Fully Homomorphic Encryption

Onderzoeksrapport

V1.0

2 april 2021

Salt Cyber Security

Teun Westbroek

Jan de Groot

Opgesteld door:

Thomas van den Nieuwenhoff

Redacted

Redacted

Samenvatting

SALT Cyber Security adviseert en ondersteunt software-ontwikkelteams bij klanten, met het bouwen van veilige software. Het bedrijf ziet een probleem bij traditionele encryptievormen, waar de opkomende technologie Fully Homomorphic Encryption (FHE) de oplossing voor zou moeten zijn. Data moet altijd eerst ontcijferd worden, voordat er een berekening mee gedaan kan worden. In dit onderzoek wordt onderzocht of FHE daadwerkelijk de oplossing kan bieden voor dit probleem. Dit wordt gedaan door antwoord te geven op de volgende vraag: “Voor welke bedrijfstoepassingen uit de hedendaagse praktijk is Fully Homomorphic Encryption geschikt?”

Dit onderzoek is methodisch uitgevoerd door de onderzoeksmethode *literature study* toe te passen. Door middel van zoektermen is informatie opgevraagd uit verschillende zoekmachines en databanken.

FHE is een techniek waarbij gegevens in cijfertekst kunnen worden omgezet, terwijl ze geanalyseerd kunnen worden alsof ze nog in de oorspronkelijke vorm zijn. De techniek ondersteund in principe alle mogelijke operaties op data, maar is momenteel nog beperkt tot optellingen en vermenigvuldigingen. Doordat gegevens in versleutelde vorm verwerkt kunnen worden, kan de verwerking plaatsvinden op hardware van een onbetrouwbare partij. Dit biedt nieuwe kansen voor het genereren van inkomsten uit datasets. Helaas zijn de prestaties van FHE momenteel zeer slecht door inefficiënties. Ook het te groot worden van cryptografische ruis is nog een probleem. De webbrowsers Google Chrome en Microsoft Edge gebruiken al vormen van homomorfe encryptie. Het wordt gebruikt voor het opslaan van wachtwoorden en controleren of deze een keer gelekt zijn. De techniek zou ook gebruikt kunnen worden door een ziekenhuis om patiëntgegevens te delen met andere instellingen, zonder de privacy in het geding te brengen.

De conclusie is daarom dat FHE geschikt is voor bedrijfstoepassingen waar de verwerking van data individueel versleutelde getallen betreft welke opgeteld of vermenigvuldigd moeten worden, de verwerkingstijd hoog mag zijn en het niet nodig is het resultaat oneindig vaak te gebruiken voor nieuwe berekeningen. Voorbeelden hiervan zijn wachtwoordmonitoring, stemmachines, Internet of Things en het delen van bedrijfsdata met derde partijen.

Inhoud

1. Inleiding

1.1 Aanleiding

SALT Cyber Security adviseert en ondersteunt software-ontwikkelteams bij klanten, met het bouwen van veilige software. Vanuit die context volgt SALT ook de ontwikkelingen van Fully Homomorphic Encryption (FHE) en spelen er ook actuele klantvragen over dit onderwerp. Voor SALT is het dus van groot belang om de huidige status van FHE, de toepassing ervan, maar ook de nabije toekomst goed te snappen. Zodat op die manier hun klanten goed geadviseerd en uiteindelijk ook begeleid kunnen worden in het implementeren van FHE.

1.2 Afbakening

In dit onderzoek wordt voor SALT gekeken of FHE al geschikt is om toe te passen en welke bedrijfstoepassingen hiervoor geschikt zouden zijn. Er wordt specifiek onderzocht of dit voor de volledig homomorfe vorm geldt, niet alle andere vormen. Ook worden er geen voorspellingen gedaan over de toekomst van de technologie. Er wordt dus echt gekeken naar FHE in zijn huidige vorm.

1.3 Organisatie

SALT is een start-up van net een jaar oud. Het bedrijf is ontstaan door een idee van de Managing Director en oprichter Teun Westbroek. Hij merkte op dat hij vaak beveiligingsproblemen in software aan het oplossen was, terwijl dit niet echt bij zijn werk hoorde. Teun trok de conclusie dat er eigenlijk geen bedrijven waren die hun klanten begeleiden in het veiliger maken van hun applicaties. Toen heeft hij SALT Cyber Security opgericht. Het bedrijf is gevestigd in Amsterdam en heeft naast Teun een directe werknemer. Het bedrijf valt onder de Cronos groep, dus ze werken veel samen met andere bedrijven binnen de groep.

1.4 Huidige kennis

Er zijn veel technische papers over dit onderwerp te vinden (Armknecht, et al., 2015; Brakerski, Gentry, & Vaikuntanathan, 2011; Brakerski & Vaikuntanathan, 2011) en ook wel wat mogelijke toepassingen (Archer, et al., 2017; Bel Korchi & El Mrabet, 2019), maar nog weinig mensen hebben deze punten echt samen laten komen om te kunnen concluderen of FHE al geschikt is om te gebruiken (Arampatzis, 2020; Armknecht, et al., 2015; Huynh, 2020; Will & Ko, 2015).

1.5 Relevantie

Hoe praktisch FHE is en welke bedrijfstoepassingen het goed kunnen gebruiken, is interessant om te onderzoeken omdat dit het informatielandschap een stuk veiliger kan maken en de privacy kan waarborgen. FHE als principe wordt gezien als de “heilige graal” van encryptie (Will & Ko, 2015), dus een zeer interessant onderwerp voor de toekomst van cryptografie.

1.6 Onderzoeksvragen

De hoofdvraag van dit onderzoek luidt:

Voor welke bedrijfstoepassingen uit de hedendaagse praktijk is Fully Homomorphic Encryption geschikt?

Om hierop antwoord te kunnen geven, worden in dit onderzoek de volgende deelvragen gesteld:

1. “Wat is Fully Homomorphic Encryption?”
2. “Wat zijn de voor- en nadelen van Fully Homomorphic Encryption?”
3. “In hoeverre maken bedrijfstoepassingen momenteel gebruik van Fully Homomorphic Encryption?”

1.7 Onderzoeksopzet

Voor dit onderzoek is de onderzoeksmethode *literature research* gebruikt. Hierbij wordt met zoektermen informatie opgevraagd uit verschillende zoekmachines en databanken. Vervolgens wordt informatie geselecteerd om te gebruiken in dit rapport. Met de voorwaarde dat er een goed geëvenaard antwoord op de hoofdvraag gegeven kan worden.

1.8 Leeswijzer

Allereerst is in hoofdstuk 2 de methodebeschrijving te vinden waarin benoemd wordt hoe dit onderzoek methodisch wordt aangepakt. In hoofdstuk 3 worden de resultaten van het onderzoek weergegeven. In paragraaf 3.1 worden de resultaten van de eerste deelvraag getoond, in paragraaf 3.2 de tweede en in paragraaf 3.3 de derde. Vervolgens wordt in hoofdstuk 4 de conclusie geformuleerd, in hoofdstuk 5 de discussie en in hoofdstuk 6 is de literatuurlijst te vinden.

2. Methodebeschrijving

In dit onderzoek is een keuze gemaakt uit verschillende onderzoeksmethodes om tot een goed antwoord op de hoofdvraag te kunnen komen. Deze methodes staan beschreven op hboictresearchmethods.nl (Bonestroo, et al., 2018).

2.1 Literature study

Om algemene informatie, begeleiding en best practices te vinden, worden met relevante sleutelwoorden gerelateerde bronnen geraadpleegd voor informatie. Binnen een bron wordt naar interessante verwijzingen en nieuwe sleutelwoorden gezocht. Hiermee wordt het zoekproces herhaald. De sleutelwoorden worden toegepast op een aantal zoekmachines, namelijk:

- Google: https://google.com/
- Google Scholar: https://scholar.google.nl/
- DuckDuckGo: https://duckduckgo.com/
- WindeSearch: https://mediacentrumwindesheim.nl/windesearch/

Tenslotte wordt geselecteerd welk materiaal in detail gelezen en gebruikt wordt in dit onderzoeksrapport.

Zoektermen Wat is Fully Homomorphic Encryption?

Fully homomorphic encryption, fully homomorphic encryption definition.

Zoektermen Wat zijn de voor- en nadelen van Fully Homomorphic Encryption?

Fully homomorphic encryption.

Zoektermen In hoeverre maken bedrijfstoepassingen momenteel gebruik van Fully Homomorphic Encryption?

Fully homomorphic encryption, homomorphic encryption pilot, who uses homomorphic encryption, homomorphic encryption real world.

