Security Rapport

The Pink Banker

Inhoudsopgave

Security van de Arduino

Risico's:

Wanneer de Arduino niet goed wordt beveiligt is, zal het mogelijk zijn dat iemand van buitenaf de pincode kan onderscheppen. Hierdoor gaat de beveiliging over de desbetreffende bankrekening omlaag.

Mogelijkheden:

Één van de mogelijkheden tot het beveiligen van de Arduino, is om de Arduino in een pin kast in te bouwen. Hierdoor blijft de snelheid van de Arduino gelijk en is er alleen een fysieke mogelijkheid tot het extra uitlezen van de pincode die wordt doorgestuurd via de Arduino. Dit zal kunnen door het eerst slopen van de kast om vervolgens een eigen kabeltje in de Arduino te kunnen stoppen. De pinkast zullen wij in de volgende periode bouwen en zal dus op dit moment nog niet klaar zijn voor de oplevering.

Een andere mogelijkheid voor de Arduino en de pincode is om een encryptie te gebruiken. Een voorbeeld is om gebruik te maken van "Cape Library". Cape Library is een String encryptie library voor Arduino, dit betekent dat de Strings die de data bevatten voor het pasnummer, bankrekeningnummer en pincode encrypted worden verstuurd. Wanneer er gebruik wordt gemaakt van "Cape Library", zal er meer tijd worden besteed aan het programmeren. Verder is er geen garantie dat deze encryptie er 100% zeker voor zal zorgen dat de pincode niet onderschept kan worden of dat de arduino niet gehackt kan worden. Verder kan er, wanneer er alleen encryptie gebruikt wordt, gemakkelijk een kabeltje in de Arduino gedaan worden om de pincode uit te kunnen lezen. Daarnaast zal de Arduino aanzienlijk langzamer worden indien de data ge-encrypt wordt, dit doordat de Arduino niet veel rekenkracht heeft. Tot slot is het niet zeker of dat we in staat zijn om te de-encrypten op de client. Dit aangezien er vrij weinig informatie te vinden is over wat voor encryptie techniek deze library gebruikt.

De mogelijkheid bestaat ook om beide mogelijkheden tegelijk te gebruiken. Dit houdt in dat de Arduino een encryptie zal bevatten en deze vervolgens ingebouwd zal worden. Hiermee zal de Arduino optimaal beschermt zijn, maar zal het ook inhouden dat de Arduino langzamer zal zijn.

(zie: https://github.com/gioblu/Cape)

Vergelijkend overzicht:

Security Arduino	Doorstuur snelheid	Extern uitlezen van Arduino
Arduino inbouwen	De snelheid van het doorsturen van de pincode zal gelijk blijven. De gebruiker zal er niks van merken dat de pincode langs de arduino gaat.	De pinkast zal eerst gesloopt moeten worden, voordat een ander de Arduino met een eigen laptop kan gaan uitlezen.
Encryptie gebruiken	De snelheid van het doorsturen van de pincode zal langer gaan duren.	De Arduino zal gewoon bij de client liggen en zo zal het heel makkelijk zijn voor iemand om een eigen draadje in de Arduino te doen om deze vervolgens te kunnen uitlezen

Ons advies:

Aan de hand van bovenstaande informatie is er een conclusie te trekken waaruit blijkt dat het gebruiksvriendelijker is om de Arduino in te bouwen in de pinkast. Dit doordat de gebruiker dan er geen last van heeft dat de pincode door de Arduino geencrypt moet worden en daardoor enige vertraging oploopt voordat deze verder wordt doorgestuurd. Daarnaast is het een stuk veiliger om de Arduino in te bouwen dan deze te encrypten. Dit omdat er dan eerst fysiek iets gesloopt moet worden voordat deze door iemand van buitenaf uitgelezen zou kunnen worden.

Security van de client

Risico's:

Indien de client niet goed beveiligd is zou kritieke data zoals pincodes kunnen lekken. Ook zou de pinpas kunnen worden gestolen tijdens het pin proces. Tot slot zou het kunnen gebeuren dat de pinnende persoon opeens weg moet en daardoor vergeet de sessie te sluiten waardoor een ander persoon zou kunnen pinnen met de gegevens van de vorige gebruiker.

