegenwoordig te vinden in veel alledaagse apparaten zoals op de laptops, smartphones, koptelefoons en zelfs horloges. Bluetooth is een communicatieprotocol, het protocol zorgt voor een draadloze verbinding tussen twee apparaten op korte afstand. Het protocol is ontwikkeld in 1994 door het bedrijf Ericsson, maar sinds 1998 wordt verdere ontwikkeling gedaan door de Special Interest Group (SIG). Bluetooth werkt op een 2.4Ghz frequentie en maakt gebruik van de Frequency-Hopping Spread Spectrum (FHSS) techniek om data te verzenden. FHSS zorgt ervoor dat de data in kleine stukken verzonden wordt en dat die verzonden wordt binnen een bepaalde radiofrequentie. Binnen een seconde wordt tot aan maximaal 1600 keer gehopt naar een ander kanaal binnen de radiofrequentie om te voorkomen dat een kanaal te druk wordt. Het hoppen is ook een soort beveiliging tegen een meeluisterende partij.

Vanaf bluetooth 4.0 is er een aftakking gekomen in het protocol. Het standaard bluetooth protocol dat meer bedoeld is voor het streamen van een hoge data hoeveelheid wordt BR/EDR genoemd. Een nieuwere variant is Bluetooth Smart ook wel Bluetooth Low-Energy (BLE) genoemd. Het Bluetooth LE-protocol verreist minder batterijvermogen, is vele malen sneller, maar kan geen hoge data hoeveelheden verzenden. Het protocol is ontwikkeld door de SIG met de bedoeling dat het veel gebruikt zou worden binnen de IoT-omgeving, sport en security. Bij het gebruik van Bluetooth als second-factor authenticatie kan bijvoorbeeld een apparaat worden gebruikt dat iemand altijd bij zich heeft. Tegenwoordig heeft bijna iedereen een smartphone die als tweede factor kan dienen in een multi-factor authenticatie proces.

Bij Bluetooth Low-Energy 4.0 en 4.1 ook wel LE Legacy genoemd wordt er gebruik gemaakt van het Secure Simple Pairing model. In deze versie van BLE kan er worden gekozen uit Just Works, Passkey Entry en OOB, het ligt aan het apparaat welke methode gekozen wordt. Vanaf Bluetooth Low-Energy 4.2 ook wel LE Secure Connections genoemd is daar verandering in gekomen. SIG heeft een sterkere methode Nummeric Comparison toegevoegd aan de bestaande methodes. De Nummeric Comparison methode maakt gebruik van het ECDH-algoritme voor het sleutel uitwisselingproces, dit is een veel sterker sleutel uitwisselingsalgoritme vergeleken

met de andere methodes[24]. De data over een BLE-verbinding is encrypted met AES-CCM. De klassieke Bluetooth (BR/EDR) is veel minder "uitgekleed" en heeft sinds 4.1 al de mogelijkheid tot Secure Connections

Begin 2017 heeft de SIG Bluetooth 5 uitgebracht. Er wordt aangegeven dat deze versie van bluetooth 2x sneller is, 4x meer bereik heeft en 8x meer broadcast ruimte heeft[25]. Ondanks de verbeterde beveiliging geven critici aan dat hackers altijd wel een mogelijkheid hebben ontdekt om bluetooth te misbruiken. DeviceLock geeft het volgende aan "2x the speed + 4x the range = 8x the security risk"[26]. DeviceLock bedoelt hiermee dat door het uitbreiden van het bereik in bluetooth 5 hackers vanaf een grotere afstand en sneller Bluetooth zouden kunnen misbruiken. De korte afstand die nodig was om een Bluetooth verbinding tot stand te brengen zorgde voor een vorm van beveiliging. Hackers kunnen met Bluetooth 5 al van ver het Bluetooth verkeer onderscheppen.

Hieronder in Tabel 5.1 worden de beveiliging voor- en nadelen van de verschillende versies van Bluetooth Low-Energy weergegeven.

Bluetooth Low-Energy	Bluetooth Legacy (4.0 & 4.1)	Bluetooth Secure Connections (4.2)	Bluetooth 5
Voordelen	- Sterke encryptie	- Sterke sleuteluitwisseling - Sterke encryptie	- Sterke sleuteluitwisseling - Sterke encryptie
Nadelen	- Zwakke sleuteluitwisseling	- Geen	- Groot bereik

Tabel 5.1 Beveiliging Voor- en Nadelen van Bluetooth Low-Energy

In Tabel 5.2 worden de beveiliging voor- en nadelen van de verschillende versies van klassieke Bluetooth (BR/EDR) weergegeven.

Klassieke Bluetooth (BR/EDR)	Bluetooth 4.1	Bluetooth 4.2	Bluetooth 5
Voordelen	- Sterke sleuteluitwisseling - Sterke encryptie	- Sterke sleuteluitwisseling - Sterke encryptie	- Sterke sleuteluitwisseling - Sterke encryptie
Nadelen	- Geen	- Geen	- Geen

Tabel 5.2 Beveiliging Voor- en Nadelen van klassieke Bluetooth

Uit de informatie van beide tabellen kan worden vastgesteld dat klassieke Bluetooth het best te gebruiken is als het apparaat alleen tot aan Bluetooth 4.1 wordt ondersteund. Als het apparaat dat beveiligd moet worden het protocol Bluetooth 4.2 ondersteunt dan zou klassieke (BR/EDR) en BLE Secure gebruikt kunnen worden. Bij Bluetooth 5 wordt er LE long-range gebruikt waardoor bluetoothverkeer vanaf grotere afstand kan worden opgevangen. Dit is nadelig bij het toepassen van een MFA, omdat het dan juist de bedoeling is dat de gebruiker het Bluetoothverkeer gebruikt als een authenticatie middel.

5.2 One-Time Password

OTP is een algoritme voor het generen van een wachtwoord dat één keer geldig is voor een authenticatieproces. Vaak wordt deze authenticatiemethode gebruikt in combinatie met MFA. Er zijn verschillende soorten OTP's. Bij HMAC-based One-Time Password (HOTP) wordt er een wachtwoord gegenereerd gebaseerd op een HASH bericht. Het wachtwoord is geldig voor een onbepaalde tijd. Bij Time-

[24] Bluetooth-low-energy-security.pdf – Bluetooth SIG
[25] Bluetooth-5-faq.pdf – Bluetooth SIG
[26] http://www.devicelock.com/blog/2842.html - DeviceLock DLP

based One-Time Password (TOTP) wordt er een op tijd gebaseerd wachtwoord gegenereerd. Het wachtwoord heeft vaak een beperkte geldigheidsduur. Dit maakt TOTP veiliger in gebruik vergeleken met HOTP.

OTP kan op verschillende manieren worden geïmplementeerd. Er bestaan tokens die de code kunnen genereren. Het kan ook zo zijn dat de code wordt opgestuurd via sms naar een telefoon of via een app op een smartphone. Al deze apparaten werken op batterijen en als deze leeg is kunnen gebruikers niet meer inloggen. Dit is vaak te voorkomen door bijvoorbeeld op tijd de telefoon op te laden. Bij het gebruik van een app kan de softwareversie niet meer up-to-date zijn. Als een gebruiker op het moment van inloggen geen toegang heeft tot het internet wordt het lastig de app nog bij te werken.

Het OTP-protocol heeft zijn kwetsbaarheden. De wachtwoorden zouden kunnen worden achterhaald met behulp van een phishing-aanval. Het protocol zelf heeft geen bescherming tegen dit soort aanvallen, maar TOTP heeft als voordeel dat de tijdsduur van een code heel beperkt is en dus zou de aanvaller in real-time zijn aanval moeten uitvoeren.

Hieronder worden de voor- en nadelen van het OTP-protocol in punten beschreven.

Voordelen:

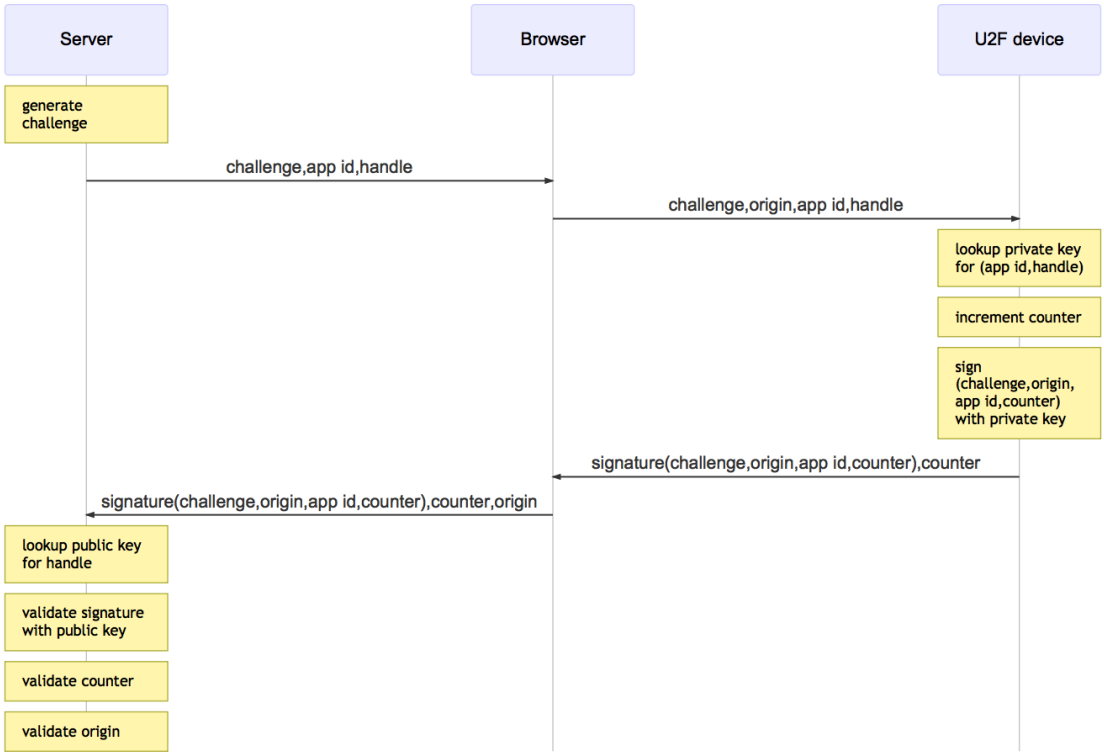
- **Makkelijke implementatie** – Het is niet moeilijk op te zetten. Er bestaan apps op de smartphone die deze dienst aanbieden.
- **Beperkte tijdsduur TOTP** – Bij het gebruik van TOTP zijn de wachtwoorden heel beperkt geldig, waardoor het nog lastig wordt een aanval uit te voeren. Een aanval zou in real-time moeten gebeuren.

Nadelen:

- **Werkt op batterij** – Apparaten waarop de token op wordt gegenereerd werken op batterijen. Als deze op is kan de gebruiker niet meer inloggen. Tenzij er een mogelijkheid is om het apparaat op te laden of batterijen te vervangen.
- **Verouderde software** – Als het protocol gebuikt wordt op een smartphone app dan kan het voorkomen dat de software verouderd is. Dit kan ervoor zorgen dat de codes die gegenereerd worden niet meer overeenkomen met elkaar.
- **Geen phishing-bescherming** – Het protocol heeft geen bescherming tegen phishing-aanvallen. Bij het protocol TOTP is door de beperkte geldigheidsduur van het wachtwoord lastiger te achterhalen wat het wachtwoord is op het moment.

5.3 U2F

U2F is een authenticatie standaard ontwikkeld door Google en Yubico. De standaard wordt nu ondersteunt door de FIDO Alliance. De U2F standaard is open-source en het U2F-apparaat lijkt veel op een USB-stick. Het grootste verschil met OTP en Bluetooth is dat het apparaat echt meedoet in het authenticatieproces. De browser communiceert met het U2F-apparaat over een usb-poort met challenges. Op de U2F-stick wordt een privé-sleutel opgeslagen die de challenges ermee signeert.



Figuur 5.1 Sequentiediagram U2F proces

In Figuur 5.1[27] wordt een sequentiediagram van het U2F proces weergegeven. Om het U2F proces te begrijpen zal er eerst een uitleg worden gegeven en de reden waarom bepaalde handelingen worden uitgevoerd.

Om een phishing-aanval te voorkomen wordt er door de browser een “origin” meegestuurd. De origin bevat de URL en hostnaam. Als de server een andere origin uitleest weet de server dat er een phishing-aanval wordt uitgevoerd. Om te voorkomen dat providers van de diensten kunnen bijhouden welke services een gebruiker gebruikt wordt er voor elke service een andere public- en privatekey paar aangemaakt. De server geeft een App_ID en Handle mee. Zo kan de U2F-stick de privatekey opzoeken die behoort bij de dienst die een challenge heeft verstuurd. U2F bevat ook een cloning bescherming. De U2F-stick zorgt ervoor dat er bij elke signering van een challenge een counter wordt opgeteld. De waarde van deze counter wordt opgeslagen op de server en als er bij een opvolgende challenge een te lage waarde wordt gedetecteerd dan heeft de server door dat de stick gekloond is. Hieronder wordt het U2F proces in stappen beschreven.

1. De server genereert en verstuurt een challenge met een App_ID en Handle.
2. De browser ontvangt de challenge en voegt daar een origin aan toe.
3. U2F-apparaat ontvangt de challenge en zoekt met behulp van de App_ID en Handle de juist privatekey.
4. U2F-apparaat incrementeert de counter en signeert de challenge.
5. U2F-apparaat stuurt gesigneerde challenge terug naar de browser en server.
6. De server valideert de gesigneerde challenge met zijn public-key
7. De server controleert de counter.
8. De server valideert de origin.

[27] https://robn.io/talks/u2f-lca-2017/U2F-notes.pdf - *U 2 can U2F - linux.conf.au 2017 – Hobart, Tasmania*

Voordelen:

- **Usability –** Het is simpel te gebruiken. De gebruiker steekt de sleutel in de USB-poort van de computer en met een druk op de knop is er een complex authenticatieproces afgerond.
- **Phishing-bescherming** – Het U2F protocol heeft een ingebouwde phishing-bescherming. Door middel van het meesturen van een Origin kan worden achterhaald of er data is aangepast.
- **Privacybescherming** – Door het aanmaken van een verschillende private- en public-key, wordt voorkomen dat providers kunnen zien van welke diensten een gebruiker gebruik maakt

Nadelen:

- **Zwakke cloning beveiliging –** Het optellen van een counter is geen sterke beveiliging tegen het kopiëren van de U2F-stick. Als de gekopieerde stick eerder gebruikt wordt dan de originele, dan zal de server de gekopieerde stick herkennen als de originele.

6 Onderzoek biometrische beveiliging

In dit hoofdstuk zal de derde factor van multifactor-authenticatie worden beschreven. De derde factor binnen MFA staat voor “Something you are”. Dit wordt ook wel biometrische beveiliging genoemd. Bij een biometrische beveiliging wordt een unieke eigenschap van een persoon gebruikt. Dit kan een vingerafdruk, de stem of het gezicht zijn.

De opdrachtgever heeft bij het eerste interview niet aangegeven dat het project een biometrische beveiliging moet hebben. Later tijdens het project is besloten deze wel toe te voegen aan het onderzoek. Er zullen eerst verschillende biometrische beveiligingsmethodes worden onderzocht en daarna wordt er bepaald of het nodig is deze toe te voegen aan het project.

De volgende biometrische beveiligingsmogelijkheden zijn gevonden:

- **Vingerafdrukherkenning** – Met behulp van een vingerafdrukscanner de vingerafdruk van de gebruiker vergelijken met een vingerafdruk in een database.
- **Stemherkenning** – Met behulp van een microfoon de stem van de gebruiker vergelijken met een opname in een database.
- **Gezichtsherkenning** – Met behulp van een camera het gezicht van de gebruiker vergelijken met een foto in een database.

Deze drie manieren zijn gekozen, omdat deze ook haalbaar zijn binnen het project. Bij een biometrische beveiliging komt er extra apparatuur aan te pas. De opdrachtgever heeft aangegeven de proof of concept “zo generiek en vervangbaar mogelijk” te ontwikkelen. De apparatuur mag dus niet te groot zijn. Een kleine camera, microfoon of een vingerafdrukscanner sluit nog wel aan bij de wensen van de opdrachtgever. Deze drie biometrische methodes sluiten daar bij aan.

6.1 Vingerafdrukherkenning

Met vingerafdrukherkenning wordt het proces bedoeld waar op basis van een vingerafdruk een persoon kan worden herkend. In het proces wordt er een vergelijking gedaan met eerder vastgelegde vingerafdrukken. Er bestaan verschillende methodes om vingerafdrukken met elkaar te vergelijken.

- **Afstandsmethode** – Het definiëren van verschillende punten in de vingerafdruk en het vaststellen van de afstand tussen die punten.
- **Huffmancodering** – De binaire code van de vingerafdrukafbeelding comprimeren met als uitkomst een unieke vector. De vector wordt dan vergeleken met andere vectoren in een database.

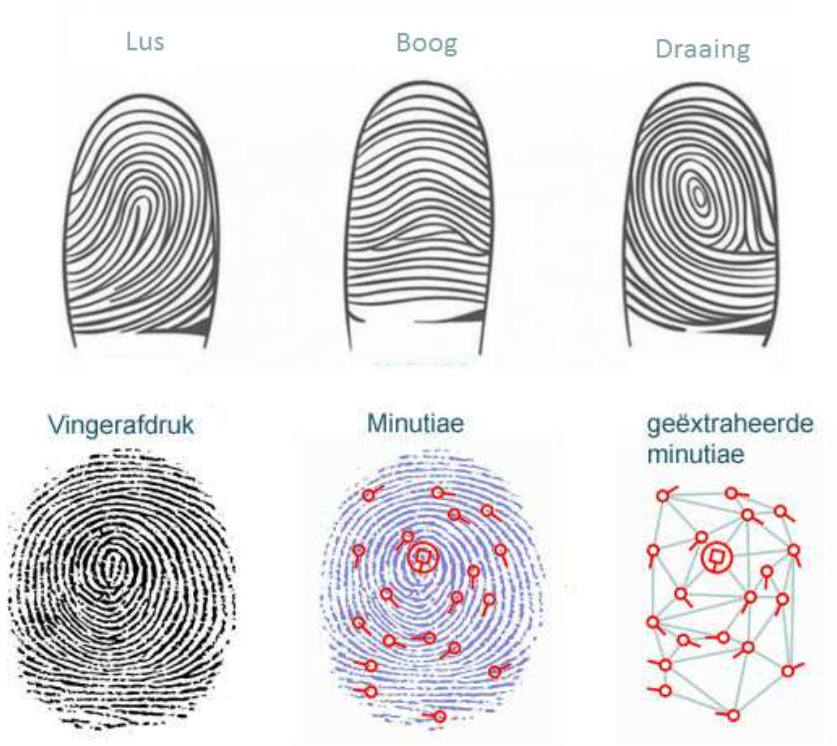
Vingerafdrukherkenning is een van de meest bekende biometrische identificatie-methodes. Het wordt al meer dan een eeuw gebruikt en tegenwoordig wordt het ook toegepast in smartphones[28]. Biometrie met vingerafdrukken heeft een hoge betrouwbaarheids- en veiligheidspercentage. De afdrukken blijven voor een persoon zijn gehele leven hetzelfde en kunnen niet makkelijk worden veranderd.