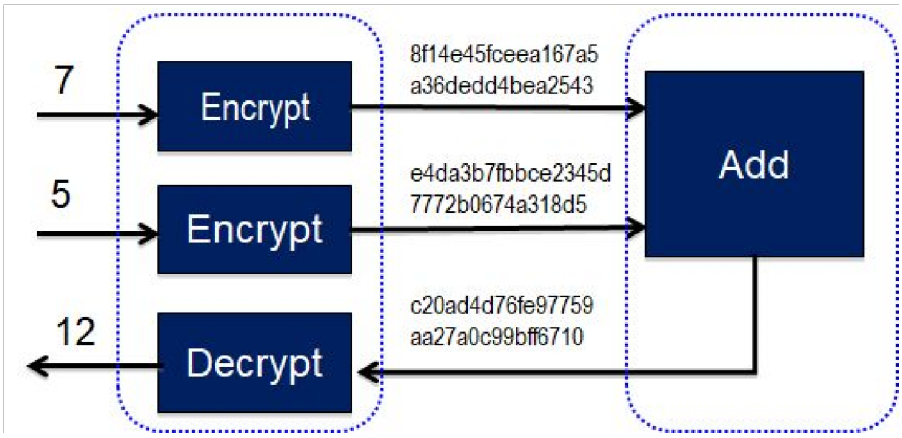
3. Resultaten

3.1 Wat is Fully Homomorphic Encryption?

Definitie

Homomorphic Encryption (Homomorfe Encryptie): de omzetting van gegevens in cijfertekst die kunnen worden geanalyseerd en verwerkt alsof ze nog in de oorspronkelijke vorm zijn (Wikipedia contributors, 2021).

In de wiskunde beschrijft homomorf de transformatie van de ene dataset in een andere, terwijl de relaties tussen elementen in beide sets behouden blijven (TechTarget Contributor, 2011). Anders gezegd maakt Homomorphic Encryption (HE) het dus mogelijk om (bepaalde) berekeningen te maken op versleutelde tekst (Wikipedia-bijdragers, 2019). Dit proces is weergegeven in Figuur 1. Traditioneel gezien, is het onmogelijk om met versleutelde data te werken. Het moest namelijk altijd ontcijferd worden voordat het gebruikt werd. Versleutelde data zijn namelijk expres onleesbaar gemaakt.



Figuur 1. Homomorphic Encryption processing diagram. Overgenomen uit *Building Applications with Homomorphic Encryption* door Hallman et al. (2018).

HE kan verschillende vormen van encryptie gebruiken, die verschillende soorten computaties uit kunnen voeren (Wikipedia contributors, 2021). Welke vorm van encryptie gebruikt wordt, dicteert of de encryptie volledig homomorf is, of maar deels. Het resultaat van een berekening met versleutelde data is overigens ook versleuteld, maar komt na ontcijfering overeen met het resultaat als dezelfde berekening was uitgevoerd op de oorspronkelijke tekst (Wikipedia-bijdragers, 2019).

Nu bekend is wat HE is, kan het verschil tussen Fully (FHE) en Partial Homomorphic Encryption (PHE) beschreven worden.

Wanneer de encryptie volledig homomorf is (ook wel de “heilige graal” van encryptie genoemd), kan het voor elk gewenst doel gebruikt worden (Armknecht, et al., 2015; Will & Ko, 2015; Wikipedia contributors, 2021). PHE kan slechts gebruikt worden voor optellingen óf multiplicaties, niet beide (Will & Ko, 2015). FHE ondersteund wél beide en het heeft de potentie om nog veel meer te ondersteunen. Naast FHE en PHE zijn

er nog een aantal andere categorieën van HE. Zo is er Somewhat HE (SWHE) en Leveled Fully HE (LFHE). Dit zijn verschillende combinaties van HE waarin extra functionaliteit is toegevoegd.

Geschiedenis

HE heeft al een behoorlijke geschiedenis en is op verschillende manieren aangevlogen (Wikipedia contributors, 2021). Er is jarenlang geprobeerd om FHE werkend te krijgen, zonder succes (Vaikuntanathan, 2012). Totdat Craig Gentry bij IBM in oktober 2008 de eerste volledig homomorfe encryptie had bedacht.

Pre-FHE

Voor zover men weet, bestaat het idee dat homomorfe encryptie gebruikt kan worden voor het beschermen van data al decennia (Will & Ko, 2015). Het probleem van het vormen van een volledig homomorfe encryptie, werd voor het eerste voorgesteld in 1978 (Rivest, Adleman, & Dertouzos, 1978). Er werden speciale encryptie functies genaamd “privacy homomorphisms” voorgesteld. De auteurs bespreken het gebruiken van hardware om data veilig te verwerken. De data zouden dan alleen ontcijferd worden op een fysiek veilige processor. Op deze manier wordt de data altijd versleuteld wanneer het de processor verlaat (bv. wanneer het naar het geheugen wordt verplaatst). Het probleem met deze soort chips, is dat het op maat gemaakte hardware is, duur om te implementeren en altijd nog een decoderingssleutel nodig heeft. Voor meer dan 30 jaar was het onduidelijk of er een oplossing bestond. In die periode zijn er een aantal gedeeltelijke oplossingen verschenen (Wikipedia contributors, 2021):

- RSA-cryptosysteem (onbeperkt aantal modulaire vermenigvuldigingen);
- ElGamal cryptosysteem (onbeperkt aantal modulaire vermenigvuldigingen);
- Goldwasser-Micali cryptosysteem (onbeperkt aantal XOR-operaties);
- Benaloh cryptosysteem (onbeperkt aantal modulaire toevoegingen);
- Paillier cryptosysteem (onbeperkt aantal modulaire toevoegingen);
- Sander-Young-Yung-systeem (loste na meer dan 20 jaar het probleem op voor logaritmische diepteschakelingen) (Sander, Young, & Yung, 1999, pp. 554-566);
- Boneh-Goh-Nissim cryptosysteem (onbeperkt aantal optelbewerkingen, maar maximaal één vermenigvuldiging) (Boneh, Goh, & Nissim, 2005);
- Ishai-Paskin cryptosysteem (vertakkingsprogramma’s van polynoomgrootte) (Ishai & Paskin, 2007).

Eerste generatie FHE

Craig Gentry beschreef de eerste plausibele constructie voor een volledig homomorfe encryptie met behulp van op roosters gebaseerd cryptografie[1] (Gentry C. , 2009; Wikipedia contributors, 2021). Gentry’s constructie ondersteunt zowel optel- als

[1] Zie voor een goede uitleg van op roosters gebaseerde cryptografie de Wikipediapagina ‘Lattice-based cryptography’.

vermenigvuldigingsbewerkingen op versleutelde teksten, van waaruit het mogelijk is om schakelingen te maken voor het uitvoeren van willekeurige berekeningen. De constructie volgt een aantal stappen waarin eerst ruis[2] wordt geïntroduceerd en vervolgens wordt gereduceerd. Hierdoor is het mogelijk om een willekeurig aantal optellingen en vermenigvuldigen te berekenen zonder de ruis te veel te verhogen. De Gentry-Halevi implementatie van Gentry's originele cryptosysteem rapporteerde een timing van ongeveer 30 minuten per simpele bit operatie (Gentry & Halevi, Implementing Gentry's Fully-Homomorphic Encryption Scheme, 2011). Uitgebreide ontwerp- en implementatiewerkzaamheden in de daaropvolgende jaren, hebben de runtime-prestaties van deze vroege implementaties met vele ordes van grootte verbeterd.

In 2010 presenteerde Marten van Dijk, Craig Gentry, Shai Halevi en Vinod Vaikuntanathan een tweede volledig homomorf encryptieschema (Van Dijk, Gentry, Halevi, & Vaikuntanathan, 2009), wat veel van de tools van Gentry's constructie gebruikt.

Tweede generatie FHE

De homomorfe cryptosystemen die momenteel worden gebruikt, zijn afgeleid van technieken die vanaf 2011-2012 zijn ontwikkeld door Zvika Brakerski, Craig Gentry, Vinod Vaikuntanathan en anderen (Wikipedia contributors, 2021). Deze innovaties leidden tot de ontwikkeling van veel efficiëntere enigszins en volledig homomorfe cryptosystemen. Deze omvatten:

- Het Brakerski-Gentry-Vaikuntanathan-schema (Brakerski, Gentry, & Vaikuntanathan, 2011), voortbouwend op technieken van Brakerski-Vaikuntanathan (Brakerski & Vaikuntanathan, 2011);
- Het op NTRU[3] gebaseerde schema van Lopez-Alt, Tromer en Vaikuntanathan (LTV) (Lopez-Alt, Tromer, & Vaikuntanathan, 2013);
- Het Brakerski/Fan-Vercauteren-schema (Fan & Vercauteren, 2012), wat voortbouwt op Brakerski's schaal-invariante cryptosysteem (Brakerski, 2012);
- Het op NTRU[3] gebaseerde schema van Bos, Lauter, Loftus en Naehrig (BLNN) (Bos, Lauter, Loftus, & Naehrig, 2013), voortbouwend op het schaal-invariante cryptosysteem van LTV en Brakerski (Brakerski, 2012).

De beveiliging van de meeste van deze schema's, is gebaseerd op de hardheid van het (Ring) Leren Met Fouten (Engels: Ring Learning With Errors of RLWE) probleem. Behalve de LTV- en BLLN-schema's, want die vertrouwen op een overbelaste

[2] Door de cryptografische bewerkingen bevat elke cijfertekst een hoeveelheid ruis (Wikipedia contributors, 2021). Deze ruis groeit wanneer cijferteksten bij elkaar opgeteld of met elkaar vermenigvuldigd worden, totdat de ruis de cijfertekst uiteindelijk niet meer ontcijferbaar maakt.

[3] NTRU is een open source public-key cryptosysteem wat op roosters gebaseerde cryptografie gebruikt om gegevens te coderen en decoderen (Wikipedia contributors, 2020).

(Albrecht, Bai, & Ducas, 2016) variant van het NTRU-rekenprobleem[3] (Cheon, Jeong, & Lee, 2016). Deze NTRU-variant bleek vervolgens kwetsbaar te zijn voor sub-veldroosteraanvallen, en daarom worden deze twee schema's in de praktijk niet meer gebruikt.

Alle cryptosystemen van de tweede generatie volgen nog steeds de basisblauwdruk van de oorspronkelijke constructie van Gentry. Ze construeren namelijk eerst een ietwat homomorf cryptosysteem en converteren het vervolgens naar een volledig homomorf cryptosysteem met behulp van 'bootstrapping'.

Een onderscheidend kenmerk van de tweede generatie cryptosystemen is dat ze allemaal een veel langzamere groei van de ruis vertonen tijdens de homomorfe edit-berekeningen. Een ander kenmerk is dat ze efficiënt genoeg zijn voor veel toepassingen, zelfs zonder 'bootstrapping' aan te roepen, in plaats daarvan opereren ze in de 'leveled' FHE-modus.

Derde generatie FHE

In 2013 stelden Craig Gentry, Amit Sahai en Brent Waters (GSW) een nieuwe techniek voor, voor het bouwen van FHE-schema's die een dure stap van 'relinearisering' bij homomorfe vermenigvuldigingen vermijden (Gentry, Sahai, & Waters, 2013). Zvika Brakerski en Vinod Vaikuntanathan merkten op dat het GSW-cryptosysteem, voor bepaalde typen schakelingen, een nog langzamere groeisnelheid van ruis vertoont (Brakerski & Vaikuntanathan, 2013). Dit geeft dus een betere efficiëntie en sterkere beveiliging. Jacob Alperin-Sheriff en Chris Peikert beschreven vervolgens een zeer efficiënte 'bootstrapping'-techniek op basis van deze observatie (Alperin-Sheriff & Peikert, 2014).

Deze technieken werden verder verbeterd om efficiënte ringvarianten van het GSW-cryptosysteem te ontwikkelen. Hier kwamen het FHEW- (Ducas & Micciancio, 2014) en TFHE-schema (Carpov, Chillotti, Gama, Georgieva, & Izabachene, 2016) uit voort. Het FHEW-schema was het eerste wat aantoonde dat door de versleutelde teksten na elke bewerking te vernieuwen, het mogelijk is om de 'bootstrapping'-tijd terug te brengen tot een fractie van een seconde. FHEW introduceerde een nieuwe methode om Booleaanse poorten op versleutelde gegevens te berekenen. Hiermee werd 'bootstrapping' aanzienlijk vereenvoudigt en werd een variant van de 'bootstrapping'-methode geïmplementeerd (Alperin-Sheriff & Peikert, 2014). De efficiëntie van FHEW werd verder verbeterd door het TFHE-schema, wat een ringvariant van de 'bootstrapping'-procedure (Gama, Izabachène, Nguyen, & Xie, 2014) implementeert met behulp van een methode die vergelijkbaar is met die in FHEW.

Vierde generatie FHE

Het CKKS-schema (Cheon, Kim, Kim, & Song, 2017) ondersteunt efficiënte afrondings-bewerkingen in versleutelde toestand (Wikipedia contributors, 2021). De afrondings-bewerking regelt de toename van de ruis bij gecodeerde vermenigvuldiging, waardoor het aantal 'bootstrapping' in een schakeling wordt verminderd. In Crypto2018 is op CKKS gefocust als een oplossing voor versleutelde machine learning. Dit komt door een kenmerk van het CKKS-schema dat geschatte waarden codeert in plaats van exacte waarden. Wanneer computers gegevens met een reële waarde opslaan, onthouden ze waarden bij benadering met lange significante bits, niet exact met werkelijke waarden. Het CKKS-schema is ontworpen om efficiënt om te gaan met de fouten die voortvloeien uit de benaderingen. Het schema is bekend bij machine learning, wat inherent ruis in zijn structuur heeft. Ieder schema heeft zo zijn sterke en zwakke punten.

Doordat CKKS met afgeronde waarden werkt, in plaats van exacte is het mogelijk dat hierdoor kwetsbaarheden ontstaan. Een artikel uit 2021 van Baiyu Li en Daniele Micciancio bespreekt passieve aanvallen tegen CKKS (Li & Micciancio, 2020). De auteurs passen de aanval toe op vier moderne homomorfe encryptie libraries (HEAAN, SEAL, HElib en PALISADE) en melden dat het mogelijk is om de geheime sleutel te herstellen in verschillende parameter-configuraties. Volgens het artikel zou een mogelijke beperking, het bijwerken van het encryptiealgoritme van CKKS vereisen om te voorkomen dat de encryptieruis van een versleutelde tekst wordt achterhaald.

Techniek

Er is een *evaluatie sleutel* nodig om functies los te laten op FHE (Armknecht, et al., 2015). Daarnaast is een geheime sleutel waarmee de data ontcijferd kan worden. Deze twee operaties worden beveiliging technisch gezien dus gescheiden door twee verschillende sleutels.

3.2 Wat zijn de voor- en nadelen van Fully Homomorphic Encryption?

Als de "heilige graal" van encryptie heeft FHE natuurlijk een aantal sterke voordelen, maar het heeft ook een aantal nadelen.

Operaties

Het grootste voordeel van FHE is dat het meerdere operaties (momenteel alleen optellingen en vermenigvuldigingen) ondersteund op versleutelde data (Will & Ko, 2015). Dit was voor FHE onmogelijk, dus een groot voordeel. Dit bespaart niet alleen de moeite van iets moeten ontcijferen voordat het gebruikt kan worden, maar de veiligheid van de data wordt verhoogd. Wanneer data nooit naar een leesbare vorm gebracht hoeft te worden, kan alleen de versleutelde vorm gestolen worden. Dit dus allemaal terwijl er wel met de data gewerkt kan worden. Er zijn echter momenteel beperkingen met betrekking tot het ondersteunen van een breed scala aan

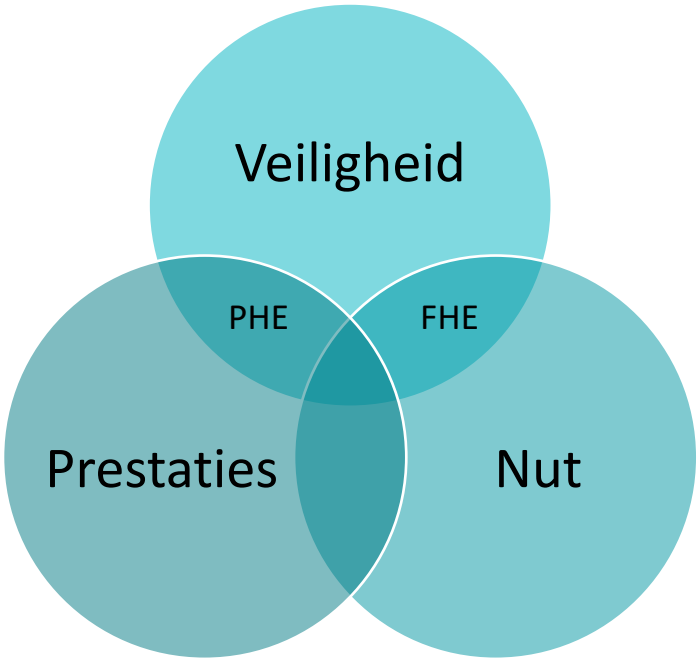
bewerkingen/functies, ook al zouden deze wel mogelijk moeten zijn (Will & Ko, 2015). Dit komt doordat twee bewerkingen kunnen worden gebruikt om elkaar op te heffen en de beveiliging zinloos te maken. Daarnaast wordt FHE momenteel gezien als de perfecte oplossing. Het moet echter per applicatie worden bekeken. Een one-size-fits-all oplossing zal niet zo veilig zijn als een schema wat is ontworpen voor de toepassing in gedachten. In de toekomst bestaat de mogelijkheid dat deze operaties uitgebreid worden. Daarnaast kan met FHE versleutelde data altijd nog ontcijferd worden om andere operaties op los te laten. Dit is alleen niet erg praktisch.

Onbetrouwbare verwerkers

Een voordeel wat aansluit op de vorige, is het kunnen inzetten van de cloud voor operaties die normaal alleen op eigen hardware worden uitgevoerd (Will & Ko, 2015). Denk hierbij aan het verwerken van zeer gevoelige gegevens. De verwerker kiest er dan waarschijnlijk voor om de data en de verwerking daarvan niet bij een externe partij neer te leggen. Het risico is in deze situatie dat een cloud provider de ontcijferde data lekt, omdat deze ergens in het proces niet meer versleuteld is. FHE staat de verwerker toe om zijn data veilig in de cloud op te slaan én te verwerken. Dit opent de deur om gebruik te maken van alle voordelen van de cloud provider.

Prestaties

Helaas zijn de prestaties van FHE momenteel niet bepaald efficiënt (Will & Ko, 2015). Gentry en Halevi schrijven in hun paper uit 2011 dat simpele operaties seconden tot uren kunnen duren, afhankelijk van de veiligheidsgraad. Ko en Will verwijzen naar deze resultaten in hun boek uit 2016. Het lijkt er echter niet op dat het prestatieprobleem sindsdien is opgelost. In een paper van Atiquzzaman, Khalil, Rahman en Yi uit 2020 wordt nog steeds een groot verschil tussen het verwerken van ontcijferde en homomorf versleutelde data beschreven. Ook de blogpost van Ravital Solomon (2020) omschrijft problemen rondom de efficiëntie van FHE. Daarom is HE momenteel nog een evenwichtsoefening tussen nut, veiligheid en prestaties. FHE is veilig, maar heeft suboptimale prestaties. In Figuur 2 is met een venndiagram de combinatie tussen de verschillende factoren weergegeven. Zelfs wanneer er in de toekomst een extreem efficiënte FHE wordt gevonden, blijven er problemen (Armknecht, et al., 2015). Het is bijvoorbeeld niet mogelijk om een operatie op versleutelde data af te breken. Het volledige proces moet doorlopen worden, wat veel tijd in beslag neemt.



Figuur 2. Veiligheid versus nut versus prestaties voor homomorfe encryptie. Aangepast overgenomen uit *A guide to homomorphic encryption* door R. K. Ko en M. A. Will, 2016, Amsterdam: Syngress. Copyright 2015, Elsevier Inc.

Compromis

Weer een voordeel is dat FHE precies tussen bestaande oplossingen kan vallen. Neem bijvoorbeeld Dropbox (Will & Ko, 2015), een aanbieder van cloudopslag. Het bedrijf gebruikt moderne encryptie methodes om zowel data te transporteren als op te slaan (SSL en AES-256 bit encryptie) en is het voor hun medewerkers verboden om de inhoud van de bestanden van gebruikers te bekijken (Dropbox Inc, z.d.). Alhoewel de webpagina waar Dropbox dit vermeldt niet meer bestaat, zal dit niet zozeer veranderd zijn. Het is duidelijk dat het desondanks het versleutelen van data, nog steeds voor iemand anders mogelijk is om naar andermans data te kijken in een leesbare vorm. Een andere aanbieder op het gebied van cloudopslag is MEGA (z.d.). Zij bieden privacy en bescherming door alleen het apparaat van de eindgebruiker gegevens te laten versleutelen of ontcijferen. Dit zorgt ervoor dat alleen de gebruiker toegang heeft tot de niet-versleutelde gegevens. Het nadeel van deze service is dat het lastig is om zoekfuncties en het delen van bestanden aan te bieden. FHE kan ervoor zorgen dat een cloudoplossing veiliger wordt, terwijl de extra functionaliteiten bewaard blijven. Of FHE ooit zoekfunctionaliteit zal ondersteunen is nog onbekend, de tijd zal het leren. Alhoewel FHE nog geen zoekfuncties ondersteund, is het ideaal voor het berekenen van wiskundige functies op versleutelde data.

Volwassenheid

HE is nog relatief jong en wordt niet snel opgenomen door de IT-wereld (Will & Ko, 2015). Alhoewel FHE misschien nog niet klaar is om in de praktijk gebruikt te worden, zou PHE gebruikt kunnen worden als tussenoplossing. Tegen de tijd dat FHE ver genoeg doorontwikkeld is, kan er overgeschakeld worden.

Dataverzameling

Doordat HE gebruikersinformatie/-gegevens beschermt, wordt voorkomen dat cloud services informatie over hen verzamelen (Will & Ko, 2015). Dit kan het einde van gerichte advertenties betekenen, maar ook het kunnen verkopen van anonieme gebruikersgegevens en vele andere manieren waarop cloud services geld verdienen. Het probleem is dat, hoewel gebruikers online veiliger willen zijn, ze misschien niet bereid zijn om te betalen voor een service.

Meerdere gebruikers

Ook heeft FHE geen ondersteuning voor meerdere gebruikers (Armknecht, et al., 2015). Een database zou bijvoorbeeld maar één klant kunnen ondersteunen, omdat de data van andere klanten niet afgeschermd zou kunnen worden. Elke gebruiker zou zijn eigen database moeten krijgen, welke versleuteld is met een eigen sleutel. Alhoewel er voortgang wordt geboekt op dit gebied (Lopez-Alt, Tromer, & Vaikuntanathan, 2013), is het nog niet beschikbaar om toe te passen.

Ruis

Een ander probleem van FHE is het te groot worden van ruis (Armknecht, et al., 2015; Solomon, 2020). Telkens wanneer er een calculatie wordt gedaan met een versleutelde waarde, groet de hoeveelheid ruis in die waarde. Er is een punt waarop de versleutelde waarde niet meer gebruikt kan worden of ontcijferd kan worden. Dit heeft tot gevolg dat er bijgehouden moet worden hoeveel ruis de versleutelde data bevat en hoe vaak deze dus nog gebruikt kan worden. Een oplossing voor het ruisprobleem is ‘bootstrapping’, maar is lang niet de perfecte oplossing.

Post-kwantum cryptografie

Alle vormen van FHE gebruiken een specifiek type cryptografie die bestand is tegen aanvallen van een kwantumcomputer (Solomon, 2020). In het Nederlands noemen we dit op roosters gebaseerde cryptografie, maar er wordt vaak gesproken over het Engelse ‘lattice cryptography’. Mocht dat als een risico worden gezien, is het natuurlijk een voordeel.

Verschillende implementaties

Er zijn al veel verschillende implementaties van FHE beschikbaar, wat op zich iets goeds is (bv.: HElib, SEAL en FHEW) (Wikipedia contributors, 2021). Het probleem is alleen dat de verschillen en relaties tussen deze schema’s slecht gedocumenteerd is, en er veel tijd en energie geïnvesteerd moet worden om een schema te begrijpen en te kunnen gebruiken (Solomon, 2020). Vervolgens kan deze kennis grotendeels niet gebruikt worden om een andere implementatie te leren.

Bruikbaarheid

FHE is niet beginner-vriendelijk, niet gebruikersvriendelijk en al helemaal niet non-cryptograaf-vriendelijk (Solomon, 2020). De meeste FHE-implementaties vereisen

expertise op het gebied van het onderliggende cryptografische schema. Solomon omschrijft nog de oorzaken van deze onvriendelijkheden in haar blogpost.

Data-inkomsten genereren

Omdat berekeningen worden gedaan op versleutelde gegevens, kunnen organisaties gevoelige bedrijfsgegevens delen voor analyse of brancheoverstijgende samenwerking zonder toegang te geven tot de privégegevens of deze openbaar te maken (IBM, z.d.).

Regelgeving

Naarmate de straffen voor overtredingen van bijvoorbeeld privacyregelgeving toenemen, kan FHE helpen bij het verwerken van versleutelde gegevens zonder ooit onversleutelde en gevoelige informatie openbaar te maken (IBM, z.d.). Datalekken worden ineens een veel minder groot risico, omdat er niet meer met leesbare tekst gewerkt wordt.

3.3 In hoeverre maken bedrijfstoepassingen momenteel gebruik van Fully Homomorphic Encryption?

Huidige toepassingen

Er zijn niet veel organisaties die nu al gebruik maken van FHE. Een aantal profiteren van PHE als tussenoplossing. CryptDB, Helios Voting en Intuit (vroeger: Porticor) zijn hier een voorbeeld van.

IBM zegt in twee gevallen FHE geïmplementeerd te hebben. Een keer in 2019 bij de Braziliaanse bank Banco Bradesco (Masters, et al., 2019) en een tweede keer bij een Europese bank (onderzoekspaper is nog niet gepubliceerd) (Salter, 2020). De Braziliaanse bank wilde met machine-learning voorspellen wat de kans was dat iemand in de komende drie maanden een lening aan zou gaan vragen. FHE is hier aantrekkelijk omdat er met privé financiële gegevens wordt gewerkt. Ten eerste versleutelden ze de gegevens en het model homomorf (Moskvitch, 2020). Ze toonden aan dat het mogelijk was om voorspellingen uit te voeren met dezelfde nauwkeurigheid als zonder versleuteling. Dit betekent dat banken het uitvoeren van voorspellingen veilig kunnen uitbesteden aan een niet-vertrouwde omgeving. Vervolgens hebben ze het model getraind met behulp van gecodeerde gegevens, waarmee ze aantoonden dat het mogelijk was om homomorfe encryptie te gebruiken om de privacy van gegevens te behouden.

Waarschijnlijk de meest gebruikte bedrijfstoepassing is momenteel de wachtwoordmonitoring functie van Google Chrome (Pullman, Thomas, & Bursztein, 2019) en Microsoft Edge (Lauter, Kannepalli, Laine, & Moreno, 2021). De browsers bieden de functionaliteit van een wachtwoordkluis aan en zijn er dus bij gebaat deze informatie goed te beschermen. Er zitten verschillen in de implementaties van de twee browsers, maar in grote lijnen doen ze hetzelfde. In eerste instantie worden alle

wachtwoorden (en gebruikersnamen) gehasht en met een traditionele encryptievorm onleesbaar gemaakt. Vervolgens wordt de informatie homomorf versleuteld. Edge gebruikt hiervoor een methode die is afgeleid van Microsofts SEAL FHE-schema. Chrome gebruikt een schema genaamd Argon2, wat niet volledig homomorf lijkt te zijn. De sterk versleutelde informatie wordt naar de cloud gestuurd om vergeleken te worden met grote wachtwoorddatabases. De resultaten daarvan, worden lokaal verwerkt om te kunnen bepalen of de betreffende gebruikersnaam en wachtwoord combinatie daadwerkelijk voorkomt in de database. Mocht dit het geval zijn, dan wordt de gebruiker gewaarschuwd. De externe partij kan met geen mogelijkheid het ontcijferde wachtwoord inzien, maar kan wel een vergelijkingsoperatie uitvoeren.

Een laatste voorbeeld van FHE in de praktijk, is het product ElectionGuard van Microsoft (Thornton, 2020). Het product levert een manier om te controleren of de verkiezingsuitslagen juist zijn en of de stemmen op geen enkele manier zijn gewijzigd, onderdrukt of mee geknoeid. Individuele kiezers kunnen zien dat hun stem nauwkeurig is geregistreerd en dat hun keuze correct is toegevoegd aan de uiteindelijke telling. Iedereen die de verkiezingen wil volgen, kan controleren of alle stemmen correct zijn opgeteld om een nauwkeurig en eerlijk resultaat te verkrijgen. Het principe van geheime stembiljetten houdt niet alleen in dat de stem van elke persoon privé *zou* moeten zijn, maar privé *moet* zijn, zodat stemmen niet kunnen worden gekocht, verkocht of afgedwongen. ElectionGuard gebruikt HE om dit te waarborgen. Elke stemmer krijgt een tracking code om te kunnen controleren of hun stem ongewijzigd door het systeem gaat en in de uiteindelijke telling terechtkomt. Deze code kan echter niet gebruikt worden om te bewijzen hoe er gestemd is, er kan alleen bewezen worden dat de stem niet is gewijzigd. Tijdens de telling kunnen alle versleutelde stemmen simpelweg bij elkaar opgeteld worden, zonder dat ooit iemand de inhoud van de stem heeft kunnen zien.

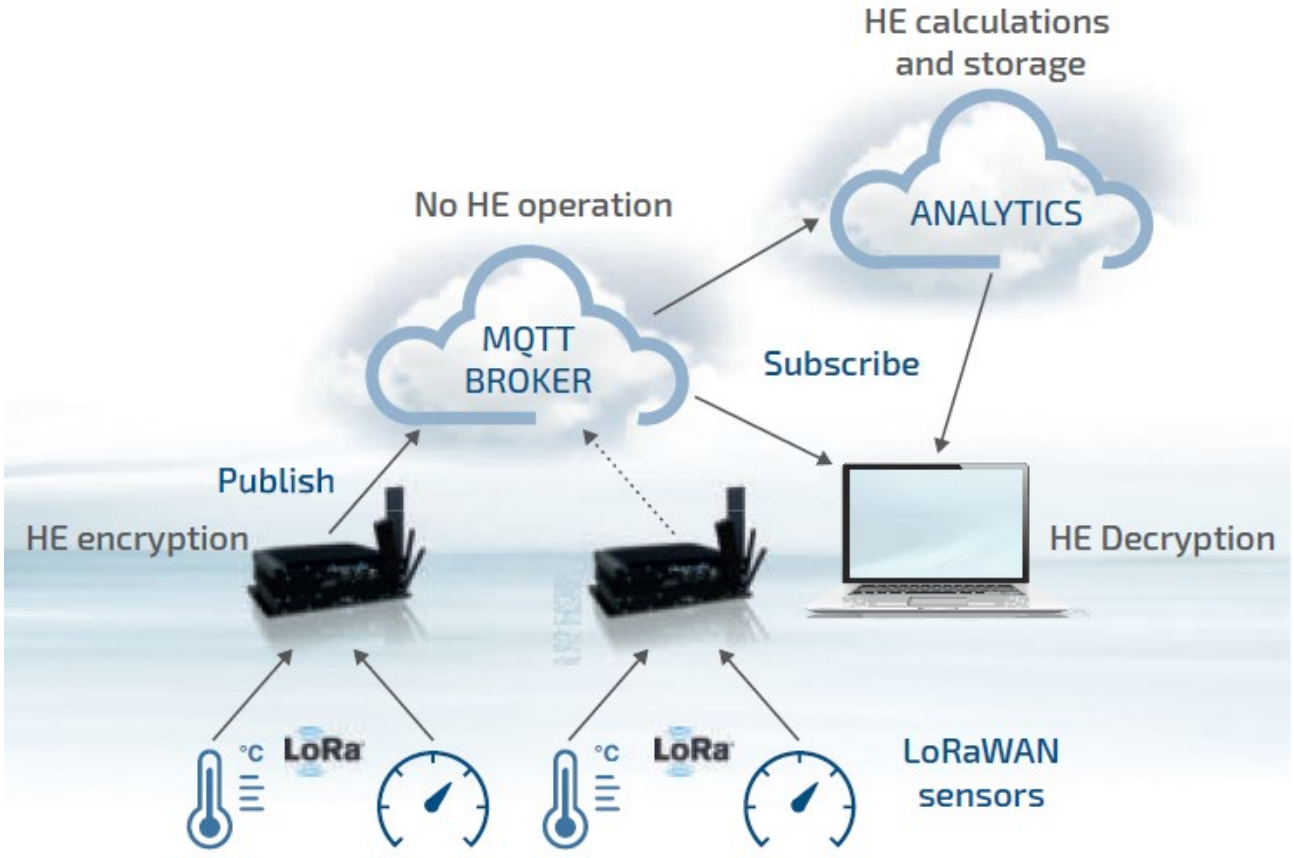
Mogelijke toepassingen

FHE wordt dus al op een aantal manier toegepast in de praktijk, maar welke toepassingen zijn er nog meer mogelijk?

IOT

HE heeft de potentie om de belangrijkste problemen van IoT op te lossen: beveiliging, opslag en berekeningen. Bel Korchi en El Mrabet (2019) omschrijven in een whitepaper een drietal use-cases, waarin HE wordt toegepast om de vertrouwelijkheid, privacy en anonimisering worden gewaarborgd. Het komt erop neer dat berichten die door sensoren naar een basisstation verstuurd worden, op dit station homomorf versleuteld worden en vervolgens naar de cloud worden gestuurd. Door de HE-versleuteling is het mogelijk om op een veilige manier statistieken te

halen uit de ontvangen sensordata. In Figuur 3 is weergegeven hoe dit er uit zou kunnen zien.



Figuur 3. IoT use case: the Edge/Cloud solution. Overgenomen uit *A Practical Use Case of Homomorphic Encryption* door Bel Korchi en El Mrabet, 2019, Kontron.

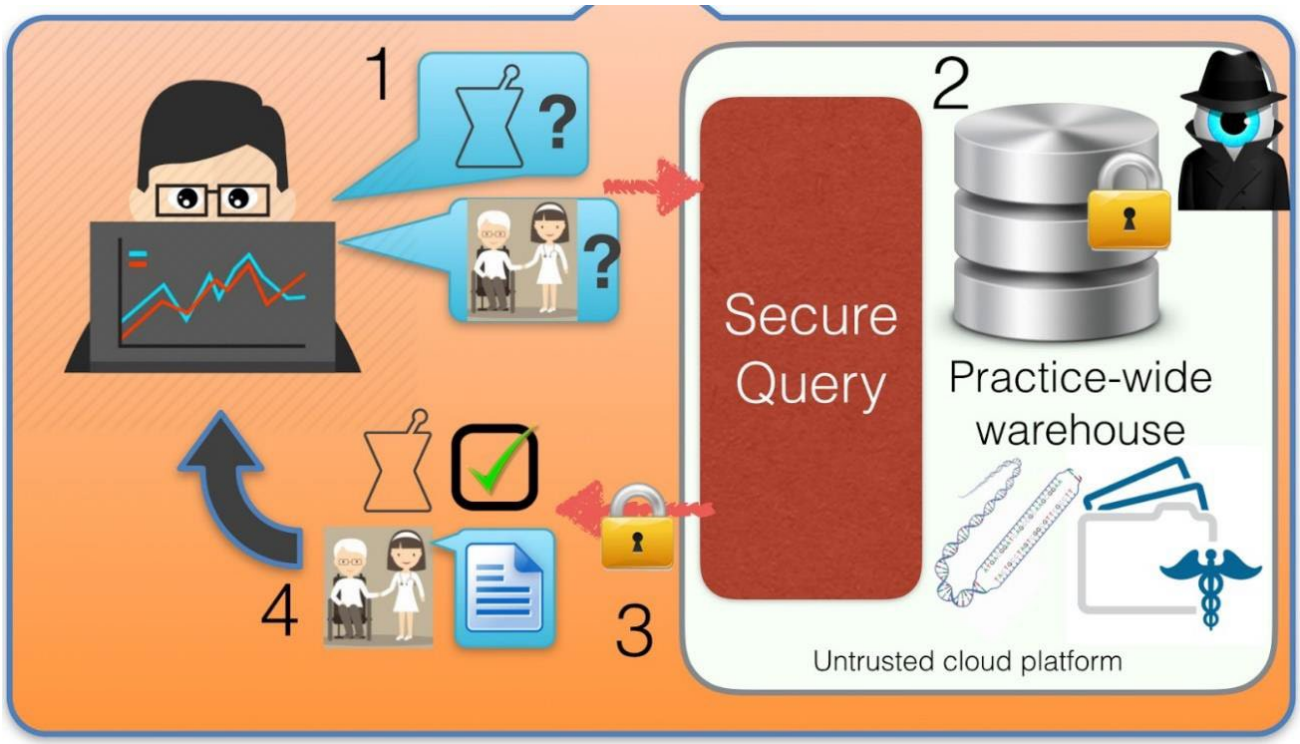
Genomica

Het delen van gegevens met behoud van privacy is een beperkende factor geworden op het gebied van genomica (Archer, et al., 2017). Menselijke DNA- en RNA-sequenties zijn biometrische identificatiemiddelen, net zoals een vingerafdruk. Zodra deze publiekelijk bekend zijn, kunnen ze nooit meer worden teruggehaald of teruggetrokken. Ze kunnen medisch significante informatie bevatten zoals een ziekterisico of sociaal gevoelige kenmerken. Momenteel wordt deze informatie vaak opgeslagen in een centrale database waarvan de toegang beperkt wordt. Een aantal use-cases voor het delen van genomica-gegevens, maken gebruik van eenvoudige bewerkingen op de gegevens en kunnen zeer geschikt zijn voor HE. Het DNA van iemand met een bepaalde ziekte kan bijvoorbeeld vergeleken worden met het DNA van iemand anders, om te bepalen of deze tweede persoon genetische aanleg heeft voor de afwijking in kwestie.

E-health

Gezondheidszorgsystemen werken in een omgeving waar gevoelige informatie moet worden beschermd tegen openbaarmaking, maar toch beschikbaar moet zijn als invoer voor alledaagse berekeningen (Archer, et al., 2017). Er worden tegenwoordig steeds hogere boetes uitgedeeld wanneer privégegevens gelekt worden, dus het is

zaak deze systemen goed te beveiligen. Hoewel ‘cyberverzekeringen’ enige bescherming bieden tegen dergelijke schade, kunnen kleinere ziekenhuizen en klinieken dit vaak niet betalen. Een verzekering afsluiten moet alleen geen excuus zijn om nalatig te zijn in de beveiliging. HE kan helpen om de balans tussen risico’s en bruikbaarheid van informatie aan te pakken. Facturering en het genereren van rapporten zijn twee toepassingen die hiervan kunnen profiteren. In beide gevallen hebben analisten toegang nodig tot individuele medische dossiers, om berekeningen te kunnen doen over een deel van hun inhoud. Door dergelijke berekeningen toe te staan zonder de onderliggende informatie inzichtelijk te maken, kunnen inbreuken worden vermeden zonder afbreuk te doen aan de dagelijkse processen. Figuur 4 visualiseert hoe dit proces eruit zou kunnen zien.



Figuur 4. Homomorphic encryption maakt analytische workflows betreffende gevoelige gegevens van de kliniek mogelijk. Overgenomen uit *Applications of Homomorphic Encryption* door Archer et al., 2017, Homomorphic Encryption Standardization.

Naehrig et al. (2011) stellen een scenario voor waarin medische gegevens van een patiënt (continu) in een versleutelde vorm worden geüpload naar een dienstverlener. De gebruiker is hier de gegevenseigenaar, dus de gegevens worden versleuteld onder de publieke sleutel van de gebruiker. Alleen de gebruiker kan de gegevens dus ontsleutelen. De dienstverlener laat vervolgens berekeningen los op de versleutelde gegevens, die kunnen bestaan uit zaken als bloeddruk, hartslag, gewicht of bloedsuikermeting. Hiermee kan de waarschijnlijkheid van bepaalde aandoeningen voorspeld worden of meer in het algemeen om de gezondheid van de patiënt bij te houden. Het belangrijkste voordeel hiervan is dat real-time gezondheidsanalyses mogelijk zijn op basis van metingen uit verschillende bronnen, zonder dat deze gegevens aan een enkele bron hoeven te worden bekendgemaakt.

Schoolverlaters

Elk jaar verlaten meer dan 1,2 miljoen studenten voortijdig de Amerikaanse 'high school' (DoSomething.org, z.d.). In Nederland is dit probleem een stuk kleiner, maar nog steeds verlaatten er in het schooljaar 2018-2019 26.894 leerlingen hun school (Nederlands Jeugdinstituut, 2020). In Nederland is dit vooral een probleem op het mbo. Om dit probleem te verminderen, zou het handig zijn als er voor alle studenten een risico van uitvallen berekend kan worden (Archer, et al., 2017). Het is echter onwaarschijnlijk dat scholen hiervoor voldoende informatie hebben. De kans op uitval zou bijvoorbeeld beïnvloed kunnen worden door gezondheidsproblemen. Daarom is het nodig om informatie van verschillende instellingen, in dit geval scholen, ziekenhuizen, welzijnssystemen, politie en meer samen te brengen. Deze instellingen zijn uiteraard verplicht om hun gegevens te beschermen omwille van de privacy, wat een ernstig probleem vormt voor deze integratie. HE kan hier de oplossing bieden. De gegevens zijn beschikbaar voor berekening, zonder het risico de wet te overtreden.

Privacy bij advertenties

Hoewel advertenties vaak ongewenst zijn, kunnen ze nuttig zijn wanneer ze afgestemd worden op de behoeften van de gebruiker (Armknecht, et al., 2015). Veel gebruikers maken zich echter zorgen over de privacy van hun gegevens, in dit geval hun voorkeuren of locatie. Er zijn al verschillende benaderingen voor dit probleem geweest. Jeckmans et al. (2013) schetsen een scenario waarin een gebruiker aanbevelingen voor een product wil. Het scenario is ontworpen rond een sociaal netwerk waar aanbevelingen zijn gebaseerd op de smaak van de vrienden van de gebruiker met de voorwaarde van vertrouwelijkheid. Het voorgestelde systeem past HE toe zodat een gebruiker aanbevelingen van vrienden kan krijgen zonder dat de identiteit van de aanbeveler wordt onthuld.

Financiële privacy

Stel een scenario voor waarin een bedrijf gevoelige gegevens heeft en ook eigen algoritmen die ze niet openbaar willen maken (Armknecht, et al., 2015). Dit kunnen bijvoorbeeld algoritmen zijn voor koersvoorspelling in de financiële sector. Naehrig et al. (2011) stellen voor om HE te gebruiken om zowel de gegevens al het algoritme in versleutelde vorm te uploaden om de berekeningen uit te besteden aan een clouddienst. Het geheim houden van het algoritme is echter niet iets wat HE biedt, maar is eerder onderdeel van verduisteringsonderzoek. Wat HE biedt, is de oplossing voor een gerelateerd probleem. Stel dat bedrijf A gevoelige gegevens heeft, zoals een aandelenportefeuille, en een ander bedrijf B heeft geheime algoritmen die voorspellingen doen over de aandelenkoers. Als A de algoritmen van B zou willen gebruiken, zou A ofwel de aandelenportefeuille aan B moeten bekendmaken, of B moet het algoritme aan A geven. Met HE kan A de gegevens echter versleutelen met een circuit privéschema en het naar B sturen. B kan vervolgens het algoritme

uitvoeren en alleen het resultaat terugsturen. A kan tot slot dit resultaat ontcijferen met hun geheime sleutel.

Forensische beeldherkenning

Bosch et al. (2014) beschrijven hoe forensische beeldherkenning kan worden uitbesteed (Armknecht, et al., 2015). Dergelijke tools worden door de politie en andere wetshandhaving instanties gebruikt om illegale afbeeldingen op een harde schijf, netwerk datastromen en andere datasets op te sporen. De politie gebruikt een database met hash-waarden van “slechte” afbeeldingen. Een grote zorg is dat daders deze database kunnen bemachtigen, kunnen controleren of hun afbeeldingen zouden worden gedetecteerd en, zo ja, ze kunnen wijzigen. HE kan in dit geval toegepast worden om deze database te versleutelen, terwijl de verwerkers het kunnen blijven gebruiken om criminelen op te sporen.