Mogelijkheden:

Een mogelijkheid tot security van de client is om een tijdslimiet te stellen op een pin sessie. Bijvoorbeeld een limiet van twee minuten. Dit houdt in dat de gebruiker twee minuten heeft om bij een pinautomaat te doen waarvoor deze persoon is gekomen en dat na de twee minuten de pinautomaat weer terug gaat naar het inlogscherm. Mocht de gebruiker dan een keer ineens weg moeten bij de pinautomaat en vergeten de pin sessie te cancelen, zal een ander met verkeerde intenties niet zomaar er toe instaat zijn om geld te pinnen van het nog openstaande account.

Een variatie van deze mogelijkheid is om het scherm terug te laten keren naar het inlogscherm na een inactiviteit van 30 seconden. Dit houdt in dat wanneer de gebruiker gedurende 30 seconden geen actie heeft uitgevoerd op de pinautomaat, de pinautomaat automatisch terugkeert naar het inlogscherm. Zo is de gebruiker beschermt tegen misbruik van het account wanneer de persoon ineens weg is van de pinautomaat.

Een andere mogelijkheid zou zijn dat de pas van de gebruiker gedurende het gehele pin proces op de RFID scanner moet blijven liggen. Dit zorgt ervoor dat er alleen een pintransactie gedaan kan worden als de gebruiker met de pinpas bij de pinautomaat is. Een nadeel hiervan is dat de gebruiker de pas zou kunnen vergeten aan het eind van het pin proces en dat dus iemand anders er vandoor zou kunnen gaan met de desbetreffende pinpas. Daarnaast bestaat er ook een mogelijkheid dat de pinpas gestolen kan worden gedurende het pin-proces. Dit omdat de gebruiker hoogstwaarschijnlijk geen aandacht besteed aan de pinpas gedurende het pin-proces.

Vergelijkend overzicht:

Security Client	Gebruiksvriendelijkheid	Veiligheid
Pin proces van 2 min	De gebruiker heeft een tijdslimiet voor hoe lang hij heeft voor het doen van een pintransactie.	Wanneer de gebruiker ineens weg moet, blijven de 2 min optellen en zal er lastig misbruik gemaakt kunnen worden van een openstaand account
Afbreken na 30 sec inactiviteit	De gebruiker kan zo lang doen over een pintransactie als gewenst.	Wanneer de gebruiker ineens weg moet, kan een ander oneindig lang nog misbruik maken van het nog openstaande account mits de persoon acties blijft uitvoeren
Pas op de scanner laten	De gebruiker kan zolang doen over een pintransactie, maar er is een kans dat de gebruiker de pas vergeet na afronden van de transactie.	Wanneer de gebruiker ineens weg moet, kan hij de pas vergeten en kan er misbruik gemaakt worden van het openstaande account en daarnaast zal de gebruiker geen pin- transacties meer kunnen doen.

Ons advies:

Aan de hand van de bovengenoemde informatie, is er een conclusie te trekken dat het het veiligst is voor de gebruiker wanneer er een tijdslimiet van twee minuten wordt gebruikt voor een pin proces. Zo heeft de gebruiker tijd om een pintransactie te doen en is deze beschermd tegen misbruik wanneer hij ineens weg moet van de pinautomaat. Deze manier is gebruiksvriendelijker dan de manier om een pin transactie alleen te kunnen doen wanneer de pas op de RFID scanner blijft liggen. Dit vanwege de nadelen die het met zich meebrengt wanneer de pas op de scanner moet blijven liggen. De 30 seconden inactiviteit valt af vanwege het feit dat er dan nog steeds vrij veel misbruik gemaakt kan worden van een nog openstaand account op de pinautomaat.

Security van de server

Risico's:

Wanneer de server die goed beveiligd zou zijn, bestaat er een kans dat iedereen zomaar bij alle opgeslagen gegevens in de database zou kunnen. In de database bevind zich alle informatie die een bank nodig heeft, waaronder de rekeningnummers met de bijbehorende pincodes. Wanneer deze informatie in de verkeerde handen valt, is de security van de desbetreffende rekening niet gegarandeerd. Nu is het als bank zijnde zo dat je je klanten wilt kunnen garanderen dat hun geld veilig is.

Mogelijkheden:

De eerste mogelijkheid voor wanneer het aankomt op de beveiliging van de server, is om gebruik te maken van AES. AES staat voor Advanced Encryption Standard en is een encryptie methode die software efficiënt te gebruiken is. Bij deze manier van encryptie gebruiken de server en de client dezelfde key om data te encrypten. Wanneer iemand deze key weet te ontcijferen, is de data niet meer veilig. Het grootste nadeel bij deze manier van encrypten is dus dat de key altijd veilig opgeborgen moet zijn.