Er bestaan drie basispatronen van een vingerafdruk. De basispatronen zijn een lus-, boog- en draaiing-patroon. Binnen de drie basispatronen kunnen afgeleide patronen bestaan. Een vingerafdruk bestaat uit verschillende eigenschappen, ook wel minutiae genoemd. De afstandsmethode gebruikt de minutiae om herkenningspunten vast te stellen. De afstand tussen de herkenningspunten wordt gebruikt om te verifiëren of de herkenningspunten met elkaar overeenkomen. De Huffmancodering gebruikt de binaire code van de afbeelding om een Huffman-coding-compressie uit te voeren. Uit de compressie komen unieke waardes die kunnen worden vergeleken met andere opgeslagen Huffman-code-compressie waardes. Bij de Huffman-

[28] Fingerprint Recognition.pdf - FBI

methode is het van belang dat de plaatsing van de vingerafdruk hetzelfde is. Een kleine aanpassing in de afbeelding zorgt ervoor dat de unieke waardes niet meer overeenkomen.

In Figuur 6.1[29] wordt een voorbeeld van vingerafdrukpatronen en minutiae weergegeven.



Figuur 6.1 vingerafdrukpatronen en minutiae

De afstandsmethode lijkt voor dit project het meest bruikbaar. De Raspberry wordt gebruikt door verschillende gebruikers die moeten inloggen. Bij elke Raspberry nog een vingerafdrukscanner te plaatsen wil de opdrachtgever niet. De opdrachtgever heeft gevraagd om een zo generiek en vervangbaar mogelijke oplossing. Smartphones hebben tegenwoordig van zichzelf al een goede beveiliging. Als het nodig is een biometrische beveiliging toe te voegen aan de proof of concept, zou een oplossing het gebruik van de vingerafdrukscanner die in de smartphones zit kunnen zijn.

Hieronder worden de voor- en nadelen in punten weergegeven.

Voordelen:

- Het biedt een hoge nauwkeurigheid bij het bepalen van een identiteit
- De scanner zit vaak al in smartphones

Nadelen:

- Niet alle smartphones hebben een vingerafdrukscanner
- Bij gebruik van Huffman is dezelfde scanner nodig

[29] http://vemichron.eu/be/nl/eur/nieuws/algemeen/biometrie -- http://dailyvibes.org/fingerprint-reveal-personality/

6.2 Stemherkenning

Met stemherkenning wordt de stem van een persoon herkend om zijn identificatie vast te stellen. De stem van ieder mens is anders, maar deze kan bewust of onbewust veranderen. Stemherkenning kan op twee manieren worden toegepast.

- **Identificatie** – Op basis van karakteristieken van de stem achterhalen wie de persoon is.
- **Authenticatie** – Op basis van het patroon van wat de persoon zegt vergelijken met een al eerder opgenomen bericht.

Een stem heeft te weinig karakteristieken om nauwkeurig vast te stellen wie de persoon is. Dat neemt de betrouwbaarheid van stemherkenning weg. De omgeving speelt ook een grote rol in stemherkenning. Als de gebruiker zich in een luidruchtige omgeving bevindt kan de identiteit niet worden vastgesteld. Een persoon die ziek is of een op een andere manier niet meer dezelfde stem heeft kan ook niet inloggen op basis van stemherkenning.[30]

Een Raspberry Pi heeft geen audio-chip. Om geluid op te nemen kan er een USB-microfoon worden gebruikt. Er bestaan niet veel voorbeelden van identificatie via stemherkenning van en het is een redelijk onbekend terrein in beveiliging. Er bestaan beter bruikbare manieren om biometrisch een identiteit vast te stellen die een hogere betrouwbaarheidspercentage hebben.

6.3 Gezichtsherkenning

Met de eigenschappen van het gezicht kan de identiteit van een persoon worden bepaald. Er wordt met behulp van een eerdere opname bepaald of het de juiste persoon is. Er bestaan meerdere manieren voor het toepassen van gezichtsherkenning. Meestal wordt de verhouding van de oren, ogen, neus en mond gemeten. Een andere manier is met behulp van de temperatuur in het gezicht herkennen, dit is bij ieder mens anders.

Bij gezichtsherkenning met temperatuur wordt gebruik gemaakt van infraroodcamera's. De IR-camera kan de weefselstructuur en temperatuur op bepaalde punten herkennen. Dit is bij iedere persoon anders. Het mooie van het gebruik van infrarood is dat de emoties van de gebruiker geen invloed hebben in het authenticatie-proces. Een ander voordeel is het kunnen onderscheiden van gezichten op plaatjes tegenover echte gezichten. Een infraroodcamera wordt nog niet gebruikt in smartphones. De camera's zijn nog te duur om te implementeren in smartphones.[31]

Gezichtsherkenning die gebruik maakt van herkenningspunten in het gezicht is makkelijker toe te passen. De camera's hoeven niet duur te zijn, maar de kwaliteit van een opname speelt wel een grote rol bij het herkenningsproces. De smartphones hebben vaak camera's die ook een hoge kwaliteit camera's ingebouwd hebben. Soms zitten er zelfs 2 camera's ingebouwd, waarbij de één vooral bedoeld is voor selfies. Met het gebruik van een smartphone zou gezichtsherkenning op basis van herkenningspunten haalbaar zijn. Er hoeft dan geen externe camera te worden aangeschaft.

Het gezicht van de mens verandert naar mate iemand ouder wordt en er komt een punt dat het gezichtsherkenning-systeem de persoon niet meer herkent. Ook mag de persoon geen voorwerpen op het gezicht dragen zoals een bril.

Hieronder wordt duidelijk de voor- en nadelen bij het gebruik van gezichtsherkenning weergegeven.

[30] https://www.security.nl/posting/24772/Stemherkenning%3A+biometrie+op+afstand
[31] https://www.intechopen.com/books/reviews-refinements-and-new-ideas-in-face-recognition/thermal-infrared-face-recognition-a-biometric-identification-technique-for-robust-security-system

Voordelen:

- **Goedkoop** – Normale camera's zijn goedkoop en zitten vaak al in smartphones.
- **Usability** – De gebruiker hoeft zelf geen specifieke handeling te verrichten.

Nadelen:

- **Usability** – Als een persoon een bril draagt moet hij deze elke keer als hij probeert in te loggen afdoen.
- ***Duur –** Infraroodcamera's zijn prijzig.

* = Geldt alleen voor temperatuur gezichtsherkenning.

7 Systeemconcepten voor een mogelijke systeemarchitectuur

Na een onderzoek naar verschillende encryptiemethoden en multifactor-authenticatie zal er een keuze worden gemaakt voor een definitieve systeemarchitectuur. Om deze keuze te maken zullen een aantal mogelijke systeemarchitecturen naast elkaar gelegd en vergeleken. Om een afweging te maken uit de mogelijke architecturen worden de eisen van de opdrachtgever meegenomen in de afweging. Dit hoofdstuk beschrijft de systeemconcepten die een mogelijke architectuur kunnen bieden voor de proof of concept.

Opstellen Systeemconcepten

Om een systeemconcept te ontwerpen is er als eerste gekeken naar de infrastructuur waarop de Proof of Concept gaat draaien. De infrastructuur bepaalt de algehele beveiliging die het systeem heeft. Als de infrastructuur niet veilig is, dan is het gehele Proof of Concept niet veilig.

Na onderzoek naar versleuteling van data is gebleken dat symmetrische encryptie sneller is vergeleken met asymmetrische encryptie. Er zal om deze reden in elk systeemconcept symmetrische encryptie worden gebruikt. In hoofdstuk 4.1 wordt beschreven dat de symmetrische encryptie AES het best te gebruiken is als het gaat om veiligheid, snelheid en flexibiliteit. AES-encryptie zal in elk systeemconcept worden gebruikt voor het versleutelen van de daadwerkelijke data.

In hoofdstuk 4.3 zijn er verschillende infrastructuren onderzocht, waaruit de meest bruikbare infrastructuur als mogelijkheid voor dit project naar voor is gekomen. Het certificaten PKI zal worden gebruikt bij het ontwikkelen van een systeemconcept.

Als tweede stap is er gekeken naar de meest geschikte MFA voor de Proof of Concept. Het resultaat uit het onderzoek in hoofdstuk 5 heeft geen “beste” MFA opgeleverd. De onderzochte protocollen hebben alle hun voor- en nadelen. Wel is gebleken dat het U2F-protocol als enige tegenover de andere protocollen een vorm van beveiliging heeft tegen “phishing”-aanvallen. Dat maakt het U2F-protocol aantrekkelijker om te gebruiken.

Bij het onderzoek in hoofdstuk 6 zijn biometrische beveiligingsprotocollen onderzocht. Deze hebben alle ook weer hun voor- en nadelen. De biometrische beveiliging was er later als optie bijgekomen in de opdrachtscope, maar deze wordt alleen toegevoegd aan de Proof of Concept als er nog voldoende tijd over is. Als er later tijdens het project besloten wordt biometrische beveiliging toe te voegen worden de systeemconcepten daaraan bijgewerkt.

De volgende systeemconcepten lijken mogelijkheden te bieden voor dit project op een RPI 3.

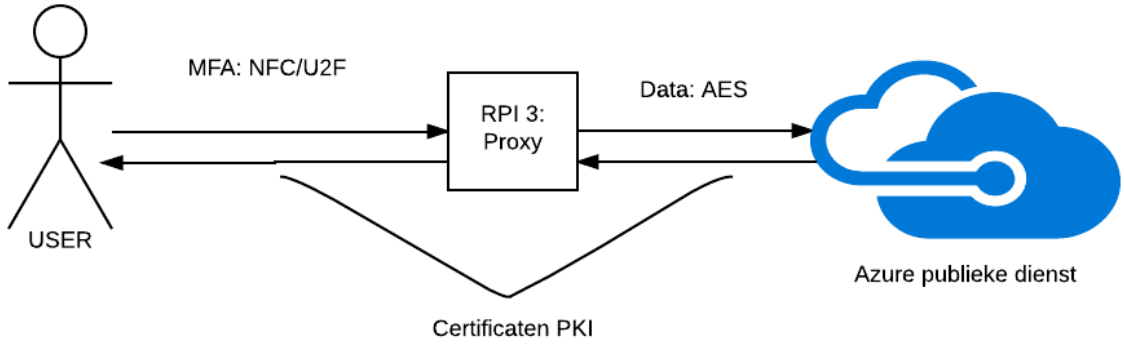
- Systeemconcept 1: Certificaten PKI & U2F-Stick
- Systeemconcept 2: Certificaten PKI & U2F-BLE
- Systeemconcept 3: Certificaten PKI & Bluetooth Klassiek

De concepten zijn opgesteld omdat het U2F protocol alleen kan werken op een certificaten gebaseerde infrastructuur. Uit het onderzoek is het U2F-protocol als het meest aantrekkelijke MFA gekomen zijn er twee systeemconcepten met de combinatie U2F en PKI ontworpen. Klassiek Bluetooth heeft sinds Bluetooth 4.1 de Secure upgrade gekregen waardoor er ook een sterke beveiliging in het protocol klassiek bluetooth zit.

7.1 Systeemconcept 1: Certificaten PKI & U2F - Stick

In het eerste systeemconcept wordt er een Certificaten PKI-omgeving opgezet waarop de proxy-dienst van de RPI3 wordt uitgevoerd. De data die verzonden wordt naar de publieke dienst wordt met AES versleuteld. De gebruiker authenticeert zich met een gebruikersnaam, wachtwoord en een u2f-stick die een NFC-chip bevat.

In Figuur 7.1 wordt een ontwerp van het systeemconcept weergegeven.



Figuur 7.1 Systeemconcept 1: Certificaten PKI & U2F-stick/NFC

Hieronder wordt een lijst van voor- en nadelen weergegeven bij het gebruik van het systeemconcept bij dit project.

Voordelen:

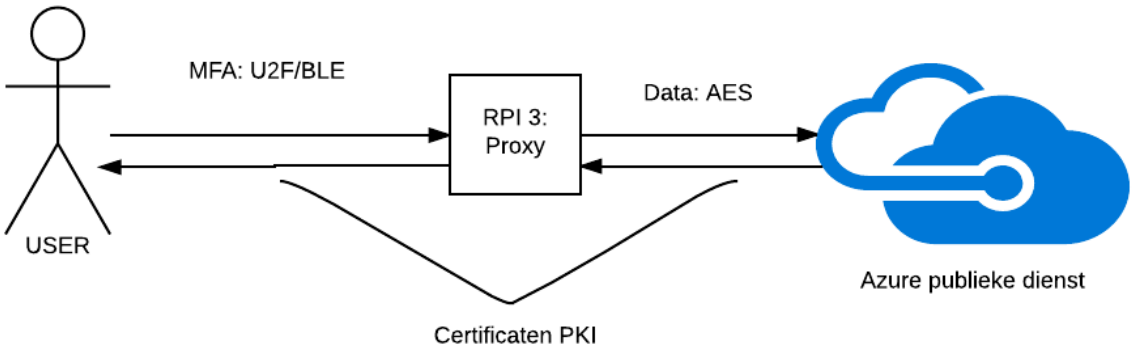
- Korte afstand NFC
- Veiligheid van U2F

Nadelen:

- Fysiek apparaat dat makkelijk kan worden vergeten in de RPI

7.2 Systeemconcept 2: Certificaten PKI & U2F/BLE

Het tweede concept wordt er een certificaten PKI gebruikt met een combinatie van BLE en U2F. BLE verreist weinig energie, maar is zo uitgekleed dat het een onveilig protocol is om te gebruiken als een MFA. Door het toevoegen het U2F-protocol wordt er wel een sterke authenticatie toegevoegd.



Figuur 7.2 Systeemconcept 2

Hieronder wordt een lijst van voor- en nadelen weergegeven bij het gebruik van het systeemconcept bij dit project.

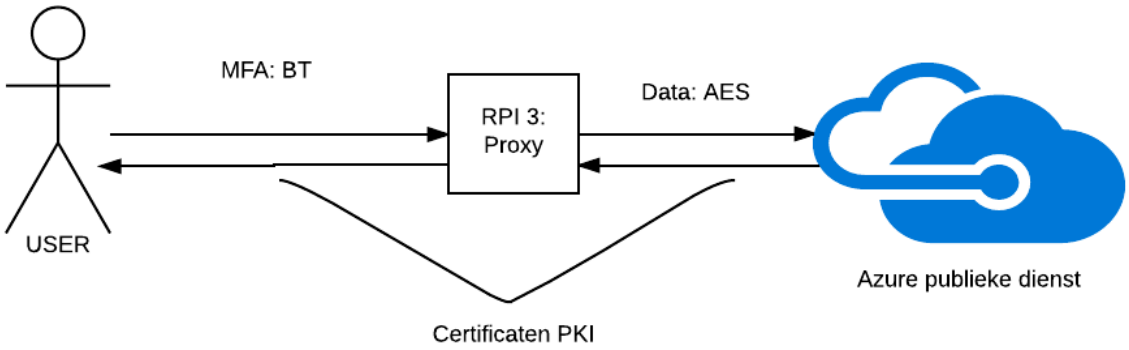
Voordelen:

- Redelijk korte afstand. (Kan worden aangepast)
- De gebruiker heeft niet de mogelijkheid de sleutel te laten liggen bij het apparaat. (Bijv. usb-sleutel in de RPI3 laten)
- Een sterke MFA, maar de Usability blijft goed te gebruiken.
- Een telefoon is lastiger te klonen, stelen of vergeten.
- Low- energy, het vergt weinig stroom van authenticatie apparaat.

Nadelen:

- Er bestaat nog geen werkend product van op de markt

7.3 Systeemconcept 3: Certificaten PKI & Klassieke Bluetooth



Figuur 7.3 Systeemconcept 3

Hieronder wordt een lijst van voor- en nadelen weergegeven bij het gebruik van het systeemconcept bij dit project.

Voordelen:

- Het protocol is op zichzelf een veilig protocol, er hoeven geen ingewikkelde methodes/protocollen worden toegevoegd.
- Usability – het protocol is makkelijk te gebruiken en veel hedendaagse apparaten hebben het.

Nadelen:

- Het vergt veel energie van het authenticatie apparaat.

8 Requirements informatiesysteem

De requirements worden vastgesteld aan de hand van het onderzoek naar het probleemdomein en de gesprekken die gevoerd zijn met de opdrachtgever en begeleider.

Om onderscheid te kunnen maken tussen de wensen en de systeemeisen wordt er een prioriteit aan de requirements toegekend. De eisen worden aan de hand van de MoSCoW-methode geprioriseerd. Er is voor de MoSCoW – methode gekozen, omdat de student ervaring met deze methode heeft. De MoSCoW-methode geeft een goed overzicht van de meest en minder belangrijke eisen.

De letters van MoSCoW staan voor:

- M → Must Have: Moet voorkomen in het systeem
- S → Should Have: Hoge prioriteit, maar zonder is het systeem nog bruikbaar
- C → Could Have: Alleen als er nog voldoende tijd is
- W → Would Have: Geen prioriteit, eventueel in een vervolgend project

Categorisatie wordt gedaan volgens het FURPS+ model. Het FURPS+ model geeft een inzicht in de niet – functionele eisen. Het model is ontwikkeld bij Hewlett-Packard en de letters staan voor de volgende categorieën:

- F → Functionality – Funtioneel
- U → Usability – Bruikbaarheid
- R → Reliability – Betrouwbaarheid
- P → Performance – Efficiëntie
- S → Supportability – Onderhoudbaarheid

De “+” staat voor een aantal extra wensen die klant zou kunnen hebben.

- Ontwerp – Hoe het systeem wordt opgebouwd
- Implementatie – Wat is er nodig voor invoering van het systeem
- Fysiek – Wat voor hardware is er nodig om het systeem te kunnen uitvoeren

In de volgende paragrafen worden de functionele en niet-functionele eisen weergegeven in tabellen. De functionele en niet-functionele eisen zijn opgesteld aan de hand van interviews met de opdrachtgever en de begeleider.

8.1 Functionele eisen

In **Fout! Verwijzingsbron niet gevonden.** worden de eisen van de Proof of concept weergegeven. De eisen b eschrijven de diensten die het systeem aan de gebruiker levert.[32]

ID	Omschrijving	Prioriteit (MoSCoW)	Commentaar
REQ01	Het POC bestaat uit een Publieke dienst	M	
REQ02	Het POC heeft een Raspberry Pi die data versleutelt en ontsleutelt op de publieke dienst	M	Dit wordt op een RPI3 ontwikkeld en wordt de "Security-Pi" genoemd
REQ03	Het POC heeft een MFA	M	
REQ04	Er mogen geen aanpassingen worden gedaan aan de publieke dienst	M	
REQ05	Op de publieke dienst kan worden ingelogd met een gebruikersnaam en wachtwoord	M	
REQ06	De publieke dienst kan berichten plaatsen	M	
REQ07	De publieke dienst kan geplaatste berichten weergeven	M	
REQ08	De publieke dienst kan nieuwe gebruikers registreren	M	
REQ09	De Security-Pi gebruikt een AES-256 encryptie voor data	M	Data die opgeslagen wordt op publieke dienst
REQ10	De Security-Pi kan de versleutelde data ontsleutelen en weergeven	M	
REQ11	Op de Security-Pi kan worden ingelogd met gebruikersnaam en wachtwoord	M	
REQ12	Op de Security-Pi kan worden ingelogd met het MFA-apparaat	S	
REQ13	Op de Security-Pi kan een MFA-apparaat worden geregistreerd	S	
REQ14	De MFA heeft een krachtige bescherming tegen phishing-aanvallen	S	
REQ15	De MFA is een fysiek apparaat	S	USB of Smartphone
REQ16	De MFA kan niet makkelijk worden vergeten	S	"Een USB vergeet je makkelijker dan een Smartphone"
REQ17	De MFA heeft een laag energieverbruik	S	
REQ18	De MFA kan biometrisch identificeren	C	

Tabel 8.1 proof of concept eisen

[32]http://portal.ou.nl/documents/informatica/snapshots/T07341_01.pdf

9 Bepalen definitieve systeemarchitectuur

Dit hoofdstuk wordt er een definitieve systeemarchitectuur geselecteerd. Er zal een keuze worden gemaakt uit de mogelijke systeemconcepten met de eisen van de proof of concept. Uit de afweging wordt er een besluit genomen welk van de mogelijke systeemconcepten zal worden ontwikkeld.

9.1 Afweging van systeemconcepten

In **Fout! Verwijzingsbron niet gevonden.** wordt een afweging gedaan van de systeemconcepten. Er wordt per eis b ekeken in hoeverre deze voldoet aan de eis. Een "+" geeft een onvoldoende aan, een "++" geeft een voldoende aan en een "+++" geeft aan dat het goed is.

Requirements / Opstelling	1: Certificaten PKI & U2F – Stick	2: Certificaten PKI & U2F/BLE - Smartphone	3: Certificaten PKI & Klassieke Bluetooth - Smartphone
REQ13	+++	+++	++
REQ14	+++	++	++
REQ15	+	++	++
REQ16	+++	++	+
REQ17	+	+++	+++
Totaal +	**11**	**12**	**10**

Tabel 9.1 Afweging van mogelijke opstellingen | + = Onvoldoende ++ = Voldoende +++ = Goed

Uit de afweging is opstelling 2 als het meest geschikt voor het proof of concept. Van dit product bestaan niet veel voorbeelden en is nog in ontwikkeling bij een bedrijven[33]. De MFA in opstelling 2 is een samenvoeging het U2F protocol in combinatie met BLE. De Raspberry-Pi 3 ondersteunt Bluetooth 4.1 wat niet zo veilig bleek te zijn. Door het toevoegen van het U2F-protocol wordt het juist een sterk MFA.

Om dit werkend te krijgen is besloten het product op te bouwen in delen. Als eerste wordt het BLE-gedeelte ontwikkeld. Vervolgens wordt het U2F gedeelte ontwikkeld en als laatst worden deze twee samengevoegd tot een MFA voor het Security-Pi systeem.

[33] Yubico, BLE/U2F -- https://www.yubico.com/2016/06/yubikey-u2f-tracking-bluetooth-maturity/

10 Conclusie Definitiefase

Aan de hand van de informatie die is vergaard tijdens het onderzoek kan er een antwoord worden gegeven op de deelvragen. Aan de hand van de antwoorden op de deelvragen wordt er een conclusie getrokken voor de definitiefase.

Deelvraag: Versleutelde data onderscheiden

Aan het begin van het project is door de opdrachtgever aangegeven op een publieke dienst een extra veiligheid toe te voegen. Dit is al gauw uit de scope van het project vervallen. Het dataverkeer van Facebook is versleuteld en kan niet makkelijk worden uitgelezen. Er zijn een aantal manieren geprobeerd om alsnog verkeersdata te decoderen. Dit was deels mogelijk, maar niet genoeg om duidelijk de werking van de dienst te zien. Al snel is er met de opdrachtgever gezocht naar een alternatief. Er is besloten een eigen service te ontwikkelen en daaraan een extra veiligheid aan toe te voegen.

Deelvraag: Bepalen van definitieve cryptografie

In deze definitiefase zijn verschillende cryptografie-methodes onderzocht. Als resultaat blijkt AES-encryptie meest optimale symmetrische encryptiemethode en ECC het sterkst als het gaat om een asymmetrische encryptie. AES zal worden gebruikt voor encryptie van netwerkverkeer, omdat symmetrische encryptie veel sneller werkt vergeleken met asymmetrische encryptie. Om de sleutel van AES veilig te vervoeren wordt er een asymmetrische infrastructuur opgezet met een certificaten PKI dat ECC gebruikt om de certificaten de encrypten.

Deelvraag: Bepalen definitieve multi-factor authenticatie

Na het onderzoeken van verschillende manieren om een MFA op te zetten is als resultaat het U2F-protocol gekomen. Het U2F – protocol biedt bescherming tegen phishing-aanvallen en heeft een privacybescherming. Het U2F -protocol wordt het meest toegepast op USB-stick of NFC-chip. Deze fysieke objecten kunnen makkelijk vergeten, gestolen of gekloond worden. In een overleg met de opdrachtgever werd er toen gekeken naar het gebruik van de smartphone als het fysieke object in combinatie met bluetooth. Het Bluetooth -protocol kost de smartphone veel energie vergeleken met het nieuwe BLE-protocol van Bluetooth. BLE bleek onveilig te zijn om te gebruiken als authenticatiemethode. Er is toen besloten een combinatie te maken van BLE en het U2F protocol.

Deelvraag: Bepalen definitieve systeemarchitectuur

Om een definitieve systeemarchitectuur te bepalen zijn een aantal systeemconcepten ontworpen. Elk systeemconcept bestaat uit de resultaten uit eerder onderzoek naar infrastructuur, cryptografie en MFA. Elk concept heeft zijn voor- en nadelen die zijn aangegeven bij de opdrachtgever. De eisen van de opdrachtgever zijn toen vastgelegd en op basis van de eisen kon er een afweging worden gemaakt voor een definitieve systeemarchitectuur. Als resultaat is de Certificaten PKI ECC & BLE/U2F geselecteerd.

Hoofdvraag: “Wat is de meest optimale extra beveiliging die met behulp van een Raspberry Pi 3 aan applicaties van klanten toegevoegd kan worden zonder de applicaties aan te passen?”

Uit het onderzoek van de definitiefase lijkt een certificaten PKI ECC & BLE/U2F & AES-omgeving het meest geschikt. Het systeem wordt verder ontwikkeld en aan het eind van het project wordt er opnieuw gekeken naar enige veranderingen in de hoofd- en deelvragen.

Hoofdvraag: “Op welke manier zou dit project bestaande cybersecurityoplossingen kunnen vervangen?”

Het project is bedacht door de opdrachtgever met als gedachten dat het niet mogelijk bij de klant een VPN-tunnel te gebruiken. Het zou niet het gebruik van VPN-tunnels kunnen vervangen, wel kan het worden toegepast als het niet mogelijk is om de applicatie-omgeving van de klant aan te passen en er geen mogelijkheid is voor een VPN-verbinding.

11 Bronnenlijst

André Clerc - Temet. (2017, February 9). *About & Beyond PKI - Blockchain and PKI.* Opgehaald van http://www.sig-switzerland.ch: http://www.sig-switzerland.ch/wp-content/uploads/2015/06/SIGS_Feb2017_TEMET_Does_Blockchain_secure_PKIs_in_the_long-term.pdf

Atmel. (2015). *RSA vs ECC Comparison for Embedded Systems.*

Bluetooth SIG. (2016). *Bluetooth Core Specification 5.0 FAQ.* SIG.

Bluetooth SIG. (2017). *Bluetooth Core Specification v5.0.* Bleutooth SIG.

Bluetooth SIG. (sd). *Bluetooth low energy security.*

Bluetooth SIG Security Expert Group. (2002, April 19). *Bluetooth Securtiy White Paper.* Retrieved from http://grouper.ieee.org: http://grouper.ieee.org/groups/1451/5/Comparison%20of%20PHY/Bluetooth_24Security_Paper.pdf

Cobb, C. (2004). *Cryptography For Dummies.* John Wiley & Sons.

Dams, J. (2012, october 12). *An introduction to elliptic curve cryptography*. Opgehaald van http://www.embedded.com: http://www.embedded.com/design/safety-and-security/4396040/An-Introduction-to-Elliptic-Curve-Cryptography

Developers, S. B. (Regisseur). (2015). *SF Bitcoin Devs Seminar: RevokeSSL: An Independent Revocation Service using the Bitcoin Blockchain* [Film].

DLP, D. (2016, june 16). *Bluetooth 5: 2x the speed + 4x the range = 8x the security risk*. Opgehaald van http://www.devicelock.com/: http://www.devicelock.com/blog/2842.html

Earle, J. (2015, May 16). *Elliptic Curve Cryptography & Diffie-Hellman.* Opgehaald van http://www.csbreakdown.com/: https://www.youtube.com/embed/yDXiDOJgxmg?rel=0

Facebook. (2013, July 31). *Secure browsing by default*. Opgehaald van https://www.facebook.com: https://www.facebook.com/notes/facebook-engineering/secure-browsing-by-default/10151590414803920/

Giry, D. (2017, February 23). *https://www.keylength.com*. Opgehaald van BlueKrypt: https://www.keylength.com/en/compare/

Knafo, J. (2016, October 21). *Most Popular 2-Factor Authentication (2FA) Compared*. Opgehaald van https://blog.devolutions.net: https://blog.devolutions.net/2016/10/most-popular-2-factor-authentication-2fa-compared.html

London, S. B.-S. (Regisseur). (2015). *Martijn Grooten - Elliptic Curve Cryptography for those who are afraid of maths* [Film].

N, R. (2017). *U 2 can U2F.* Opgehaald van https://robn.io/: https://robn.io/talks/u2f-lca-2017/U2F-notes.pdf

National Institute of Standards and Technology. (2012, June). *Guide to Bluetooth Security.* Opgehaald van http://nvlpubs.nist.gov: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-121r1.pdf

National Science and Technology Council (NSTC). (2013). *Fingerprint Recognition.* FBI.

Open Universiteit Nederland. (sd). *OU.* Opgehaald van Portal.ou.nl: http://portal.ou.nl/documents/informatica/snapshots/T07341_01.pdf

Robles, P. (2015, March 17). *Can the blockchain replace SSL?* Opgehaald van https://www.programmableweb.com/: https://www.programmableweb.com/news/can-blockchain-replace-ssl/analysis/2015/03/17

Schneider, P. F. (2005). *Something You Know, Have, or Are*. Opgehaald van https://www.cs.cornell.edu: https://www.cs.cornell.edu/courses/cs513/2005fa/NNLauthPeople.html

Shaver, j. (2015, February 11). *Decrypting TLS Browser Traffic With Wireshark – The Easy Way!* Opgehaald van https://jimshaver.net/: https://jimshaver.net/2015/02/11/decrypting-tls-browser-traffic-with-wireshark-the-easy-way/

Tel, G. (2006). *Cryptografie - Beveiliging van de digitale maatschappij.* Universiteit Utrecht.

Tripura University, I. (2011, July 27). *https://www.intechopen.com.* Opgehaald van Thermal Infrared Face Recognition – a Biometric Identification Technique for Robust Security System: https://www.intechopen.com/books/statistics/reviews-refinements-and-new-ideas-in-face-recognition/thermal-infrared-face-recognition-a-biometric-identification-technique-for-robust-security-system

WhatsApp. (2016). *WhatsApp Encryption Overview.*

ICT Group N.V.

HET ONTWIKKELEN VAN EEN EXTRA BEVEILIGING VOOR NETWERKCOMMUNICATIE VAN KLANTEN NAAR DE CLOUD VAN ICT GROUP

Ontwikkelrapport

Technische Informatica
De Haagse Hogeschool
Versie 0.2

Inhoudsopgave

1 Inleiding

Dit rapport is het product uit de ontwikkelfase. In de definitiefase zijn de incrementen bepaald die in de ontwikkelfase worden uitgevoerd. Het ontwikkelen van het proof of concept wordt gedaan in incrementen. Elke increment kan worden verdeeld in activiteiten. De activiteiten bestaat uit een gedeelte ontwerpen, realisatie en testen. In het ontwerp activiteit wordt behulp van diagrammen het ontwerp van het systeem weergegeven. Met deze ontwerpen wordt de werking van het systeem duidelijk en kan er code worden geïmplementeerd. Dit gedeelte van een increment wordt de realisatie genoemd. Als laatste wordt de implementatie getest. Met behulp van de diagrammen uit het ontwerp gedeelte worden testscenario's opgesteld om de werking van het systeem te testen.

In de definitiefase zijn increment bepaald die in deze ontwikkelfase worden uitgevoerd. In Tabel 1.1 worden deze incrementen weergegeven.

Increment	beschrijving	Activiteiten
Increment 1: Publieke dienst opzetten	In deze increment wordt een publieke dienst opgezet, zodat de werking van het proof of concept kan worden weergegeven.	- Ontwerpen omgeving - Realiseren omgeving - Testen van realisatie omgeving
Increment 2: Ontwikkelen Security-Pi	Er wordt een security-pi ontwikkeld dat data van de gebruiker kan versleutelen en ontsleutelen op de publieke dienst.	- Ontwerpen werking van Security- Pi - Realisatie van de Security-Pi - Testen Security - Pi
Increment 3: Opzetten MFA	Deze increment wordt het ontwikkelen van de MFA dat uit het onderzoek als meest geschikte is gekomen.	- Ontwerpen werking MFA - Realisatie van ontwerp MFA - Testen realisatie MFA

Tabel 1.1 Overzicht van incrementen

2 Increment 1: Publieke dienst opzetten

De opdrachtgever wilt dat het systeem werkt op een publieke dienst zoals Facebook of Twitter. De publieke dienst moet het systeem bij de klant voorstellen waar verder niks aan aangepast mag worden. Het was al snel duidelijk dat de dienst Facebook of Twitter niet een geschikt testomgeving was en is met de opdrachtgever besproken dat er zelf een dienst wordt opgezet dat systeem bij de klant voorstelt.

2.1 Ontwerp Publieke dienst

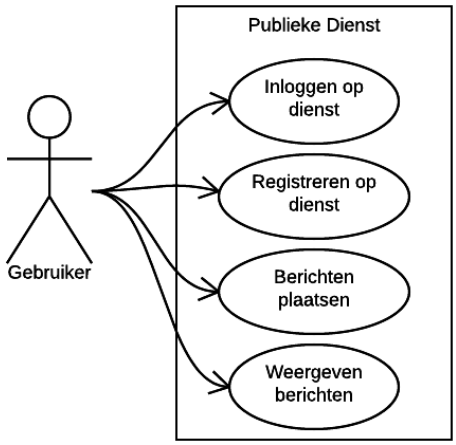
In deze activiteit worden ontwerpen gemaakt van de werking van de publieke dienst. Het ontwerp beschrijft de werking van het systeem. Het ontwerp wordt gemaakt met behulp van UML. UML staat voor Unified Model Language en is een manier voor het visualiseren van de werking van het proof of concept. Om een ontwerp te kunnen maken zijn de eisen genomen die betrekking hebben tot de publieke dienst. Deze zijn eerder vastgesteld in de definitiefase en staan beschreven in het definitierapport. In Tabel 7.1 worden de eisen beschreven die gaan over hoe de publieke dienst is opgebouwd. Deze eisen worden omgezet naar een ontwerp van de publieke dienst.

ID	Omschrijving
REQ05	De publieke dienst kan worden ingelogd met een gebruikersnaam en wachtwoord
REQ06	De publieke dienst kan berichten plaatsen
REQ07	De publieke dienst kan geplaatste berichten weergeven
REQ08	De publieke dienst kan nieuwe gebruikers registreren

Tabel 2.1 requirements publieke dienst

Use-Case Diagram Publieke Dienst:

Use-Case diagrammen geven globaal de werking van een systeem aan. In elke Case staat een activiteit van het systeem beschreven. In Figuur 7.2**Fout! Verwijzingsbron niet gevonden.** wordt er een Use-Case diagram w eergegeven dat de mogelijke activiteiten van de publieke dienst weergeeft. De gebruiker kan zich laten registreren of inloggen op de dienst en een gebruiker kan berichten plaatsen en geplaatste berichten laten weergeven.



Figuur 2.1 Use-Case Diagram - Publieke Dienst

Scenario's Publieke Dienst

De volgende scenario's beschrijven de werking van de publieke dienst. Een scenario heeft een doel dat aangeeft wat er met een scenario bereikt wordt. Een pre-conditie dat aangeeft waar er van tevoren aan is

voldoen om het doel te kunnen bereiken. De activiteiten die uitgevoerd worden om het doel te bereiken. Als laatst de alternatieven die aangeeft wat er niet zou kunnen gaan, waardoor het doel niet wordt bereikt.

Scenario IC1-SC1 beschrijft het inloggen van een gebruiker op het publieke systeem. Scenario IC1-SC2 beschrijft het registreren van een gebruiker op de publieke dienst. Een bericht plaatsen op het publieke systeem wordt beschrijven bij scenario IC1-SC3.Als laatst wordt in IC1-SC4 het weergeven van het bericht beschreven.

IC1 – SC1	Inloggen gebruiker
Doel	Het publieke systeem logt de gebruiker in
Pre-Condities	De gebruiker is geregistreerd
Activiteiten	7. De gebruiker voert inloggegevens in 8. Het systeem controleert de gegevens 9. Het systeem logt de gebruiker in
Alternatieven	5. De ingevoerde gegevens van de gebruiker kloppen niet 6. De gebruiker wordt terug gestuurd naar het inlogscherm

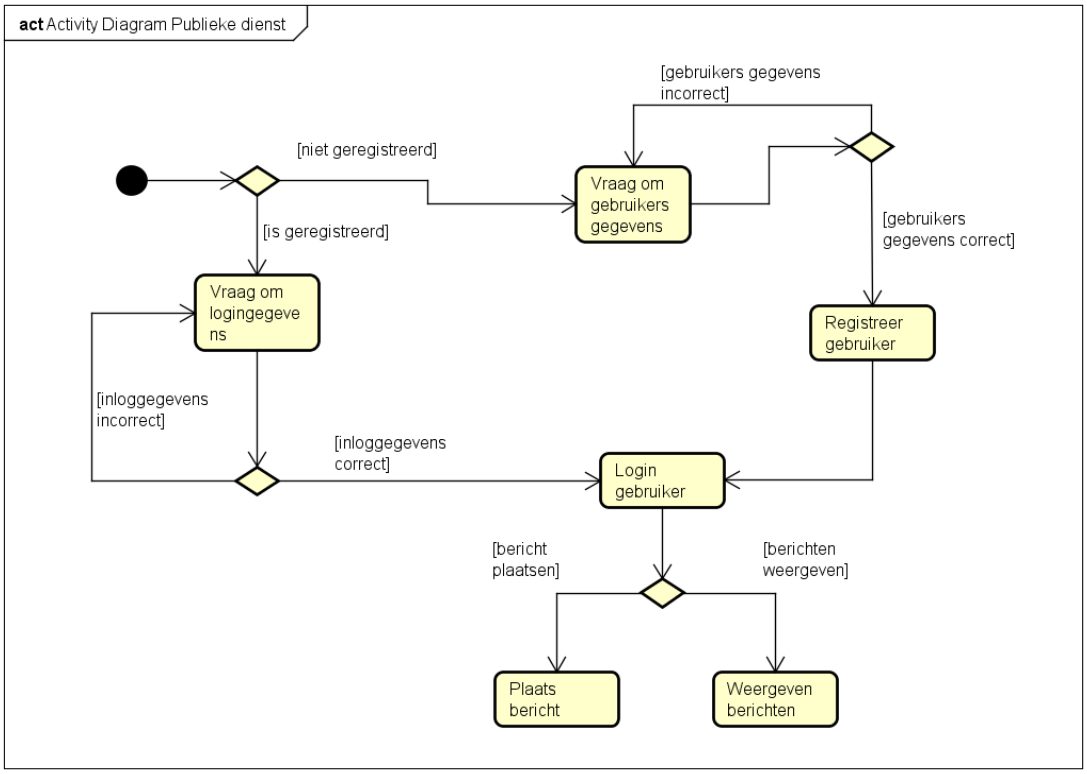
IC1 – SC2	Registreer gebruiker
Doel	Het publieke systeem registreert de gebruiker
Pre-Condities	De gebruiker niet geregistreerd
Activiteiten	1. De gebruiker voert registratiegegevens in 2. Het systeem controleert de registratiegegevens 3. Het systeem registreert de gebruiker
Alternatieven	1. De ingevoerde registratiegegevens van de gebruiker voldoet niet aan de eisen van het systeem 2. Er wordt met een melding aan de gebruiker duidelijk gemaakt welke van de inloggegevens niet kloppen

IC1 – SC3	Bericht plaatsen
Doel	De gebruiker plaatst een bericht op het publieke systeem
Pre-Condities	De gebruiker is ingelogd
Activiteiten	7. De gebruiker voert een bericht in 8. De gebruiker geeft aan dat het bericht gedeeld mag worden 9. Het systeem bewaard het bericht in een database
Alternatieven	Geen

IC1 – SC4	Weergeven bericht
Doel	Het publieke systeem weergeeft de geplaatste berichten
Pre-Condities	Geen
Activiteiten	1. Het systeem geeft alle opgeslagen berichten weer op een tijdlijn
Alternatieven	Geen

Activiteitsdiagram Publieke Dienst

Met een activiteitsdiagram worden de mogelijke activiteiten van het systeem weergegeven. Door een proces van keuzes kan een systeem in zo een toestand terecht komen. Met behulp van de scenario's die eerder zijn opgezet is er een activiteitsdiagram gemaakt. In Figuur 7.3 wordt het activiteitsdiagram weergegeven van de publieke dienst. Als de gebruiker het systeem opstart kan hij een keuze maken om in te in te loggen met inloggegevens of zich laten registreren als gebruiker. Beide routes leiden tot een ingelogde gebruiker. Waarna de gebruiker kan kiezen om een bericht te plaatsen of het weergeven van berichten.

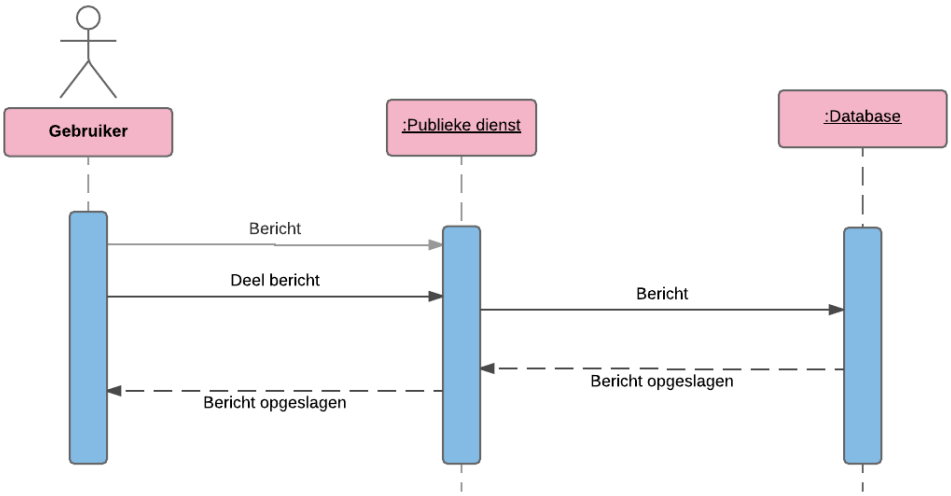


Figuur 2.2 Activiteitsdiagram - Publieke dienst

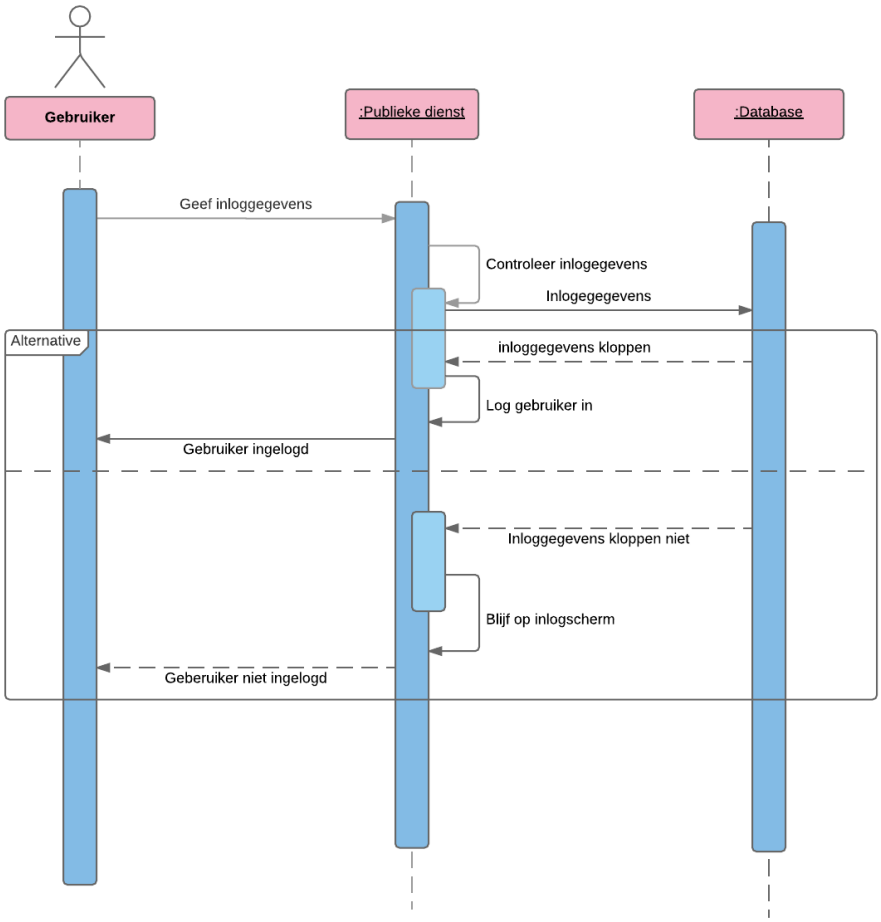
Sequentiediagrammen Publieke Dienst

Er wordt in elke iteratie mogelijke scenario's beschreven die de werking van het systeem beschrijft. Van een aantal van deze scenario's worden sequentie diagrammen gemaakt om duidelijk beeld weer te geven van de scenario's.

In Figuur 2.3 wordt een sequentie diagram van scenario IC1-SC3 weergegeven. In het diagram wordt laten zien hoe een gebruiker een bericht kan delen op de publieke dienst. In Figuur 2.4 wordt een sequentiediagram weergegeven van IC1-SC1 dat de werking van het systeem beschrijft als een gebruiker wilt inloggen.



Figuur 2.3 Sequentie Diagram - Publieke Dienst – IC1-SC3 Bericht plaatsen



Figuur 2.4 Sequentie Diagram - Publieke dienst – IC1-SC1 Inloggen gebruiker

2.2 Realisatie publieke dienst

Voordat er zelf een systeem wordt ontwikkeld is er gekeken naar mogelijke open-source dat voldoet aan het ontwerp dat is gemaakt de vorige paragraaf. Op het internet zijn er een aantal mogelijke systemen gevonden dat voldoet aan deze voorwaarden. De volgende punten zijn opgezet bij het zoeken van een mogelijk publieke dienst:

- Registreren
- Inloggen
- Berichten plaatsen
- Berichten weergeven

De volgende mogelijke systemen zijn gevonden en voldoen aan de bovenstaande eisen voor het selecteren van een publieke dienst.

Dienst	Beschrijving
Minitwit	Een mini Twitter kloon ontwikkelt in Python Flask
Firefeed	Een open-source Twitter kloon ontwikkelt in Firebase
Scaffeinate	Een social-network platform ontwikkelt in Ruby on Rails

Figuur 2.5 Lijst van mogelijke publieke dienst

Minitwit is gekozen als meest geschikt voor het project. Het Minitwit project is ontwikkeld in Python, de taal waar de student meer ervaring in heeft. Er zijn ook veel voorbeelden gevonden van het gebruik van Minitwit waar bij de andere gevonden applicaties minder was.

Werking Minitwit

In Kader 2.1 wordt in Python code weergegeven hoe het inloggen van een gebruiker wordt gedaan. Er wordt een sessie opgezet voor de gebruiker. De ingevoerde gebruikersnaam en een hash van het wachtwoord worden beiden gecontroleerd in de database met de opgeslagen gebruikersnaam en wachtwoord hash. Bij Flask wordt er gevraagd een @application.route(webpagina, methode) te plaatsen voor een methode. Dit geeft aan dat als de webpagina wordt geopend de methode die er onder staat zal worden uitgevoerd.

```
@application.route('/login', methods=['GET', 'POST'])
def login():
    """Logs the user in."""
    if g.user:
        return redirect(url_for('timeline'))
    error = None
    if request.method == 'POST':
        user = query_db("""SELECT * FROM user WHERE
            username = ?""", [request.form['username']], one=True)
        print user,"dit is de user"
        if user is None:
            error = 'Invalid username'
        elif not check_password_hash(user['pw_hash'],
                                     request.form['password']):
            error = 'Invalid password'
        else:
            try:
                session['u2f_user_id'] = user['user_id']

                flash('you where logged in’)
                session['user_id'] = user['user_id']
                return redirect(url_for('timeline'))

            except exc.NoEligableDevicesException as e:
                error = e.message
    return render_template('login.html', error=error)
```

Kader 2.1 Minitwit inloggen gebruiker – Minitwit.py

In Kader 2.2 wordt er in python weergegeven hoe het registreren van een gebruiker wordt gedaan. Ook hier wordt er een @application.route toegevoegd, zodat als de gebruiker op de pagina “register” terecht komt dan wordt “def register():” uitgevoerd. Er wordt een formulier aangeboden voor het invullen van een gebruikersnaam, email en een wachtwoord.

```
@application.route('/register', methods=['GET', 'POST'])
def register():
    """Registers the user."""
        return redirect(url_for('timeline'))
    error = None
    if request.method == 'POST':
        if not request.form['username']:
            error = 'You have to enter a username'
        elif not request.form['email'] or \
                '@' not in request.form['email']:
            error = 'You have to enter a valid email address'
        elif not request.form['password']:
            error = 'You have to enter a password'
        elif request.form['password'] != request.form['password2']:
            error = 'The two passwords do not match'
        elif get_user_id(request.form['username']) is not None:
            error = 'The username is already taken'
        else:
            db = get_db()
            db.execute('''insert into user (
              username, email, pw_hash) values (?, ?, ?)''',
              [request.form['username'], request.form['email'],
               generate_password_hash(request.form['password'])])
            db.commit()
            flash('You were successfully registered and can login now')
            return redirect(url_for('login'))
    return render_template('register.html', error=error)
```

Kader 2.2 Minitwit Registeren

In Kader 2.3 wordt in Python code weergegeven hoe het plaatsen van een bericht wordt gerealiseerd. Het bericht bevat de ID van een gebruiker en het bericht.

```
@application.route('/add_message', methods=['POST'])
def add_message():
    """Registers a new message for the user."""
    if 'user_id' not in session:
        abort(401)
    if request.form['text']:
        db = get_db()
        db.execute('''insert into message (author_id, text, pub_date)
          values (?, ?, ?)''', (session['user_id'], request.form['text'],
                                int(time.time())))
        db.commit()
        flash('Your message was recorded')
    return redirect(url_for('timeline'))
```

Kader 2.3 Minitwit - Berichten plaatsen

Kader 2.4 geeft aan hoe het weergeven van berichten wordt uitgevoerd in Python code. De database met alle opgeslagen berichten worden opgehaald

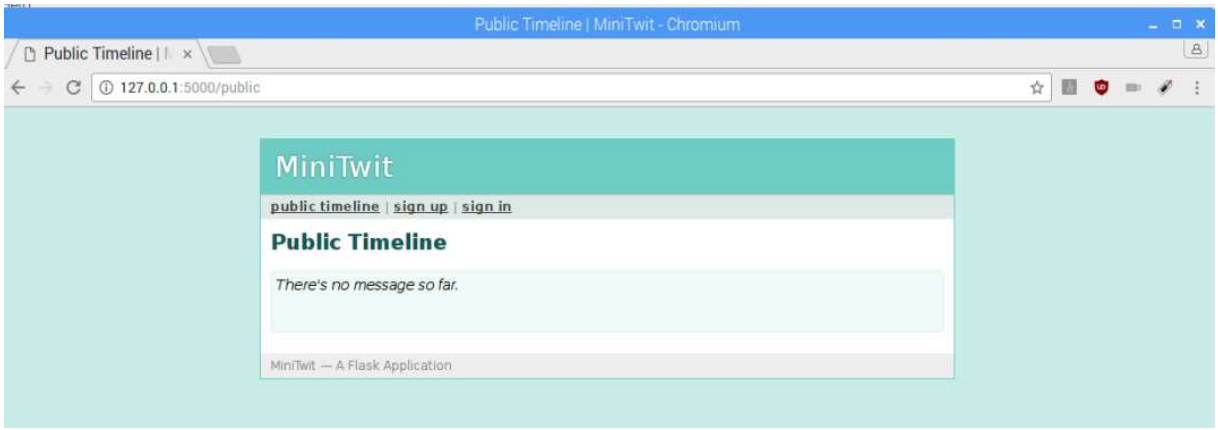
```
@application.route('/public')
def public_timeline():
    """Displays the latest messages of all users."""
    return render_template('timeline.html', messages=query_db('''
        select message.*, user.* from message, user
        where message.author_id = user.user_id
        order by message.pub_date desc limit ?''', [PER_PAGE]))
```

Kader 2.4 Minitwit weergeven berichten

Mintwit benaderbaar maken

Eerst is Minitwit lokaal geïnstalleerd op een virtuele machine. Hiervoor is Vmware Workstation 12 Pro gebruikt, deze VM-programma stond al geïnstalleerd op de student zijn werk-pc en de student heeft hier het meeste ervaring mee. Er is als operating systeem gebruik gemaakt van Ubuntu server 16.04, deze OS was de meest recente OS van Ubuntu op het moment. De Ubuntu distributie heeft de student ook het meeste ervaring mee, daarom is deze gekozen als OS.

Om de Flask applicatie te laten werken is als eerst package Flask nodig. Deze is gemakkelijk te installeren met Python PIP. Als de Flask package is geïnstalleerd kan Minitwit werken op Ubuntu server. Met Python kan de applicatie worden gestart. Flask heeft start dan een ingebouwde server en laat daarop de applicatie werken. De applicatie kan dan alleen benaderd worden op de Ubuntu server via localhost:5000. Poort 5000 wordt door Flask standaard gebruikt als debug poort. In Figuur 2.6 de publieke dienst Minitwit weergegeven dat werkt op poort 5000.



Figuur 2.6 Mintwit werkend lokaal op poort 5000

Nginx zal worden gebruikt om de front-end van de publieke dienst op de te hosten en Gunicorn wordt gebruikt om de back-end van de applicatie draaiend te houden. Nginx kan niet beiden, omdat het niet kan communiceren met met WSGI Flask applicaties. Als eerste wordt nginx en gunicorn geinstalleerd met de volgende commando:

```
$ sudo apt-get install -y nginx gunicorn
```

Ten tweede wordt de standaard website van Nginx verwijderd. En een configureren de mappen, zodat deze dezelfde naam hebben als de map waarin de applicatie staat. Hieronder volgen de commando's die moeten worden uitgevoerd

```
$ sudo rm /etc/nginx/sites-enabled/default
$ sudo touch /etc/nginx/sites-available/Minitwit
$ sudo ln -s /etc/nginx/sites-available/Minitwit /etc/nginx/sites-enabled/Minitwit
```

Hierna wordt de volgende configuratie toegevoegd, zodat Nginx de http verzoeken als een reverse proxy naar de localhost waarop applicatie werkt.

```
server {
   location / {
      proxy_pass http://localhost:8000;
      proxy_set_header Host $host;
      proxy_set_header X-Real-IP $remote_addr;
   }
   location /static {
      alias  /home/www/minitwit/static/;
   }
}
```

Dan moet alleen de applicatie nog worden opgestart. De applicatie wordt opgestart met behulp van Gunicorn. Gunicorn zorgt ervoor dat als de terminal gesloten wordt de applicatie blijft draaien op de achtergrond. De volgende commando zorgt ervoor dat Gunicorn de applicatie opstart. Het is hierbij belangrijk dat de naam van het bestand wordt opgegeven en de naam van de applicatie in het bestand zelf. In dit geval "minitwit:application".

```
$sudo gunicorn minitwit:application -b 0.0.0.0:8000 --reload
```

De volgende commando zorgt ervoor dat de applicatie stopt met werken.

```
Sudo pkill gunicorn
```

2.3 Testen publieke dienst

In deze paragraaf worden de testen beschreven die gaan over de publieke dienst. Voor het testen van de publieke dienst zijn de scenario's genomen die zijn opgesteld in de ontwerpactiviteit. Deze scenario's zijn omgezet tot testcases. Elke testcase beschrijft een doel, een pre-conditie, de activiteiten, de alternatieven, het verwachte resultaat, het daadwerkelijke resultaat en of de testcase geslaagd is. De uitgevoerde testen van dit increment worden beschreven in het Testrapport hoofdstuk 3.1 increment 1 Publieke Dienst.

3 Increment 2: Ontwikkelen Security-Pi

Dit hoofdstuk wordt het tweede increment beschreven van dit project. Het tweede increment is het ontwikkelen van de Security-Pi. Als eerste wordt er een ontwerp beschreven, waarin met behulp van UML diagrammen zijn gemaakt. Daarna zal de realisatie van de Security-Pi worden beschreven. De realisatie wordt verricht met behulp van het ontwerp van de Security-Pi. Ten derde zal de realisatie worden getest.

3.1 Ontwerp Security- Pi

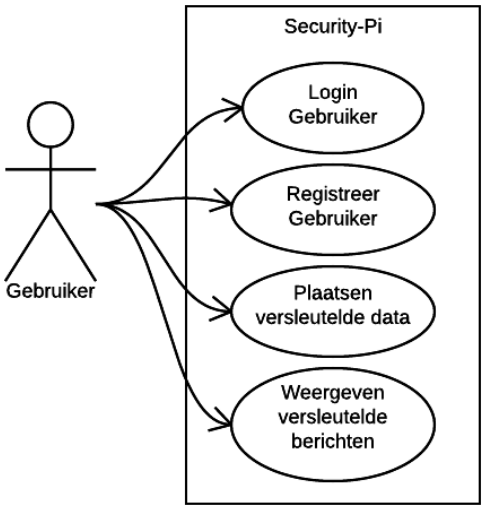
Om een ontwerp te maken van de Security-Pi zijn de requirements die in de vorige fase zijn vastgesteld erbij gehaald. In tabel worden deze eisen beschreven en daaruit is er een Use-case diagram gemaakt. Vanuit de Use-Case zijn er mogelijke scenario's opgesteld die bij de Security-Pi kunnen voorkomen. Met behulp van de Use-case en scenario's is er een activiteitsdiagram gemaakt die processen van de Security-Pi weergeeft. Als laatst zijn er sequentiediagrammen gemaakt die een beter beeld geven van de scenario's en het proces dat het systeem moet uitvoeren. De volledige ontwerpactiviteit van increment 2 staat beschreven in het ontwikkelrapport (zie bijlage, "Ontwikkelrapport – increment 2: Security – Pi")

ID	Omschrijving Requirements
REQ09	De Security-Pi gebruikt een AES-256 encryptie voor data
REQ10	De Security-Pi kan de versleutelde data ontsleutelen en weergeven
REQ11	Op de Security-Pi kan worden ingelogd met gebruikersnaam en wachtwoord
REQ12	De Security-Pi kan inloggen met een MFA

Tabel 3.1 Requirements Security-Pi

Use-Case Diagram Security-Pi

In Figuur 7.6 wordt een Use-Case diagram weergegeven van het Security-Pi systeem. Het systeem bestaat uit soortgelijke cases als de publieke dienst. De cases zijn inloggen en het registreren van een gebruiker. De andere cases zijn het plaatsen en weergeven van versleutelde diensten.



Figuur 3.1 Use-Case Diagram - Security-Pi

Scenario's Security-Pi

Ook in deze iteratie zijn er scenario's opgesteld die het doel, pre-conditie, activiteiten en alternatieven beschrijven. Er worden twee scenario's beschreven, voor een volledige beschrijving van alle scenario's wordt verwezen naar de bijlage "Ontwikkelrapport".

IC2 – SC1	Inloggen gebruiker op security-pi
Doel	De gebruiker logt in op het Security-Pi systeem
Pre-Condities	De gebruiker is geregistreerd op het Security-Pi systeem
Activiteiten	4. De gebruiker voert inloggegevens in 5. De gebruiker voert MFA uit 6. Het systeem logt de gebruiker in
Alternatieven	3. De ingevoerde inloggegevens kloppen niet 4. De uitgevoerde MFA klopt niet

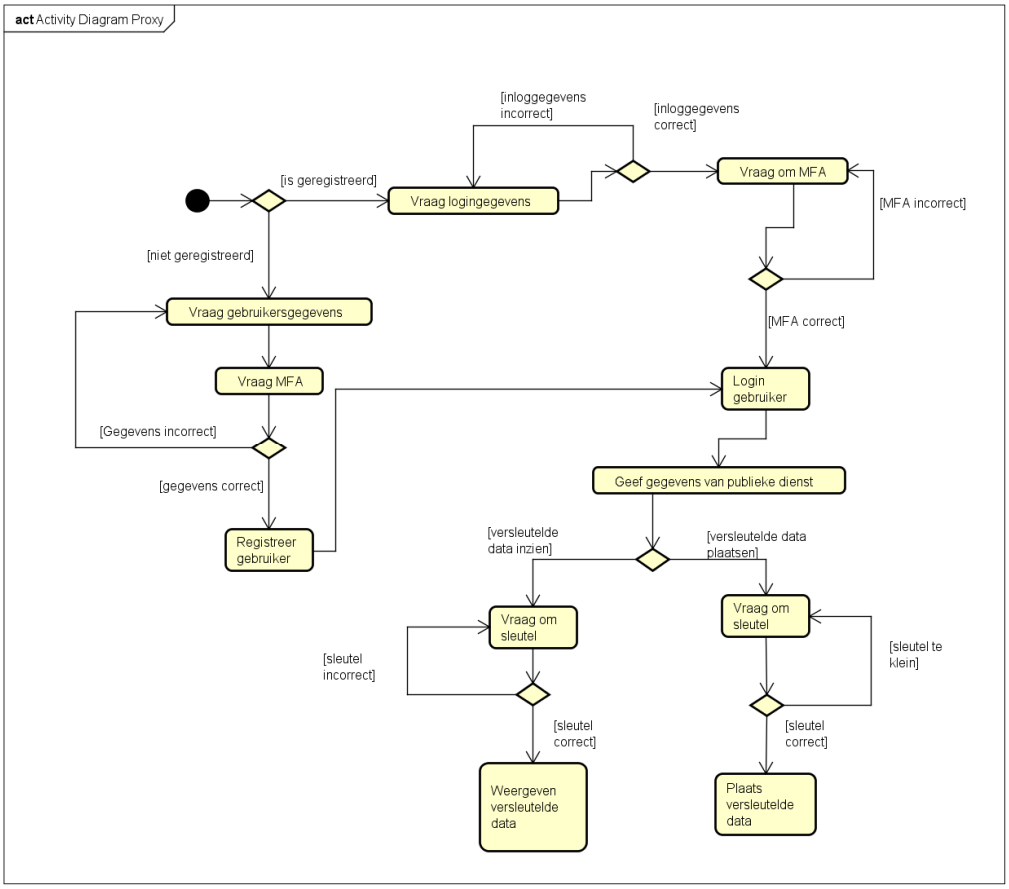
IC2 – SC2	Registreren gebruiker op security-pi
Doel	De gebruiker is geregistreerd op het Security-Pi systeem
Pre-Condities	De gebruiker is niet geregistreerd op het Security-Pi systeem
Activiteiten	1. De gebruiker geeft zijn registratiegegevens op 2. Het systeem controleert registratiegegevens 3. Het systeem vraagt om MFA-apparaat 4. De gebruiker voert MFA uit 5. Systeem registreert gebruiker met MFA - apparaat
Alternatieven	1. Registratiegegevens kloppen niet 2. Er wordt geen MFA opgegeven

IC2 – SC3	Versleutelde bericht plaatsen
Doel	De gebruiker plaatst een versleuteld bericht op het Security – Pi systeem
Pre-Condities	De gebruiker is ingelogd op het Security-Pi Systeem
Activiteiten	5. De gebruiker geeft aan dat er een versleuteld bericht geplaatst moet worden 6. De gebruiker voert bericht in 7. Het systeem versleuteld het bericht 8. Het systeem plaatst bericht op publieke dienst
Alternatieven	Geen

IC2 – SC4	Versleutelde bericht weergeven
Doel	De gebruiker kan versleutelde berichten zien
Pre-Condities	De gebruiker is ingelogd op het Security-Pi systeem
Activiteiten	1. De gebruiker geeft aan versleutelde berichten te willen zien 2. De gebruiker voert wachtwoord 3. Het systeem ontsleutelt berichten 4. Het systeem weergeeft de ontsleutelde berichten
Alternatieven	1. Het wachtwoord klopt niet 2. Er zijn geen versleutelde berichten

Activiteitsdiagram Security-Pi

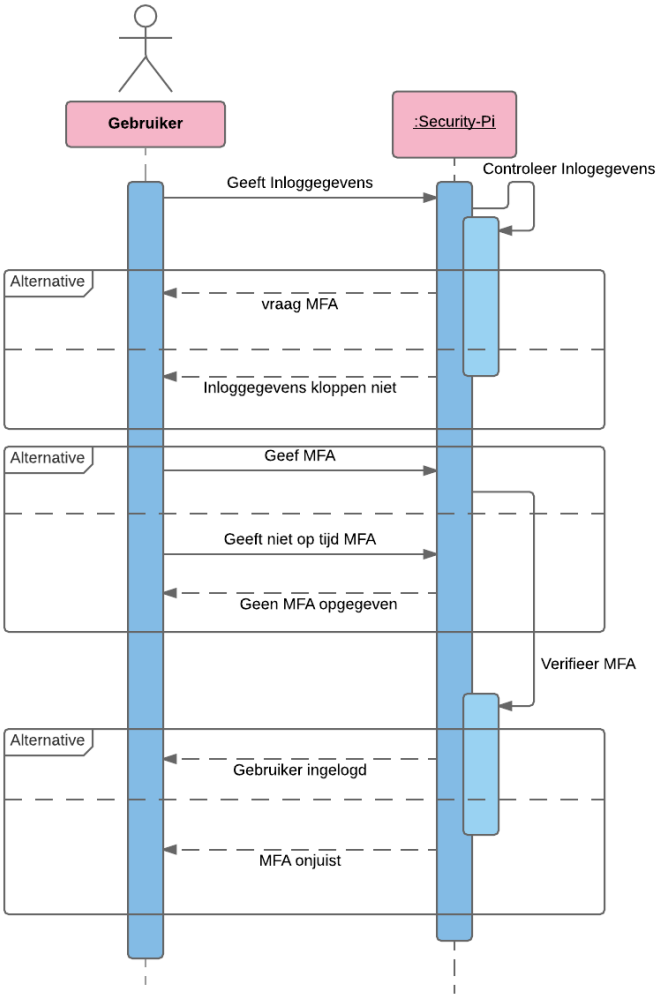
In Figuur 7.7 wordt een activiteitsdiagram weergegeven van de Security-Pi. Als een gebruiker dit systeem wil gebruiken dan kan er een keuze worden gemaakt om in te loggen of te registreren. Bij het inloggen wordt er gevraagd om inloggegevens en een MFA. Als beide ingevoerde gegevens kloppen dan wordt de gebruiker ingelogd. Bij het registreren wordt er om registratiegegevens gevraagd en MFA zodat beide geregistreerd worden. Als het registratieproces goed is verlopen is de gebruiker ook ingelogd. Een ingelogde gebruiker kan versleutelde data plaatsen en weergeven.



Figuur 3.2 Activiteitsdiagram - Security-Pi

Sequentie Diagrammen

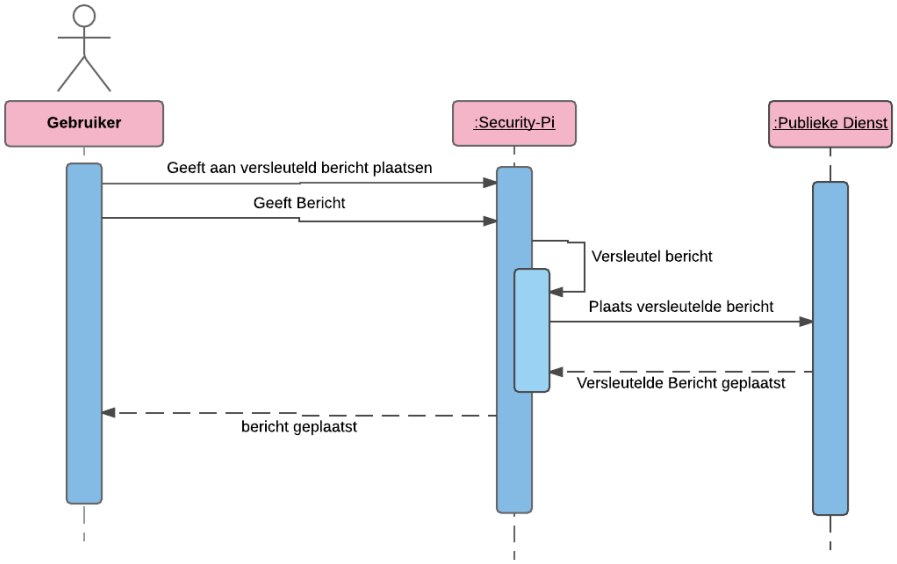
In Figuur 7.8 wordt een sequentiediagram weergegeven van het inloggen van een gebruiker op de Security-Pi. Een gebruiker geeft zijn inloggegevens op en het Security-Pi systeem controleert deze gegevens. Als de inloggegevens niet kloppen dan geeft het Security-Pi systeem aan de gebruiker een melding dat de opgegeven inloggegevens niet kloppen. Als de inloggegevens wel kloppen vraagt het Security-Pi systeem aan de gebruiker om een MFA uit te voeren. Wanneer er niet op tijd een verificatie wordt opgegeven geeft de Security-Pi een melding aan de gebruiker dat er niet op tijd een MFA is opgegeven. Als de gebruiker wel een verificatie via het MFA-apparaat heeft opgegeven wordt deze gecontroleerd. Als het MFA klopt logt het systeem de gebruiker in en stuurt een bericht naar de gebruiker dat het inloggen is gelukt. Wanneer authenticatie via het MFA-apparaat niet klopt geeft het Security-Pi systeem dit aan bij de gebruiker.



Figuur 3.3 Sequentie Diagram - Security-Pi - IC2-SC1 Inloggen gebruiker

In Figuur 3.4 wordt een sequentiediagram weergegeven dat het plaatsen van een versleutelde bericht weergeeft. De gebruiker geeft als eerst aan dat hij een versleutelde bericht wilt plaatsen en geeft dan het bericht door dat versleuteld moet worden. Versleutelt de Security-Pi het bericht en vervolgens plaatst de Security-Pi het versleutelde bericht op de publieke dienst en die geeft aan dat het bericht is geplaatst. De

Security-Pi geeft aan de gebruiker door dat het versleutelde bericht geplaatst is.



Figuur 3.4 Sequentie Diagram - Security-Pi – IC2-SC3 Versleutelde bericht plaatsen

3.2 Realisatie Security-Pi

In deze activiteit wordt het realisatiegedeelte van de Security-Pi beschreven. In de vorige activiteit zijn er ontwerpen van het Security-Pi systeem gemaakt, deze worden nu ontwikkelt tot een werkend systeem. Als eerst zal de implementatie het versleutelingsscript worden beschreven, daarna de implementatie van de Raspberry-Pi als proxy.

Implementatie versleuteling script

Er is als eerst gekeken naar het versleutelen van data met een AES-encryptie, daarna naar mogelijke manieren om de versleutelde data te plaatsen op de publieke dienst. Er zijn verschillende python cryptografie library's gevonden die mogelijk gebruikt kunnen worden in dit project. De gevonden cryptografie library's zijn:

Cryptografie Library's	Beschrijving
PyCrypto	Een Python cryptografie toolkit
M2Crypto	Python wrapper voor OpenSSL
Cryptography	Python library voor cryptografische algoritmes

Tabel 3.2 Gevonden cryptografie library's

PyCrypto

PyCrypto is de oudste en een nog veel gebruikte cryptografie toolkit als het gaat om cryptografie met Python. Veel van de hedendaagse Python library's en toolkits zijn gebaseerd op PyCrypto. Voorbeeld van een op PyCrypto gebaseerde python cryptografie library's is Google's Keyczar. PyCrypto is alleen niet zo actief en up-to-date de originele ontwikkelaar is gestopt met support van de toolkit. Een voordeel van PyCrypto is doordat het langer bestaat veel voorbeelden bestaan in het gebruik ervan.

M2Crypto

M2Crypto module is een Python wrapper om de originele c-gebaseerde OpenSSL library. M2Crypto gebruikt SWIG om Python te koppelen aan OpenSSL. Het nadeel van M2Crypto is dat het niet goed gedocumenteerd is. Vanwege onvoldoende documentatie en voorbeelden is er niet verder gekeken naar het gebruik van M2Crypto.

Cryptography

Cryptography is een nieuwer, maar een veel actiever protocol. Er komt regelmatig een nieuwe release van uit en de documentatie is heel uitgebreid. Dit maakt Cryptography het meest aantrekkelijk in gebruik. De library maakt gebruikt van CFFI, een nieuwe manier van C-code aanroepen vanaf Python.

De Cryptography library is pas later tijdens het ontwikkelen in het project naar boven gekomen. Er was te weinig tijd deze te implementeren. De PyCrypto is geen slechte keuze, omdat deze iets minder actief is als Cryptography. PyCrypto bestaat veel langer en dat geeft het gebruik ervan meer betrouwbaarheid, omdat het nog steeds veel gebruikt wordt. Als eerste moet de cryptografie methode worden voorbereid. Dit wordt hieronder weergegeven.

```
#prepare crypto method
PADDING = '{'
BLOCK_SIZE = 32
pad = lambda s: s + (BLOCK_SIZE - len(s) % BLOCK_SIZE) * PADDING

#set encryption/decryption variables
EncodeAES = lambda c, s: base64.b64encode(c.encrypt(pad(s)))
DecodeAES = lambda c, e: c.decrypt(base64.b64decode(e)).rstrip(PADDING)
```

In de volgende stuk code wordt er een encryptiekey opgegeven. Daarna wordt er een Cipherblock gegenereerd en samen met de data een AES encryptie wordt uitgevoerd.

```
        secret = request.form['encryptionkey']
        cipher = AES.new(secret)
        data = request.form['message']

        encoded = EncodeAES(cipher, data)
```

Het probleem nu is dat de encryptie nu in een ECB-modus gedaan wordt. ECB is niet veilig en het wordt afgeraden te gebruiken. Het is namelijk mogelijk in ECB-modus patronen van het originele bericht te herkennen. In Figuur 3.5 wordt ECB in vergelijking met CBC-modus weergegeven.

Figuur 3.5 ECB-modus vergeleken met CBC-modus

Het script dat de versleutelen en ontsleutelen van data uitvoert is aangepast naar het CBC-modus. De CBC-modus gebruikt het vorige versleutelde datablok om een huidige te versleutelen.

Webparsing en Webscraping

Het versleutelen en ontsleutelen is werkend. Er gezocht naar manieren om de versleutelde data te plaatsen en op de publieke dienst en ook weer te verzamelen. De volgende tools zijn gevonden die mogelijk lijken voor dit project:

Tools	Beschrijving
Scrapy	Een framework voor webcrawling, scraping en parsing
BeautifulSoup	Een python library alleen voor webparsing
Selenium	Een open-source automated testing suite voor webapplicaties

Tabel 3.3 Gevonden tools voor verzamelen en verzenden van data van/naar publieke dienst

Scrapy

Scrapy is een open source python framework voor webcrawling en webscraping. Webcrawler worden ook wel Spiders genoemd. Deze Spiders worden gebruikt om automatisch specifieke data van het internet te halen. Er wordt een lijst van URL's meegegeven en de Spider gaat ze allemaal af, maar bewaard alle links die het kan vinden op een website en voegt deze toe aan de lijst die het af moet gaan. Webscraping is het verzamelen van data die de Spider vindt op de websites. Het framework Scrapy kan mogelijk worden gebruikt voor dit project. Er zou een Spider gecreëerd kunnen worden die de versleutelde data onderscheid van de normale data.

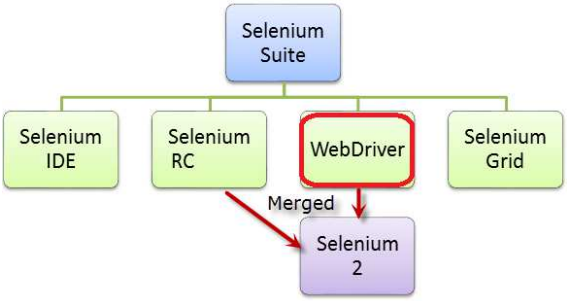
Beautiful Soup

Beautiful Soup is een webparsing python library dat zich richt op een bepaalde website. Beautiful Soup "crawlt" niet naar andere websites die het tegenkomt en gaat alleen de websites af die expliciet zijn aangegeven. Het is dan alleen mogelijk data te verzamelen van de opgegeven websites. Het voordeel bij het gebruik binnen dit project is er hoeft niet een groot framework worden gebruikt dat grotendeels alleen gebruikt wordt voor één website. Het probleem van het extracten van data zou mogelijk Beautiful Soup worden toegepast, maar het was nog altijd niet duidelijk hoe data op de publieke dienst zou worden geplaatst.

Selenium

Selenium Suite is een groot applicatiepakket dat zich richt op het automatisch testen van applicaties. Selenium Suite bestaat uit 4 componenten: Selenium IDE, Selenium RC, WebDriver en Selenium Grid. De componenten Selenium RC en WebDriver zijn samengevoegd en vormen Selenium 2. Selenium is een open-source en heeft verschillende ontwikkelaars.

Met de WebDriver is het mogelijk om vanuit verschillende programmeertalen automatisch handelingen uit te voeren via een webbrowser informatie te verzamelen van een bepaalde website. Dit maakt Scraping mogelijk, maar ook het verzenden van data naar de website. Hiermee zijn twee handelingen die de Security-Pi moet uit voeren opgelost. Het is mogelijk alleen de WebDriver component apart van Selenium Suite te gebruiken. Er hoeft ook niet een groot framework worden geïnstalleerd terwijl er maar een klein gedeelte in gebruik wordt genomen. In Figuur 7.11 wordt de opbouw van Selenium Suite weergegeven en de keuze van WebDriver.



Figuur 3.6 Opbouw van Selenium Suite met keuze WebDriver

Implementatie Security-Pi applicatie

Het samenvoegen van het versleutelingsscript en het webparsing/scraping-script is uiteindelijk de Security-Pi werkend. Het script heeft alleen nog geen UI. Het script is nu alleen werkend met een terminal. De klanten van de opdrachtgever kunnen niet allemaal omgaan met een terminal en het is nodig dat er een UI waarmee de klanten gemakkelijk hun data kunnen opgeven en dat de applicatie het verder afhandelt. In de vorige iteratie is er gewerkt met het Python framework Flask. Er is voldoende kennis nu om met Flask een webapplicatie op te zetten. Op de Raspberry-Pi is in Python Flask een webapplicatie opgebouwd geïmplementeerd zoals dat ook met Minitwit is gedaan.

Implementatie proxy

Het tweede gedeelte bestaat uit het opzetten van de Raspberry-Pi als een proxy, zodat een gebruiker bij het aansluiten van zijn PC aan de Raspberry gebruik maakt van de Raspberry als proxy om op het internet te komen. Als de gebruiker naar de betreffende publieke dienst wil gaan dan wordt de gebruiker doorverwezen naar de versleuteling applicatie op de Security-Pi. De volgende mogelijke manieren zijn onderzocht als mogelijk proxy voor de Security-Pi.

Proxy's	Beschrijving
Nginx	Webserver dat opgezet kan worden als proxyserver
Python Proxy	Een eigen in python ontwikkelde proxy
Squid	Speciaal bedoeld voor verschillende proxyconfiguraties

Tabel 3.4 Gevonden proxymogelijkheden

Nginx is al eerder gebruikt bij het opzetten van de publieke dienst, maar kan ook geconfigureerd worden als proxyserver. Op de Raspberry-Pi is om de Python Flask webapplicatie werkend te krijgen Nginx geïnstalleerd. Er is gekeken naar een manier om Nginx ook te configureren als een proxy. Het voordeel om Nginx ook als proxy te configureren is de besparing van geheugen op de Raspberry-Pi. Het nadeel is dat er geen duidelijk gescheiden implementatie is van Security-Pi applicatie en de proxy. Als er iets gebeurd met het applicatiegedeelte binnen Nginx en het programma opnieuw geïmplementeerd moet worden. Dan zal onnodig ook het proxygedeelte opnieuw moeten worden geïmplementeerd

De mogelijkheid is het creëren van een eigen in Python ontwikkelde proxy. Dit heeft als voordeel dat er geen onnodig grote proxyprogramma hoeft worden geïnstalleerd. Hiermee kan ruimte worden bespaard op de Raspberry-Pi, maar dan is het niet zeker of de proxy voldoet aan veiligheid voorwaarden. Als de Python proxy ontwikkeld wordt moet daar ook op worden gelet. Voor dit project is het niet handig om daar veel tijd in te stoppen, omdat er een tijd te kort kan ontstaan voor het implementeren van het laatste increment.

Squid is een softwarepakket gericht op het opzetten van verschillende soorten proxy's. Squid wordt wereldwijd veel gebruikt en is een goed ontwikkeld proxy met een uitgebreide documentatie. Door Squid juist te

configureren kunnen kan er niet makkelijk misbruik van de proxy gemaakt kunnen worden. Een andere reden om Squid te gebruiken is het scheiden van de webserver met de proxy. Nginx wordt al gebruikt als webserver voor de Flask applicatie. Als er iets misgaat met de applicatie of proxy is het lastig te achterhalen waar dat precies gebeurd. Squid is dan ook gekozen voor dit project om als proxy te functioneren.

Deployment proxy Squid

Squid is geïnstalleerd om te werken als proxy in het Security-Pi systeem. Al het verkeer van de gebruiker gaat eerst door de proxy daarna pas het internet op. Als gedetecteerd wordt dat de gebruiker naar de publieke dienst wil gaan dan wordt de gebruiker omgeleid naar de versleutelingsapplicatie op de Security-Pi. Daar krijgt de gebruiker de mogelijkheid om data in te voeren of te bekijken. Om dit duidelijk te maken aan Squid is SquidGuard gevonden. Met de SquidGuard is het mogelijk om bepaalde domeinen of IP-adressen te blokkeren of juist niet. Binnen SquidGuard was het mogelijk om het adres van de publieke dienst op te geven en de gebruiker dan meteen door te verwijzen naar de Security-Pi applicatie. Als de gebruiker nu naar het domein van de publieke dienst probeerde te gaan dan werd de applicatie van Security-Pi weergegeven, maar SquidGuard veranderde niet de naam van in de adresbalk. SquidGuard zorgt ervoor dat intern de verwijzing wordt gedaan. Deze werking van SquidGuard werkte nadelig op het proof of concept, want door deze interne verwijzing naar de Security-Pi pagina kon er niet verder worden geklikt naar andere pagina's van de Security-Pi applicatie. Het was een oneindige loop naar alleen de homepagina van de Security-Pi. Door SquidGuard weer te de-installeren en handmatig in Squid te de verwijzing aan te geven werkte de proxy wel goed. De gebruiker werd nu daadwerkelijk naar de pagina zelf gebracht en niet alleen een weergave van de homepagina van de Security-Pi applicatie.

Op dit moment werkt de proxy, maar de gebruiker moet zelf aangeven in de webbrowser dat hij gebruik wil maken van de proxy. De opdrachtgever verwacht een oplossing waar de klant zelf niks voor hoeft te doen. Een "plug&play" oplossing voor de klant. Dit houdt in dat de klant het alleen maar hoeft aan te sluiten en ermee aan de slag kan. Om dit op te lossen zal de Squid proxy transparant worden geconfigureerd. Met een transparante proxy hoeft de gebruiker niet zelf aan te geven en zijn browser dat hij gebruikt wil maken van een proxy. Er was helaas op dit moment geen tijd genoeg voor deze increment om aan te werken. Er is besproken met de opdrachtgever dit gedeelte later als er voldoende tijd is te implementeren.

3.3 Testen Security – Pi

In deze paragraaf worden de testen beschreven die gaan over de Security-Pi. Voor het testen van de Security-Pi zijn ook de scenario's genomen die zijn opgesteld in de ontwerpactiviteit. Deze scenario's zijn omgezet tot testcases. De uitgevoerde testen van dit increment worden beschreven in het Testrapport hoofdstuk 3.2 increment 2 Security-Pi.

4 Increment 3: Opzetten MFA

Dit hoofdstuk beschrijft het ontwikkelen van de MFA in het proof of concept. Ook in dit hoofdstuk wordt er als eerst een ontwerp van de werking van het MFA gemaakt. Als tweede zal de realisatie van de MFA worden beschreven. Als laatste zal de realisatie van de MFA worden getest.

4.1 Ontwerp MFA

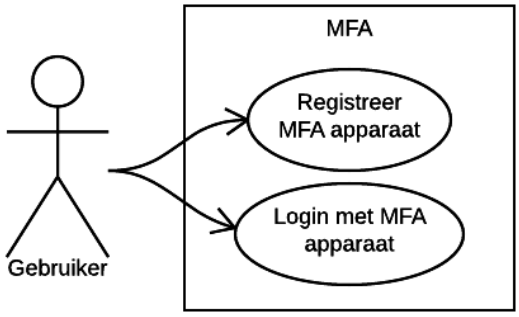
In deze activiteit wordt het ontwerp van het MFA-gedeelte van de Security-Pi gemaakt. Er worden vastgestelde eisen genomen die gaan over het MFA en omgezet naar een Use-Case diagram en daarvan weer mogelijke scenario's opgesteld. Met behulp van de Use-Case diagram en scenario's is er een activiteitsdiagram gemaakt.

ID	Omschrijving Requirements
REQ13	De MFA heeft een krachtige bescherming tegen phishing-aanvallen
REQ14	De MFA is een fysiek apparaat
REQ15	De MFA kan niet makkelijk worden vergeten
REQ16	De MFA kan biometrisch identificeren

Tabel 4.1 Requirements MFA

Use-Case diagram MFA

In Figuur 7.13 wordt een Use-Case diagram van het MFA-systeem weergegeven. De werking van het MFA is simpel. Een gebruiker kan het MFA-systeem laten registreren of ermee inloggen.



Figuur 4.1 Use-Case Diagram – MFA – U2F

Scenario's MFA

Voor elke case is er een scenario opgesteld dat de werking van elke case weergeeft. Er zijn in totaal twee mogelijke scenario's voor het MFA-systeem. De opgestelde scenario's zijn:

IC3 – SC1	Gebruikers Login Authenticatie
Doel	De gebruiker is ingelogd met MFA-apparaat
Pre-Condities	De gebruiker heeft de correcte gebruikersnaam en wachtwoord ingevoerd
Activiteiten	5. Het systeem vraagt om verificatie op het MFA-apparaat 6. De gebruiker geeft verificatie op MFA-apparaat 7. Het systeem controleert de ontvangen verificatie 8. De gebruiker is geverifieerd met MFA-apparaat

Alternatieven	2. Controle geeft aan dat de ontvangen verificatie van MFA-apparaat niet klopt

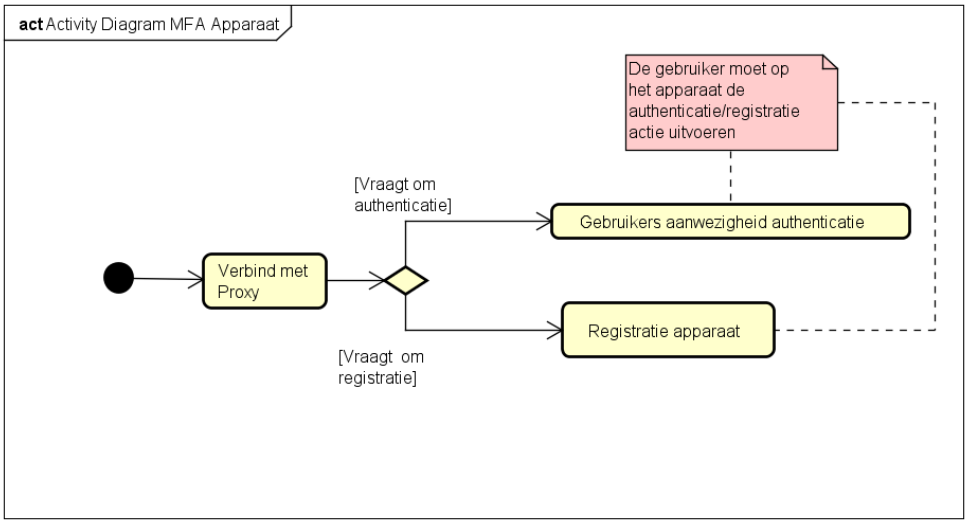
IC3 – SC2	Registratie MFA-apparaat
Doel	De gebruiker heeft een MFA-apparaat toegevoegd aan zijn account
Pre-Condities	2. De gebruiker is ingelogd op het Security-Pi Systeem
Activiteiten	6. De gebruiker geeft dat er een MFA-apparaat toegevoegd moet worden aan zijn account 7. Het systeem vraagt om bevestiging van MFA-apparaat dat toegevoegd moet worden 8. De gebruiker geeft registratie vanaf MFA-apparaat 9. Het systeem controleert de ontvangen registratie bericht 10. Het systeem registreert MFA-apparaat aan account
Alternatieven	Controle geeft aan dat MFA-apparaat al geregistreerd is op een ander account.

IC3 – SC1	Registreer U2F-stick
Doel	Het MFA apparaat is geregistreerd
Pre-Condities	Het MFA-apparaat is niet geregistreerd
Activiteiten	5. Het systeem vraagt om op U2F knop te drukken 6. De gebruiker drukt op knop op U2F-stick 7. U2F signeert de aanvraag met registratiegegevens 8. Het systeem registreert de U2F-stick als MFA
Alternatieven	3. Er wordt niet op knop van U2F-stick gedrukt 4. U2F-stick is al geregistreerd

IC3 – SC2	Authenticatie met U2F-stick
Doel	Verifieer gebruiker met MFA apparaat
Pre-Condities	Het MFA-apparaat is geregistreerd
Activiteiten	5. Het systeem vraagt om verificatie van de U2F-stick 6. De gebruiker drukt op knop van U2F-stick 7. U2F stick signeert de aanvraag met authenticatiegegevens 8. Het systeem het de gebruiker geverifieerd
Alternatieven	3. Er wordt een verkeerde U2F-verificatie verstuurd 4. Er wordt niet op de knop van U2F-stick gedrukt

Activiteitsdiagram

In Figuur 7.14 wordt een activiteitsdiagram van het MFA-apparaat weergegeven. Het MFA-apparaat heeft een BLE-connectie met de Security-Pi. Over deze connectie kan de Security-Pi vragen om een registratie of authenticatie met het MFA-systeem.



Figuur 4.2 Activiteitsdiagram - MFA

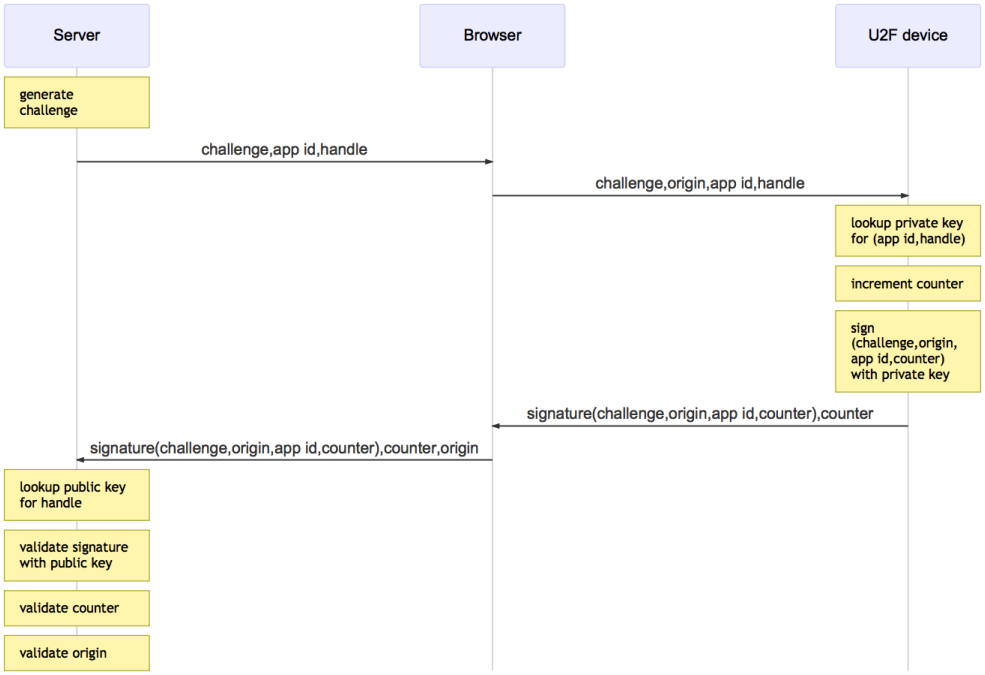
Sequentie diagrammen

In Figuur 4.3 wordt een sequentiediagram van het authenticatieproces van het U2F-protocol weergegeven. Het authenticatieproces slaagt alleen als het u2f-apparaat is geregistreerd. De server genereerd een challenge, geeft deze door aan de browser en de browser aan het U2F-apparaat. Het U2F-apparaat signeert de challenge met de private key. De challenge wordt dan weer terug gestuurd naar de server die het controleert met zijn public key.