4. Conclusie

De hoofdvraag van dit onderzoek luidt: “Voor welke bedrijfstoepassingen uit de hedendaagse praktijk is Fully Homomorphic Encryption geschikt?” Om een goed geformuleerd antwoord te kunnen geven op deze vraag, moet er naar een aantal aspecten gekeken worden.

Wat is Fully Homomorphic Encryption (FHE)? Homomorfe encryptie is een techniek waarbij gegevens in cijfertekst kunnen worden omgezet, terwijl ze geanalyseerd en verwerkt kunnen worden alsof ze nog in de oorspronkelijke vorm zijn. Wanneer deze encryptie volledig homomorf is, kan het voor elk gewenst doel gebruikt worden. FHE heeft door de jaren heen al veel ontwikkelingen meegemaakt. Het begon allemaal in 2008 toen Craig Gentry voor het eerst een volledig homomorfe encryptie had bedacht, maar eigenlijk al in 1978 toen het voor het eerst werd voorgesteld. In de daaropvolgende jaren zijn er verbeteringen geboekt op het gebied van efficiëntie en veiligheid, maar er zijn vooral verschillende implementaties gekomen van FHE. Dit laatste heeft ervoor gezorgd dat er meer mogelijk is geworden, zoals machine learning.

Wat zijn de voor- en nadelen van Fully Homomorphic Encryption? Om te beginnen met de voordelen, FHE ondersteund in theorie alle mogelijke operaties op data. Hierdoor zou data nooit meer ontcijferd hoeven worden, wat de veiligheid verhoogt. Doordat gegevens in versleutelde vorm verwerkt kunnen worden, kan de verwerking plaatsvinden op hardware van een onbetrouwbare partij. Dit biedt nieuwe kansen voor het genereren van inkomsten uit datasets. FHE kent ook genoeg nadelen. Er zijn momenteel beperkingen in het ondersteunen van een breed scala aan bewerkingen/functies. FHE wordt gezien als de perfecte oplossing voor encryptie, maar dit moet per applicatie worden bekeken. Een one-size-fits-all oplossing zal niet zo veilig zijn als een oplossing die ontworpen is voor de toepassing in gedachten. Daarnaast zijn de prestaties van FHE zijn momenteel zeer slecht. Dit komt onder andere door de inefficiënties die nog aanwezig zijn, waaronder problemen rondom cryptografische ruis. Tot slot is de technologie niet gebruiksvriendelijk. FHE vereist expertise op het gebied van het onderliggende cryptografische schema.

In hoeverre maken bedrijfstoepassingen momenteel gebruik van Fully Homomorphic Encryption? IBM zegt bij een Braziliaanse en Europese bank FHE geïmplementeerd te hebben. In het geval van de eerste bank ging dit om een machine-learning applicatie. Ook de webbrowsers Google Chrome en Microsoft Edge maken gebruik van homomorfe encryptie. In deze gevallen wordt de techniek gebruikt om wachtwoorden op te slaan en te controleren of deze voorkomen in een datalek. Of een toepassing als ElectionGuard van Microsoft, wat stemmachines veiliger en betrouwbaarder maakt. Er zijn echter nog veel meer toepassingen van FHE te

bedenken. In IoT kan FHE helpen met het beveiligen van sensordata, zowel in transit als tijdens de verwerking. In de genomica kan het veilige opslag en verwerking garanderen. Dit geldt ook voor gezondheidssystemen. Patiëntgegevens zouden gebruikt kunnen worden met derden, zonder privacy in het geding te brengen. Tot slot zou de politie FHE kunnen gebruiken om hun databank met verboden media te beveiligen. Met forensische beeldherkenning proberen de instanties namelijk illegale afbeeldingen op te sporen. De grote zorg is dat de referentiedatabase in handen komt van criminelen, waarmee de beeldherkenning omzeilt zou kunnen worden.

Dus, voor welke bedrijfstoepassingen uit de hedendaagse praktijk is Fully Homomorphic Encryption geschikt? Rekening houdend met de huidige vorm van FHE (toekomstige ontwikkelingen dus daargelaten), zijn dat verwerkingen van data waar individueel versleutelde data opgeteld of vermenigvuldigd moeten worden, de verwerkingstijd hoog mag zijn en het niet nodig is het resultaat oneindig vaak te gebruiken voor nieuwe berekeningen. Zoals wachtwoordmonitoring, stemmachines, IoT en het delen van bedrijfsdata met derde partijen. De beperkende factoren zijn hier de ondersteunde operaties, prestaties en de ruis. Het is nog niet heel praktisch om FHE te implementeren, dus het kan een goed idee zijn om ook naar gedeeltelijk homomorfe encryptievormen te kijken.

5. Discussie

In dit hoofdstuk wordt ingegaan op de conclusies van het onderzoek door de resultaten te interpreteren, deze te koppelen aan verwachtingen, de beperkingen en eventuele implicaties van het onderzoek te bespreken en suggesties te doen voor vervolgonderzoek.

5.1 Validiteit

Voor dit onderzoek was het belangrijk om te achterhalen voor welke bedrijfstoepassingen Fully Homomorphic Encryption (FHE) ingezet kan worden. Hiervoor is de literatuur geraadpleegd voor antwoorden. Dit heeft tot een goed geïnformeerde onderbouwde conclusie kunnen leiden.

De conclusies zijn generaliseerbaar, omdat deze allereerst naar het hoofdstuk Resultaten terug gevolgd kunnen worden en vanuit daar naar de originele bron. Deze bronnen zijn voor iedereen toegankelijk, al zal er soms een betaling nodig zijn om deze volledig in te kunnen zien. De zoekmachines die gebruikt zijn om deze bronnen te vinden, zouden een bedreiging kunnen vormen voor de validiteit. Menig zoekmachine gebruikt tegenwoordig algoritmen om de meest relevante zoekresultaten weer te kunnen geven. Hier kan een bepaalde bias bij betrokken zijn. Dit is de exacte reden waarom er in dit onderzoek verschillende zoekmachines gebruikt zijn om bronnen te vergaren. Ook zoekmachines die geen algoritmen gebruiken om gepersonaliseerde resultaten te kunnen tonen. De selectie van bronnen wordt vervolgens doorgenomen en vergeleken. Wanneer bronnen elkaar tegenspraken, is dit verder uitgezocht om tot de waarheid te kunnen komen. Wat betreft de taal en publicaties waarin deze bronnen verschenen, heeft dit de validiteit niet in gevaar kunnen brengen. Het gaat in dit onderzoek om feitelijke resultaten en die verschillen niet tussen verschillende talen of publicaties.

De betrouwbaarheid van het onderzoek is gewaarborgd. De methode is op consistente wijze toegepast door steeds dezelfde zoekmachines te gebruiken en bronnen te verzamelen op een centrale plek.

5.2 Resultaten

Mijn resultaten komen totaal niet overeen met mijn verwachtingen, om de simpele reden dat ik voor dit onderzoek nauwelijks wat van het onderwerp af wist. Ik had verwacht dat dit een vervanging van traditionele encryptie zou zijn, maar dat blijkt op het moment niet zo te zijn. Het blijkt grote kansen te bieden op het gebied van ‘3rd party processing’, maar wordt tegengehouden door veel kinderziektes. Ik beschreef in de inleiding dat “nog weinig mensen hebben deze punten [de technische achtergrond van FHE en praktische toepassingen] echt samen laten komen om te kunnen concluderen of FHE al geschikt is om te gebruiken”. Met de resultaten van dit onderzoek geloof ik dat ik dit bereikt heb. Daarnaast noemde ik dat FHE het

informatielandschap een stuk veiliger zou kunnen maken en het genoemd wordt als de “heilige graal” van encryptie. Dit is zeker bevestig door de onderzoeksresultaten, ook al is de technologie er nog niet helemaal in de huidige staat.

5.3 Limitaties

De grootste limitatie van dit onderzoek blijft de bronnen die gebruikt zijn. Het kan zijn dat hierin de waarheid vertekend is of de informatie verjaard is. Ook is het goed mogelijk dat er nog betere informatie te vinden is over het onderwerp FHE, maar deze niet is meegenomen. Het lijkt er echter niet op dat deze limitatie een noemenswaardige impact heeft gehad op de resultaten. Het is lastig om deze beperking volledig te voorkomen, maar het blijft zaak zoveel mogelijk verschillende bronnen te verzamelen en gebruiken. Ik geloof dat ik dat in dit geval heb gedaan.

Het ontwijken van te veel diepgang in dit onderzoek is niet zo zeer een tekortkoming, maar het is wel iets waar andere onderzoeken profijt aan kunnen hebben. Een goed voorbeeld hiervan is de uitleg rondom cryptografische ruis. Het was voor dit onderzoek niet relevant om diep in dat onderwerp te duiken, maar het is wel iets wat in een ander onderzoek nader uitgezocht zou kunnen worden.

5.4 Implicaties

Het is nu duidelijk welke voordelen en nadelen FHE te bieden heeft en op welke manieren het in de praktijk toegepast zou kunnen worden. SALT Cyber Security kan deze informatie gebruiken om te besluiten of ze verder in deze technologie willen investeren of niet.