De tweede mogelijkheid is het gebruik van TLS. TLS staat voor Transport Layer Security en is een encryptie protocol voor het beveiligen van de communicatie tussen computer en server. Een TLS protocol is tweedelig. Het eerste deel gaat erover dat de server wordt geauthenticeerd met behulp van een op asymmetrische cryptografie berustte certificaat. Hierdoor weet een gebruiker dat server ook echt is wie hij zegt dat hij is, en dat er niet met een andere server wordt gecommuniceerd. Het tweede deel gaat over de communicatie tussen client en server. Hierbij wordt er gebruik gemaakt van symmetrische cryptografie en de key die beide partijen gebruiken zijn hetzelfde. Zo bestaat er dezelfde weakness als bij AES, maar gaat de security bij TLS weer omhoog door de eerste authenticatie van de server.

Vergelijkend overzicht:

Security Server	Gebruiksvriendelijkheid	Kwetsbaarheid
AES	AES is software efficiënt te gebruiken en gemakkelijk te implementeren in andere systemen	De key die wordt gebruikt op de server en de client zal te allen tijde veilig opgeborgen moeten worden.
TLS	Met TLS is het gemakkelijk om uit te breiden of te kunnen gaan communiceren met andere instanties	Doordat TLS zo universeel is en ook met oudere versies kan communiceren, kunnen hier lekkages optreden die gemakkelijk te overzien zijn.

Ons advies:

Aan de hand van de bovenstaande informatie is ons advies om gebruik te maken van TLS als security voor de server. Dit omdat TLS gemakkelijk te implementeren is met java. Verder is TLS gemakkelijk uitbreidbaar als er gebruik gemaakt gaat worden van een website of als er gecommuniceerd moet worden met andere instanties. Al zal er wel een certificaat gekocht moeten worden bij een vertrouwt bedrijf zodat andere instanties weten dat het een betrouwbaar certificaat is.

Tevens werd er gevraagd naar Hash, omdat men had gehoord dat andere banken deze methode gebruiken.
Een hash algoritme (zoals SHA) zet de gegeven input om in een unieke output. Deze output kan je alleen niet meer terug veranderen naar de gegeven input. Hash werkt namelijk maar één kant op. Dit is handig om te checken of de data die jij hebt hetzelfde is als de data die je zou moeten hebben. Stel je wilt een groot bestand downloaden, dan kan er wel eens wat mis gaan. Als het belangrijke data is, wil je wel graag weten of er niks mis is gegaan. Dan kan je hash gebruiken om een output te genereren (Deze output is uniek voor jouw data en kleiner dat je originele bestand) en deze kan je vergelijken met de hash die de eigenaar van het bestand heeft. Zodra er ook maar een bit anders is in 1 van de 2 bestanden zal de gehele hash anders zijn. Dit word ook gebruikt in TLS, maar omdat wij onze data ook weer moeten kunnen decrypten, is het verstandiger dat er gebruikt gemaakt wordt van TLS.

Security van de pincode en overige gevoelige data

De gevoelige data wordt tijdens de pin-sessie op de client opgeslagen in private fields. De pincode slaan we zo kort mogelijk op, alleen tijdens het inloggen en daarna word de data verwijderd. Wanneer de sessie wordt beëindigd, doordat de sessie langer duurt dan 2 minuten of omdat de gebruiker klaar is met de pintransactie, zullen alle fields worden gereset naar de standaardwaarden in java. Dit betekent dus dat alle gevoelige data meteen wordt verwijderd na het beëindigen van de sessie.

Er wordt geen gevoelige data op de pas opgeslagen, omdat de pas gebruik maakt van NFC technologie. Bij NFC wordt passief data verzonden, als je de kaart bij een scanner houd wordt direct de data van de kaart gelezen. Om deze reden kan er geen wachtwoord op de kaart worden gezet. Vandaar dat ervoor gekozen is om de rekening te koppelen aan het unieke ID (UID) van de gegeven kaart. Dit ID kan niet veilig worden veranderd. Als dit toch geprobeerd wordt, zal de gehele pas niet meer werken.

Ook op de server zelf wordt de pincode niet opgeslagen. De pincode wordt na ontvangen op de server in een query gestopt naar de database. Daar wordt gecheckt of de pincode overeen komt, als dat zo is krijgt de server de nodige gegevens terug. Nadat deze gegevens weer terug naar de client zijn verzonden, is de server de pincode alweer kwijt. Zo kan er alleen aan de pincode gekomen wanneer iemand toegang heeft tot de database.

Dataflow diagram: