# Onderzoek

Het inlogsysteem maakt gebruik van een private en public keypair. In de applicatie wordt gecontroleerd of de public key bestaat. Om dit op een veilige manier af te handelen is hiervoor een onderzoek gestart.

In de frontend heeft een gebruiker zijn private key. Het is niet veilig om een private key te versturen over het web. Er moet dus een manier zijn om te verifiëren dat iemand die zijn public key instuurt ook beschikt over de bijbehorende private key. Dat kan door een bericht te signen aan de front-end en dan dat bericht met public key sturen naar de back-end, dus zonder private key.

Als de back-end deze kan decrypten met de public key, dan weet je dat degene die het ingestuurd heeft ook beschikt over de bijbehorende private key. Dit mechanisme op zich is niet genoeg om geheel op te vertrouwen, omdat iedereen met een eigen gegenereerde keypair (volgens het juiste algoritme) dan in de applicatie komt.

Er moet namelijk nog op de blockchain gekeken worden of de public key erop bestaat. Deze twee mechanismen vormen samen het inlogsysteem.

Een van de projectleden had aan zijn baas een mogelijke oplossing gevraagd. Deze cyber security expert kwam met de volgende oplossing. Het idee was om een bestand te encrypten waar de private key in staat. Met de private key wordt dan een bericht gesigned en samen met de public key opgestuurd naar de backend.

Om een bestand de encrypten is onderzoek gedaan naar welke algoritmes mogelijk zijn en welke het beste is. Hierbij is AES, RSA , TwoFish en 3Des geanalyseerd. Uit deze analyse is naar voren gekomen dat 3Des langzaam maar zeker minder wordt omdat er betere algoritmes zijn bedacht. Daarnaast is 3Des qua algoritmes traag. AES is een van de meest veilige manieren van encryptie omdat het in rondes gewoon text vertaald naar cipher tekst.

Door de bovengenoemde analyse is gekozen om AES-256 te gebruiken voor het encrypten van het bestand.[2]

# Gemaakte keuzes

Om het inloggen veilig te maken zijn de volgende keuzes gemaakt:

* Private key encrypten zodat het veilig opgeslagen kan worden op de computer
* Passphrase gekozen om het bestand te encrypten voor gebruiksvriendelijkheid
* AES-256 algoritme gekozen omdat deze het meest veilig is volgens de referenties
* Afgeweken van de al bestaande code omdat het is refactored.
* Geen gebruik gemaakt van mnenomic omdat het niet te herleiden was met het key pair

# Mogelijke uitbreidingen(aanbevelingen)

Om het inloggen nog beter te maken is het mogelijk om Captcha te implementeren. Hiermee wordt dan bruteforcing door bots tegen gegaan.

Een andere verbetering op het inlogsysteem is het toepassen van padding op de passphrase. Op het moment is het alleen mogelijk om een passphrase te gebruiken van 16,32,64 enz bits. Padding is het aanvullen van een zin met random bits om zo op het juiste aantal bits te komen.