Tegen phishing-aanvallen wordt er door de browser een origin een origin verstuurd. Het U2F-apparaat signeert de challenge en stuurt de origin mee terug naar de server. Zo kan de server controleren of de challenge klopt en er geen MITM of een ander soort aanval is uitgevoerd.

De appID en handle die worden meegestuurd vanaf de server zorgt ervoor dat de juist keypair gevonden kan worden op de het U2F-apparaat en op de server. Zo kan de provider van een dienst niet bijhouden of er een key voor meerdere diensten gebruikt wordt, een privacy-beveiliging.

Op het U2F-apparaat wordt er na elke signering een counter opgeteld. Dit zorgt voor een beveiliging tegen klonen. Als de server ziet dat de counter waarde niet klopt met wat hij voor het laatst heeft gezien. Dan is het apparaat gekloond. Het is geen veilige beveiliging, omdat als het gekloonde apparaat eerder wordt gebruikt dan het originele u2f-apparaat dan wordt de gekloonde apparaat herkend als de originele apparaat.



Figuur 4.3 Authenticatie proces U2F[34]

In Figuur 4.4 wordt een sequentiediagram van het registratieproces van het U2F-protocol weergegeven. Het is ongeveer hetzelfde als de authenticatieproces, maar er komen nog een paar handelingen bij. Het U2F-apparaat signeert ditmaal niet als hij de challenge ontvangt, maar genereerd een sleutel paar. De private key wordt opgeslagen op het U2F-apparaat en de public key wordt verzonden naar de server.

Figuur 4.4 Registratie proces U2F[35]

[34] linux.conf.au 2017 – Hobart, Tasmania
[35] linux.conf.au 2017 – Hobart, Tasmania

4.2 Realisatie MFA

In deze activiteit wordt het realisatiegedeelte van het MFA-systeem beschreven. In de vorige activiteit zijn er ontwerpen van het MFA-systeem gemaakt. De ontwerpen worden gebruikt voor het implementeren van het MFA-systeem in het proof of concept. Er is eerder besloten om een BLE/U2F – MFA op te zetten op een smartphone. Als eerst wordt er een beschrijving van een connectie met BLE gegeven. Ten tweede zal worden beschreven hoe het U2F protocol is geïmplementeerd. Als laatste zal er een beschrijving volgen van samenvoegen van het BLE met het U2F-protocol.

Implementeren BLE-connectie

Als eerst is het mogelijk gemaakt om een BLE-connectie met de Security-Pi te maken. De raspberry moet BLE-connecties kunnen opzetten. Om een connectie met de Raspberry te kunnen opzetten moet de raspberry vindbaar gemaakt worden. De Raspberry is hiervoor opgezet als BLE Beacon, zodat deze gevonden kan worden. Als de Beacon zichtbaar is moet het ook connecties toestaan. In Kader 7.3 wordt weergegeven hoe de raspberry is ingesteld als Beacon:

- Rood = Hiermee wordt aangegeven dat de Raspberry moet adverteren en connecties toelaat.
- Groen = Dit gedeelte bepaalt het type Beacon en het bericht dat de raspberry adverteert. Het type Beacon is in dit geval een Eddystone Beacon, omdat deze makkelijk in gebruik zijn met Android en Apple smartphones. Een alternatief was IBeacon, maar die type Beacon werkt officieel alleen met Apple smartphones.

```
sudo hciconfig hci0 up

sudo hciconfig hci0 leadv 0

sudo hcitool -i hci0 cmd 0x08 0x0008 1E 02 01 1A 1A FF 4C
00 02 15 E2 0A 39 F4 73 F5 4B C4 A1 2F 17 D1 AD 07 A9 61
00 00 00 00 C8 00
```

Kader 4.1 Implementatie BLE Beacon

Implementeren U2F

In dit gedeelte wordt de implementatie van het U2F-protocol beschreven. Om het U2F-protocol te implementeren over een BLE-connectie is het nodig om de werking van het U2F-protocol te bestuderen. Er is een werkende U2F registratie en authenticatie systeem opgezet op de Security-Pi applicatie. Hiervoor is een U2F-stick "YubiKey NEO" van Yubico gebruikt. Yubico biedt ontwikkelaars library's aan die het mogelijk maken om een U2F MFA te implementeren in applicatie. De library die hiervoor is gebruikt is de Python library "u2fval-client". De library maakt het mogelijk om met de U2F validatie server te communiceren. De U2F validatie server verzorgt de registratie en validatie van U2F-apparaten.

U2F Validatie Client

De U2Fval-Client library is als eerste opgezet, zodat er op de Security-Pi ingelogd kan worden met een MFA. De werking kan getest worden met een server van Yubico. Yubico heeft een U2F validatie server opgezet voor ontwikkelaars, zodat zij deze kunnen gebruiken voor hun eigen applicaties. De server kan worden gebruikt via https://u2fval.appspot.com/. De gebruiker kan kiezen om deze applicatie blijven te gebruiken voor de validaties, maar dan wordt de informatie van gebruikers opgeslagen op de Yubico server. De opdrachtgever wil graag een veilige dienst zonder dat daar nog een derde partij een rol in speelt. In Kader 4.2 wordt het registreren van een U2F-sleutel weergegeven. Het registratie proces wordt afgehandeld door de validatie server van Yubico. In de rood gemarkeerde regel wordt het registratieproces aangeroepen met "u2fval.register_begin(GebruikerID)"

```
@app.route('/u2f_register', methods=['POST'])
def u2f_register():
    """Register a U2F device"""
    reg_req = u2fval.register_begin(get_current_user())
    return render_template('u2f_add.html',
                           name=request.form['name'],
                           reg_req=reg_req)
```

Kader 4.2 Registratie U2F-Sleutel

In kader wordt het authenticatieproces weergegeven. Ook het authentiseren van de U2F-sleutel wordt afgehandeld door de validatie server van Yubico. Het is alleen nodig om de juiste gebruikersnaam mee te geven. Het authenticatieproces wordt aangeroepen met "u2fval.auth_begin(gebruikersID)"

```
        try:
            session['u2f_user_id'] = user['user_id']
            auth_req = u2fval.auth_begin(str(user['user_id']))
            return render_template('u2f_auth.html', auth_req=auth_req)
```

Kader 4.3 Authenticatie van U2F-sleutel

U2F Validatie Server

Op dit moment wordt de validatie van een U2F-sleutel gedaan door de validatie server van Yubico. Het bedrijf Yubico heeft een Python library uitgegeven om zelf de validatie lokaal op te zetten. Deze library heet “python-u2flib-server ” en geïmplementeerd op de Security-Pi, zodat validatie lokaal gedaan kan worden. Hier was onvoldoende tijd voor in het project. De opdrachtgever heeft aangegeven dat het mogelijk is in de tijd tussen de deadline van de scriptie en verdediging alsnog gerealiseerd kan worden. Als dat niet haalbaar dan kan het worden opgenomen in een vervolgend project hierop.

Implementeren BLE/U2F – MFA

Uiteindelijk worden het BLE-connectie gedeelte en U2F protocol samengevoegd tot een MFA. De MFA-applicatie is ontwikkeld in Java en er is hiervoor een Samsung Galaxy S6 gebuikt met Android 6.0.1Deze smartphone is gekozen, omdat dit de enige smartphone was die de student beschikbaar had. De smartphone heeft een vingerafdrukscanner die mogelijk gebruikt kan worden als de opdrachtgever alsnog een biometrische beveiliging erbij zou willen hebben, mits er voldoende tijd beschikbaar was. Bij de vorige increment was er al sprake van een tijdstekort. Het is niet gelukt dit gedeelte van het proof of concept te voltooien. Het proof of concept heeft op dit moment wel een werkend U2F MFA waar de opdrachtgever genoeg aan heeft.

4.3 Testen MFA

In deze paragraaf worden de testen beschreven die gaan over de MFA. Voor het testen van de MFA zijn weer de scenario’s genomen die zijn opgesteld in de ontwerpactiviteit. Deze scenario’s zijn omgezet tot testcases. De uitgevoerde testen van dit increment worden beschreven in het Testrapport hoofdstuk 3.3 increment 3 MFA.

5 Conclusie ontwikkelfase

Het eerste increment beschrijft het ontwikkelen van het publieke systeem. De opdrachtgever had aan de start van het project aangegeven het liefst een bestaand publieke dienst te gebruiken, zoals Facebook of Twitter. Het was vrij snel al duidelijk dat het niet geschikt is meteen een bestaande publieke dienst te gebruiken. Deze publieke diensten zijn goed beveiligd en dat maakt het lastig om er een systeem te ontwikkelen dat er gebruik van maakt. Er is dan ook snel een gesprek met de opdrachtgever ingepland en aangegeven om zelf een publiek systeem op te zetten. Dit heeft als resultaat een twitter kloon “Minitwit” gegeven.

In het tweede increment wordt de Security-Pi ontwikkeld. De Security-Pi is opgedeeld in twee gedeeltes. De Security-Pi applicatie die het versleutelen en ontsleutelen van data doet en een proxy-gedeelte. Beiden zijn uiteindelijk succesvol gerealiseerd. Tijdens het ontwikkelen werd duidelijk dat er een onveilige modus van AES werd gebruikt, maar deze is aangepast naar een wel veilige modus van AES.

Bij het opzetten van de tweede gedeelte van de proxy is er gezocht een geschikte proxy voor dit project. Er is een keuze gemaakt om “Squid” hiervoor te gebruiken. Een probleem was dat er library SquidGuard was gevonden die verwijzingen in Squid heel simpel zou kunnen uitvoeren. SquidGuard zorgde juist voor het probleem dat het de verwijzing niet juist deed. Uiteindelijk is het opgelost door SquidGuard weer te verwijderen en handmatig de verwijzing in Squid te verwerken.

Het derde increment beschrijft het ontwikkelen van een MFA voor het proof of concept. Er is in de definitiefase een beslissing gemaakt om het sterke U2F-protocol te combineren met BLE. Het is gelukt de BLE gedeelte op te zetten en het U2F gedeelte, maar niet samen te voegen tot een geheel. Dit is besproken met de opdrachtgever en hebben besloten alleen het U2F als MFA te gebruiken in het proof of concept.

Het proof of concept is met het uitvoeren van drie incrementen voltooid. Het resultaat toont aan dat het mogelijk is data dat bij een externe dienst staat opgeslagen kan worden beveiligd. Het aanbieden van deze extra beveiliging kan voor klanten een bepalingspunt zijn om juist klant te worden van ICT Group.

Bijlage

Testrapport

ICT Group N.V.

HET ONTWIKKELEN VAN EEN EXTRA BEVEILIGING VOOR NETWERKCOMMUNICATIE VAN KLANTEN NAAR DE CLOUD VAN ICT GROUP

Testrapport

Technische Informatica
De Haagse Hogeschool
Versie 0.2

Inhoudsopgave

1 Inleiding

Met behulp van testen wordt er gevalideerd of het product voldoet aan de eisen van de opdrachtgever. Er worden mogelijke scenario's gebruikt in de tests en er wordt beschreven of deze voldoet aan de eisen en wensen van de opdrachtgever. Als eerste is er een testplan opgesteld en na elke increment wordt het testrapport bijgewerkt met de testen per increment.

2 Testplan

Tijdens het ontwikkelen van het proof of concept is er steeds getest of een bepaalde implementatie gelukt is. Het is voor de opdrachtgever belangrijk goed om duidelijk te zien wat er precies werkend is. Het testplan beschrijft wat er precies getest zal worden.

2.1 Aanpak

Voor dit project is er besloten validatietesten uit te voeren. Deze worden door de student zelf gedaan en uiteindelijk gepresenteerd aan de opdrachtgever. De opdrachtgever kan dan aangeven of hij tevreden is met het resultaat. De mogelijke scenario's per increment worden gebruikt als een testscenario. Elke testscenario beschrijft het doel, pre-conditie, activiteiten, alternatieven, verwachte resultaat, resultaat en of de testscenario geslaagd is. Als een testscenario niet geslaagd is wordt daarbij

Increment 1:

Hier worden de scenario's weergegeven van increment 1. Er wordt getest of het mogelijk is deze scenario's uit te voeren op het proof of concept. De volgende scenario's worden getest:

- IC-SC1 Inloggen gebruiker
- IC-SC2 Registreer gebruiker
- IC-SC3 Bericht plaatsen
- IC-SC4 Weergeven bericht

Increment 2:

Dit gedeelte beschrijft de testscenario's van increment 2. De eerder opgestelde scenario's uit increment 2 worden gevalideerd.

- IC2-SC1 Inloggen gebruiker op security-pi
- IC2-SC2 Registreren gebruiker op security-pi
- IC2-SC3 Versleutelde bericht plaatsen
- IC2-SC4 Versleutelde bericht weergeven

Increment 3:

Hieronder volgen de testscenario's van increment 3.

- IC3-SC1 Registreer MFA
- IC3-SC2 Authenticatie MFA

3 Testen

In dit hoofdstuk worden de uitgevoerde testen weergegeven. Bij elke testscenario wordt een verwachte resultaat en resultaat beschreven. Als laatste wordt beschreven of de testscenario geslaagd is.

3.1 increment 1 Publieke Dienst

Deze paragraaf worden de testen beschreven die betrekking hebben tot increment 1 de publieke dienst.

IC1 – TC1	Inloggen gebruiker
Doel	Het publieke systeem logt de gebruiker in
Pre-Condities	De gebruiker is geregistreerd
Activiteiten	10. De gebruiker voert inloggegevens in 11. Het systeem controleert de gegevens 12. Het systeem logt de gebruiker in
Alternatieven	7. De ingevoerde gegevens van de gebruiker kloppen niet 8. De gebruiker wordt teruggestuurd naar het inlogscherm
Verwachte Resultaat	De gebruiker is ingelogd op publieke dienst Minitwit
Resultaat	De gebruiker is ingelogd op publieke dienst Minitwit
Geslaagd	JA

Tabel 3.1 Testcase 1- inloggen gebruiker

IC1 – TC2	Registreer gebruiker
Doel	Het publieke systeem registreert de gebruiker
Pre-Condities	De gebruiker niet geregistreerd
Activiteiten	4. De gebruiker voert registratiegegevens in 5. Het systeem controleert de registratiegegevens 6. Het systeem registreert de gebruiker
Alternatieven	3. De ingevoerde registratiegegevens van de gebruiker voldoet niet aan de eisen van het systeem 4. Er wordt met een melding aan de gebruiker duidelijk gemaakt welke van de inloggegevens niet kloppen
Verwachte Resultaat	De gebruiker is geregistreerd op publieke dienst Minitwit
Resultaat	De gebruiker is geregistreerd op publieke dienst Minitwit
Geslaagd	JA

Tabel 3.2 Testcase 2 - Registreer gebruiker

IC1 – TC3	Bericht plaatsen
Doel	De gebruiker plaatst een bericht op het publieke systeem
Pre-Condities	De gebruiker is ingelogd
Activiteiten	10. De gebruiker voert een bericht in 11. De gebruiker geeft aan dat het bericht gedeeld mag worden 12. Het systeem bewaard het bericht in een database
Alternatieven	Geen
Verwachte resultaat	Bericht is geplaatst op publieke dienst Minitwit
Resultaat	Bericht is geplaatst op publieke dienst Minitwit
Geslaagd	JA

Tabel 3.3 Testcase 3 - Bericht plaatsen

IC1 – TC4	Weergeven bericht
Doel	Het publieke systeem weergeeft de geplaatste berichten
Pre-Condities	Geen
Activiteiten	2. Het systeem geeft alle opgeslagen berichten weer op een tijdlijn
Alternatieven	Geen
Verwachte Resultaat	Alle geplaatste berichten worden weergegeven op publieke dienst Minitwit
Resultaat	Alle geplaatste berichten worden weergegeven op publieke dienst Minitwit
Geslaagd	JA

Tabel 3.4 Testcase 4 - Weergeven bericht

3.2 Increment 2 Security – Pi

In deze paragraaf worden de testen beschreven die betrekking hebben op increment 2 Security-Pi.

IC2 – TC1	Inloggen gebruiker op security-pi
Doel	De gebruiker logt in op het Security-Pi systeem
Pre-Condities	De gebruiker is geregistreerd op het Security-Pi systeem
Activiteiten	7. De gebruiker voert inloggegevens in 8. De gebruiker voert MFA uit 9. Het systeem logt de gebruiker in
Alternatieven	5. De ingevoerde inloggegevens kloppen niet 6. De uitgevoerde MFA klopt niet
Verwachte Resultaat	De gebruiker is ingelogd op de Security-Pi
Resultaat	De gebruiker is ingelogd op de Security-Pi
Geslaagd	JA

Tabel 3.5 Testcase 1 - Inloggen gebruiker op Security-Pi

IC2 – TC2	Registreren gebruiker op security-pi
Doel	De gebruiker is geregistreerd op het Security-Pi systeem
Pre-Condities	De gebruiker is niet geregistreerd op het Security-Pi systeem
Activiteiten	6. De gebruiker geeft zijn registratiegegevens op 7. Het systeem controleert registratiegegevens 8. Het systeem vraagt om MFA-apparaat 9. De gebruiker voert MFA uit 10. Systeem registreert gebruiker met MFA - apparaat
Alternatieven	3. Registratiegegevens kloppen niet 4. Er wordt geen MFA opgegeven
Verwachte Resultaat	De gebruiker is geregistreerd op de Security-Pi
Resultaat	De gebruiker is geregistreerd op de Security-Pi
Geslaagd	JA

Tabel 3.6 Testcase 2 - Registreren gebruiker op Security-Pi

IC2 – TC3	Versleutelde bericht plaatsen
Doel	De gebruiker plaatst een versleuteld bericht op het Security – Pi systeem
Pre-Condities	De gebruiker is ingelogd op het Security-Pi Systeem
Activiteiten	9. De gebruiker geeft aan dat er een versleuteld bericht geplaatst moet worden 10. De gebruiker voert bericht in 11. Het systeem versleuteld het bericht 12. Het systeem plaatst bericht op publieke dienst
Alternatieven	Geen
Verwachte Resultaat	Er is een versleutelde bericht geplaatst via de Security-Pi op de publieke dienst Minitwit.
Resultaat	Er is een versleutelde bericht geplaatst via de Security-Pi op de publieke dienst Minitwit.
Geslaagd	JA

Tabel 3.7 Testcase 3 - Versleutelde bericht plaatsen

IC2 – TC4	Versleutelde bericht weergeven
Doel	De gebruiker kan versleutelde berichten zien
Pre-Condities	De gebruiker is ingelogd op het Security-Pi systeem
Activiteiten	5. De gebruiker geeft aan versleutelde berichten te willen zien 6. De gebruiker voert wachtwoord 7. Het systeem ontsleutelt berichten 8. Het systeem weergeeft de ontsleutelde berichten
Alternatieven	3. Het wachtwoord klopt niet 4. Er zijn geen versleutelde berichten
Verwachte Resultaat	De versleutelde berichten die op de publieke dienst Minitwit staan worden weergegeven op de Security-Pi
Resultaat	De versleutelde berichten die op de publieke dienst Minitwit staan worden weergegeven op de Security-Pi
Geslaagd	JA

Tabel 3.8 Testcase 4 - Versleutelde bericht weergeven

3.3 Increment 3 MFA

Deze paragraaf beschrijft de testen die uitgevoerd zijn in increment 3 MFA. De Testscenario's in IC3-TC1 en IC3-TC2 zijn bedoeld van het U2F/BLE MFA. Er is uiteindelijk niet voldoende tijd meer geweest om dit te realiseren. De testcases zijn aangepast naar testscenario's met een U2F-stick als MFA.

IC3 – TC1	Registreer U2F/BLE-apparaat
Doel	Het MFA apparaat is geregistreerd
Pre-Condities	Geen
Activiteiten	9. Het systeem vraagt om MFA 10. De gebruiker maakt BLE connectie met systeem via MFA-apparaat 11. De gebruiker drukt op registratie knop via MFA-apparaat 12. Het systeem registreert het MFA-apparaat als MFA
Alternatieven	5. Er wordt niet op registratieknop van MFA-apparaat gedrukt 6. MFA-apparaat is al geregistreerd
Verwachte Resultaat	De gebruiker heeft zijn MFA-apparaat geregistreerd op de Security-Pi via BLE connectie
Resultaat	BLE connectie mogelijk, maar nog geen U2F registratie proces
Geslaagd	NEE, Niet voldoende tijd voor ontwikkeling

IC3 – TC2	Authenticatie met U2F/BLE-apparaat
Doel	Verifieer gebruiker met MFA apparaat
Pre-Condities	Geen
Activiteiten	9. Het systeem vraagt om verificatie van het U2F/BLE MFA-apparaat 10. De gebruiker maakt BLE connectie met Security-Pi via U2F/BLE MFA-apparaat 11. De gebruiker drukt op authenticatie knop via MFA-apparaat 12. Het Security-Pi systeem heeft de gebruiker geverifieerd
Alternatieven	5. Er wordt een verkeerde U2F-verificatie verstuurd van U2F/MFA-apparaat 6. Er wordt niet op de verificatieknop van U2F/BLE MFA-apparaat gedrukt
Verwachte Resultaat	De gebruiker is geverifieerd met zijn U2F/BLE MFA-apparaat
Resultaat	BLE connectie mogelijk, maar U2F authenticatie mogelijk
Geslaagd	NEE, niet voldoende tijd voor ontwikkeling

Hieronder volgen de aangepaste testscenario's die het testen met een U2F-stick als MFA beschrijft. De vorige testen waren niet geslaagd, omdat er gewoonweg geen tijd meer was om het te ontwikkelen. De testen zijn aangepast naar het testen met een U2F-stick. Op deze manier is er een sterk en toch nog een werkend MFA voor het proof of concept.

IC3 – TC1.1	Registreer U2F-stick
Doel	De U2F-stick is geregistreerd
Pre-Condities	De U2F-stick is niet geregistreerd
Activiteiten	5. Het Security-Pi systeem vraagt om op U2F knop te drukken 6. De gebruiker drukt op knop op U2F-stick 7. U2F signeert de aanvraag met registratiegegevens 8. Het Security-Pi systeem registreert de U2F-stick als MFA
Alternatieven	3. Er wordt niet op knop van U2F-stick gedrukt 4. U2F-stick is al geregistreerd
Verwachte Resultaat	De gebruiker heeft zijn U2F-Stick geregistreerd op de Security-Pi
Resultaat	De gebruiker heeft zijn U2F-Stick geregistreerd op de Security-Pi
Geslaagd	JA

Tabel 3.9 Testcase 1 -Registreer U2F-Stick

IC3 – TC2.1	Authenticatie met U2F-stick
Doel	Verifieer gebruiker met U2F-stick
Pre-Condities	De U2F-stick is geregistreerd
Activiteiten	5. Het Security-Pi systeem vraagt om verificatie van de U2F-stick 6. De gebruiker drukt op knop van U2F-stick 7. U2F stick signeert de aanvraag met authenticatiegegevens 8. Het Security-Pi systeem het de gebruiker geverifieerd
Alternatieven	3. Er wordt een verkeerde U2F-verificatie verstuurd 4. Er wordt niet op de knop van U2F-stick gedrukt
Verwachte Resultaat	De gebruiker is geverifieerd met een U2F-stick
Resultaat	De gebruiker is geverifieerd met een U2F-stick
Geslaagd	JA

Tabel 3.10 Testcase 2 – Authenticatie

4 Conclusie Testrapport

Als conclusie vanuit dit testrapport kan worden gehaald dat het proof of concept werkend is. De testen zijn gevalideerd door de student en gepresenteerd aan de opdrachtgever. Er waren wat problemen met de tijd, maar die zijn opgelost door aanpassingen te maken aan de testscenario's. Zo kan het project alsnog voltooid worden en is de opdrachtgever tevreden met het resultaat.

Interviewverslagen

Meeting: Interview “Opdracht ophelderen”

Datum: 10-2-2017

Medewerker: Bart Lamot (opdrachtgever), André van ‘t Hof

2. Wat is de huidige situatie bij klanten die gebruik willen gaan maken van deze extra beveiliging?

   “De huidige situatie is niet slecht. De klanten zijn er bang voor dat het kan gebeuren. Het idee is uitgewerkt door Bart, dat er een box komt dat de alleen medewerkers die via de box data verzenden naar de “cloud” alleen de mensen die ook in bezit zijn van dezelfde box de data kunnen zien. Mensen die van dezelfde dienst gebruik maken kunnen deze berichten niet zien en zien alleen de “normale” data waar zij alleen toegang toe hebben. ICT wil dit product op meerdere markten uitbrengen en niet alleen specifiek voor die havenbedrijven.”

3. Zijn er collega’s die zich hiermee bezig houden of hebben aan gewerkt binnen ICT Group?

   “Christiaan Woldendorf is een collega die wel er iets van af weet. Hij zit toch bij jou op de afdeling Deze zou je zeker een keer moeten spreken.”

4. Is het bekend welke middel er gebruikt mag worden bij klanten, want een USB is niet de enige fysieke device waar de encryptie-key op zou kunnen staan.

   “Ik ben het met je eens dat USB verouderd is. Het fysieke apparaat mag je zelf bepalen met een juiste motivatie.”

5. Waarom niet gewoon een VPN verbinding tussen het systeem van de klant en de Azure cloud van ICT?

   “De aanleiding was dus inderdaad die bedrijven in de havens, maar de opdracht is daar niet specifiek voor bedoeld. Op deze manier van een extra beveiliging willen we op meerdere markten uit brengen. Ook op plekken waar het bijvoorbeeld niet zou kunnen zo een VPN en dan moet je ook nog kijken of je de 3-factor authenticatie kan toepassen”

6. Is een Rasberry Pi wel een veilig apparaat, er zou makkelijk een MITM aanval op worden gedaan.

   “De Raspberry Pi is het apparaat dat beschikbaar is voor het onderzoek. Als er uit een onderzoek uitkomt dat een ander apparaat beter is dan kunnen we kijken of het mogelijk is die aan te schaffen. Voor nu willen we graag een Raspberry Pi, omdat deze generiek en vervangbaar is. ”

Meeting: Voortgang gesprek

Datum: 13-2-2017

Medewerker naam: André van ’t Hof

Wat is nu de definitieve ontwikkelmethode?

Er is gekozen voor RAD, omdat bij het gebruik van RUP in de beginfase veel wordt gevraagd te documenteren en nog niet bezig te houden met analyse naar probleemdomein. Bij RAD wordt er veel gewerkt met prototypes dat is handig, omdat het eindproduct een proof of concept is. De methode RAD richt zich op zo snel mogelijk een stuk werkend product op te leveren. Dit geeft als voordeel dat er al vroeg een inzicht is in de haalbaarheid van het project of dat het project moet worden aangepast.

Wat ga ik nu als eerst onderzoeken?

De planning is af en de opdracht is duidelijk. Het belangrijkste nu is onderzoeken hoe je data kan onderscheiden van elkaar. Dat punt is nog niet duidelijk hoe dit werkt. Bij de fysieke sleutel en encryptie methode is het een kwestie van een aantal mogelijkheden vergelijken en daarin een keuze te maken.

Hoe ga ik de data onderscheiden met elkaar?

Doe onderzoek op internet, je kan bijvoorbeeld met Wireshark proberen een verschil te herkennen. Er zijn collega’s die je kan raadplegen en er is ook nog Yammer een social media website maar dan alleen voor het bedrijf ICT.

Welke mogelijkheden zijn er in gebruik van fysieke sleutel?

Een USB sleutel met “bad-blocks” daarin creëren komt nog uit de tijd van de floppy. Kijk of er mogelijkheden zijn met je smartphone. Dat is iets wat je wel altijd bij je hebt en voldoet dus aan de factor “something you have”.

Mogelijkheden in keuze biometrische beveiliging?

Zelf even kijken wat het handigst is hierin. Wel een duidelijke motivatie geven in waarom je die specifieke biometrische beveiliging heb gekozen.

Meeting: Voortgang gesprek

Datum: 20-2-2017

Medewerker naam: André van ’t Hof

Testomgeving Facebook is niet mogelijk te gebruiken voor het onderscheiden van data.

In vorige meeting met de begeleider en opdrachtgever is Facebook als testomgeving gekozen. Het is niet mogelijk dataverkeer te onderscheiden van elkaar, omdat facebook gebruik maakt van HTTPS. Dit maakt het lastig de data uit te lezen zonder private key van de server. Misschien is er een mogelijkheid in deep packet inspection (DPI).

In deze meeting is aan de begeleider gevraagd in plaats van Facebook zelf een omgeving op te zetten of een andere testomgeving te gebruiken. De begeleider gaat dit met de opdrachtgever en andere collega’s bespreken. Ondertussen wordt er verder gekeken naar de optie DPI.

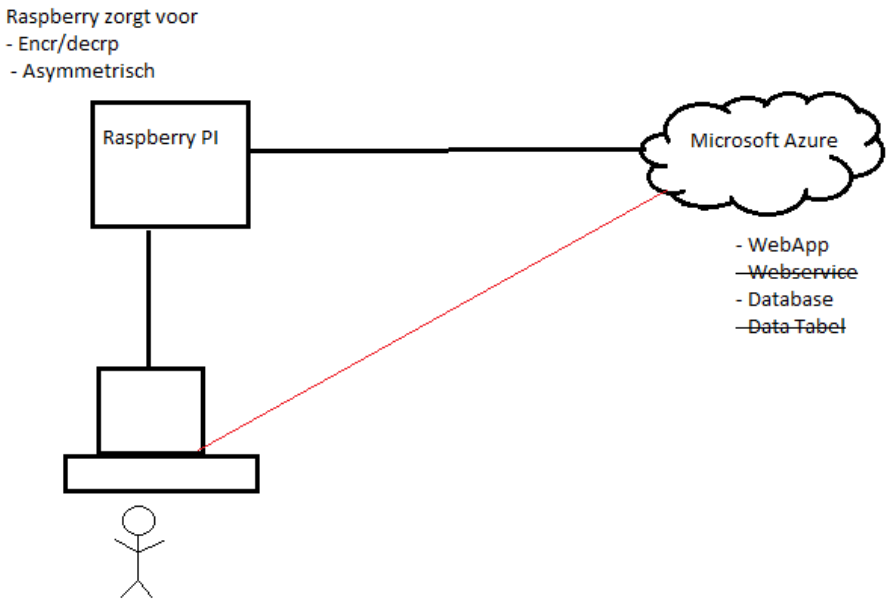
Meeting: Toegang tot Azure Cloud omgeving

Datum: 22-2-2017

Medewerker naam: Steven van den Beemt, André van ’t Hof

Wat is er nodig in de Azure Cloud omgeving?

Als eerst om de opdracht uit te leggen hebben we samen een schets gemaakt van de opdracht.



De Raspberry werkt hier als een soort proxy die ervoor zorgt dat alle dataverkeer dat erdoorheen gaat beveiligd is. De Raspberry vereist ook een two-factor authentication. Het zal waarschijnlijk een simpele webapp worden met een klein database. De webapp zou dan altijd kunnen worden uitgebreid tot iets complexer. Steven gaat ervoor zorgen dat deze ruimte op Azure beschikbaar komt, zodat ik erop kan ontwikkelen.

Meeting: Voortgangsgesprek

Datum: 28-2-2017

Medewerker naam: André van ’t Hof

Waar ben je op dit moment mee bezig?

Op dit moment ben ik bezig met een aantal encryptiemethoden naast elkaar te leggen en vergelijken met elkaar. De vergelijking maakt uiteindelijk deel uit van het definitierapport. Ik zal alvast een opzet van het definitierapport opsturen, zodat u ernaar kunt kijken.

Zijn er nog dingen waar je tegen aan loopt waardoor je nu niet meer verder kan?

Het in principe nu goed onderzoek uitvoeren. Misschien kan ik daarvoor een dagje naar de bibliotheek gaan.

Meeting: Bedrijfsbezoek

Datum: 13-3-2017

Medewerker naam: André van ’t Hof, Bart Lamot

Begeleider HHS: J.P.M de Vreught

Is het handig het biometrische gedeelte erbij te doen?

Dat is een afweging die je zelf moet maken, je kan dit wel meenemen in je analyse.

De methode staat nu op gebruik van RAD omdat RUP veel tijd vereist aan documentatie in het begin. Heeft u daar opmerkingen op.

RAD is een voorloper van IAD en is een prima methode om t gebruiken. Denk er wel bij na dat je niet 1 voor 1 de punten van RAD gaat aflopen. Kijk naar producten die echt zinnig zijn om op te leveren. Je mag ook afwijken van RAD.

Kunt u uitleggen wat ik beschrijf in het definitierapport? Op dit moment doe ik een analyse naar de onderwerpen zoals encryptie technieken, manieren voor fysieke sleutel en biometrische technieken?

In het definitierapport doe je een analyse over het onderwerp. Daarna volgen de ontwikkelrapport met een uitvoering en testen van je opdracht.

Presentatie:

- Het moet Iets zijn waarover je kan discussiëren (20 min)
- Niet over de opdracht

Meeting: Besluit MFA

Datum: 31-3-2017

Medewerker naam: André van ’t Hof , Pieter, Bart Lamot

U2F en BLE

Ik heb onderzoek gedaan naar het meest veilige multi-factor authenticatie (MFA) dat gebruikt zou kunnen worden op de Raspberry Pi 3 voor mijn project.

Uit het onderzoek wil ik het volgende gaan ontwikkelen als MFA:

U2F&BLE authenticatie

Bluetooth is op zichzelf een goed protocol voor de MFA, maar vereist veel energie.

Bluetooth Low – Energy (BLE) verbruikt veel minder energie, maar is zo “uitgekleed” dat het protocol onveilig maakt om te gebruiken voor MFA.

Het nieuwe U2F protocol wat gebruikt wordt in de Yubico’s usb sleutels is wel een sterk en veilig protocol voor MFA.

Het idee is nu deze 2 protocollen te combineren. Via een app wordt er een connectie opgezet met bluetooth low-energy en daarna wordt over de verbinding het U2F protocol gebruikt om een veilige toegang te verlenen tot de dienst.

Wanneer de gebruiker de bluetooth verbinding verbreekt dan wordt de toegang tot de dienst ontzegd.

De vingerafdrukscanner op de smartphone kan worden gebruikt om nog een biometrisch beveiliging toe te voegen.

Een aantal punten waarom dit het best is voor dit project:

- De gebruiker heeft niet de mogelijkheid de sleutel te laten liggen bij het apparaat. (bijv. usb-sleutel in de RPI3 laten)
- Een sterke MFA, maar de Usability blijft goed te gebruiken.
- BLE 4.1 heeft zijn zwakke punten, maar met het U2F protocol wordt het juist extra sterk. (De RPI3 heeft een BLE 4.1 chip)
- Een telefoon is lastiger te klonen, stelen of vergeten.
- Het is niet de allerbeste MFA, maar dit lijkt het beste voor de RPI 3 met zo min mogelijk kosten.

Dit concept bestaat al en er bestaat een enkele demo van, maar het is nog niet uitgebracht op de markt.

Als opdrachtgever van het project wil ik vragen wat je ervan vindt?

Evaluatie Beroepstaken

Aan de start van het afstudeerproject zijn er een aantal beroepstaken geselecteerd die de student moet bewijzen met behulp van een afstudeerverslag. In dit hoofdstuk wordt een evaluatie beschreven van de visie die de student heeft op deze beroepstaken.

G1 Praktische aspecten hanteren in (internationale) projecten

De beroepstaak G1 is bewezen door twee periodes lang werkzaam te zijn binnen ICT Group. Tijdens deze periode zijn er verschillende meetings ingepland met de begeleider en andere medewerkers van ICT Group. Met behulp van de meetings was het mogelijk een risicoanalyse uit te voeren en een planning op te stellen deze staan beschreven in het plan van aanpak.

A1 – Analyseren van het probleemdomein

Deze beroepstaak is bewezen door te onderzoeken wat de huidige situatie is bij ICT Group. Met behulp van gesprekken met de opdrachtgever heb ik een analyse kunnen uitvoeren. Deze analyse is aangevuld met het onderzoek dat verricht is en daarmee heb ik een lijst met eisen kunnen vaststellen. Dit proces wordt beschreven in het definitierapport.

A3 – Achterhalen van behoeften van belanghebbenden

De A3 beroepstaak is bewezen door het houden van interviews met alle belanghebbenden in het project. Dat waren in dit geval de opdrachtgever en begeleider. Door de antwoorden van de interviews om te zetten naar eisen van het project is competentie A3 bewezen. Dit wordt ook in het definitierapport beschreven.

C9 – Ontwerpen van een technische infrastructuur

In het ontwikkelrapport zijn per increment ontwerpen van het systeem gemaakt. Uiteindelijk is er meer geprogrammeerd dan dat er aanpassingen zijn gedaan aan een infrastructuur. Daarmee betwijfel ik of C9 dan ook de juiste keuze is geweest en C8 “Ontwerpen van een technisch informatiesysteem” misschien een betere keuze was geweest. Het ontwerp van het systeem wordt beschreven in het Ontwikkelrapport.

D18 – Testen van een infrastructuur

Aan de hand van de testrapportage is competentie D18 bewezen. Ook bij deze competentie betwijfel ik dat D18 voor het project de juiste keuze is geweest. Er wordt geen werking van een opgesteld infrastructuur getest, maar de opgeleverde software en scripts. De beroepstaak D17 “testen van softwaresystemen” zou misschien meer een passende beroepstaak zijn geweest voor dit project. Het proces van het testen van het systeem wordt gedaan in het Testrapport.