5.5 Vervolgonderzoek

Op de eerste plaats is er nog veel onderzoek nodig naar hoe Fully Homomorphic Encryption verbeterd kan worden. Er zijn nog simpelweg te veel nadelen om het echt goed op de kaart te kunnen zetten. In de tussentijd moet er veel geëxperimenteerd gaan worden met de technologie. Pak een van de bedrijfstoepassingen en probeer FHE hier te implementeren. Op deze manier komen praktische tekortkomingen duidelijk naar boven en kan er tegelijkertijd naar oplossingen gezocht worden. Dit kan grote positieve gevolgen hebben op de mogelijkheden voor de toekomst. Ook zou het goed zijn om te weten welke plaats Partially Homomorphic Encryption kan hebben zolang FHE nog niet goed genoeg is.

6. Literatuurlijst

Albrecht, M., Bai, S., & Ducas, L. (2016). A subfield lattice attack on overstreched NTRU assumptions. *CRYPTO 2016*. Opgeroepen op 9 februari, 2021, van https://eprint.iacr.org/2016/127.pdf

Alperin-Sheriff, J., & Peikert, C. (2014). Faster Bootstrapping with Polynomial Error. *CRYPTO 2014*. Opgeroepen op 9 februari, 2021, van https://eprint.iacr.org/2014/094.pdf

Arampatzis, A. (2020, 22 januari). *What is Homomorphic Encryption [& How It Is Used] | Venafi*. Opgeroepen op 4 februari\, 2021, van https://www.venafi.com/blog/homomorphic-encryption-what-it-and-how-it-used

Archer, D., Chen, L., Cheon, J. H., Gilad-Bachrach, R., Hallman, R. A., Huang, Z., . . . Wang, S. (2017). *Applications of Homomorphic Encryption.* Homomorphic Encryption Standardization. Opgeroepen op 15 februari, 2021, van https://homomorphicencryption.org/white_papers/applications_homomorphic_encryption_white_paper.pdf

Armknecht, F., Boyd, C., Carr, C., Gjøsteen, K., Jäschke, A., Reuter, C. A., & Strand, M. (2015). *A Guide to Fully Homomorphic Encryption.* International Association for Cryptologic Research. Opgeroepen op 4 februari, 2021, van https://eprint.iacr.org/eprint-bin/getfile.pl?entry=2015/1192&version=20160914:161519&file=1192.pdf

Bel Korchi, A., & El Mrabet, N. (2019). *A Practival Use Case of Homomorphic Encryption.* Kontron. Opgeroepen op 17 februari, 2021, van https://www.kontron.com/download/download?filename=/downloads/white_papers/usecase_homomorphic_encryption-web.pdf&type=collateral

Boneh, D., Goh, E., & Nissim, K. (2005). Evaluating 2-DNF Formulas on Ciphertexts. *Theory of Cryptography Conference*. Opgeroepen op 9 februari, 2021, van https://link.springer.com/content/pdf/10.1007/978-3-540-30576-7_18.pdf

Bonestroo, W., Meesters, M., Niels, R., Schagen, J., Henneke, L., & Van Turnhout, L. (2018). *ICT Research Methods.* Amsterdam: HBO-i. Opgeroepen op 4 februari, 2021, van http://ictresearchmethods.nl/index.php?title=Methods&oldid=100

Bos, J., Lauter, K., Loftus, J., & Naehrig, M. (2013). Improved Security for a Ring-Based Fully Homomorphic Encryption Scheme. *IMACC 2013*. Opgeroepen op 9 februari, 2021, van https://eprint.iacr.org/2013/075.pdf

Bösch, C., Peter, A., Hartel, P., & Jonker, W. (2014). SOFIR: Securely Outsourced Forensic Image Recognition. *39th IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2014* (pp. 2694-2698). IEEE. Opgeroepen op 19 februari,

2021, van https://research.utwente.nl/en/publications/sofir-securely-outsourced-forensic-image-recognition

Brakerski, Z. (2012). Fully Homomorphic Encryption without Modulus Switching from Classical GapSVP. *CRYPTO 2012*. Opgeroepen op 9 februari, 2021, van https://eprint.iacr.org/2012/078.pdf

Brakerski, Z., & Vaikuntanathan, V. (2011). Efficient Fully Homomorphic Encryption from (Standard) LWE. *FOCS 2011*. Opgeroepen op 9 februari, 2021, van https://eprint.iacr.org/2011/344.pdf

Brakerski, Z., & Vaikuntanathan, V. (2013). Lattice-Based FHE as Secure as PKE. *ITCS 2014*. Opgeroepen op 9 februari, 2021, van https://eprint.iacr.org/2013/541.pdf

Brakerski, Z., Gentry, C., & Vaikuntanathan, V. (2011). Fully Homomorphic Encryption without Bootstrapping. *ITCS 2012*. Opgeroepen op 9 februari, 2021, van https://eprint.iacr.org/2011/277.pdf

Brands, C., Golovko, D., Van den Nieuwenhoff, T., & Stokman, S. (2020). *De effecten van het "groene slotje".* Zwolle: Hogeschool Windesheim.

Carpov, S., Chillotti, I., Gama, N., Georgieva, M., & Izabachene, M. (2016). *TFHE: Fast Fully Homomorphic Encryption Library*. Opgeroepen op 9 februari, 2021, van https://tfhe.github.io/tfhe/

Cheon, J., Jeong, J., & Lee, C. (2016). An algorithm for NTRU problems and cryptanalysis of the GGH multilinear map without a low-level encoding of zero. *LMS Journal of Computation and Mathematics, 19*(A), 255-266. Opgeroepen op 9 februari, 2021, van https://www.cambridge.org/core/journals/lms-journal-of-computation-and-mathematics/article/an-algorithm-for-ntru-problems-and-cryptanalysis-of-the-ggh-multilinear-map-without-a-lowlevel-encoding-of-zero/230ECFEEE6AF4D8027FF3E13998D560C

Cheon, J., Kim, A., Kim, M., & Song, Y. (2017). Homomorphic Encryption for Arithmetic of Approximate Numbers. *ASIACRYPT 2017*, 409-437. Opgeroepen op 9 februari, 2021, van https://link.springer.com/content/pdf/10.1007%2F978-3-319-70694-8_15.pdf

DoSomething.org. (z.d.). *11 Facts About High School | DoSomething.org*. Opgeroepen op 19 februari, 2021, van https://www.dosomething.org/us/facts/11-facts-about-high-school-dropout-rates

Dropbox Inc. (z.d.). *¿Cuál es el nivel de seguridad de Dropbox?* Opgeroepen op 15 oktober, 2014, van https://web.archive.org/web/20141015083125/https://www.dropbox.com/help/27

Ducas, L., & Micciancio, D. (2014, 7 december). *FHEW: A Fully Homomorphic Encryption library*. Opgeroepen op 9 februari, 2021, van

https://github.com/lducas/FHEW/tree/0959af8daf6635a5e69013f6db7120c6d39e2319

Fan, J., & Vercauteren, F. (2012). *Somewhat Practical Fully Homomorphic Encryption.* Leuven: Katholieke Universiteit Leuven. Opgeroepen op 9 februari, 2021, van https://eprint.iacr.org/2012/144.pdf

Gama, N., Izabachène, M., Nguyen, P., & Xie, X. (2014). Structural Lattice Reduction: Generalized Worst-Case to Average-Case Reductions and Homomorphic Cryptosystems. *EUROCRYPT 2016*. Opgeroepen op 9 februari, 2021, van https://eprint.iacr.org/2014/283.pdf

Gentry, C. (2009). *A Fully Homomorphic Encryption Scheme.* Stanford: Stanford University. Opgeroepen op 19 februari, 2021, van https://luca-giuzzi.unibs.it/corsi/Support/papers-cryptography/craig-thesis.pdf

Gentry, C. (2009). Fully Homomorphic Encryption Using Ideal Lattices. *41st ACM Symposium on Theory of Computing (STOC)*. Opgeroepen op 9 februari, 2021, van https://www.cs.cmu.edu/~odonnell/hits09/gentry-homomorphic-encryption.pdf

Gentry, C., & Halevi, S. (2011). *Implementing Gentry's Fully-Homomorphic Encryption Scheme.* IBM Research. Opgeroepen op 9 februari, 2021, van https://eprint.iacr.org/2010/520.pdf

Gentry, C., Sahai, A., & Waters, B. (2013). Homomorphic Encryption from Learning with Errors: Conceptually-Simpler, Asymptotically-Faster, Attribute-Based. *CRYPTO 2013*. Opgeroepen op 9 februari, 2021, van https://eprint.iacr.org/2013/340.pdf

Golovko, D., Van 't Hoog, J., Van den Nieuwenhoff, T., Zeldenrust, W., & Zwanenburg, M. (2020). *Application Containerization Platform Onderzoeksrapport.* Zwolle: Hogeschool Windesheim.

Hallman, R. A., Laine, K., Dai, W., Gama, N., Malozemoff, A. J., Polyakov, Y., & Carpov, S. (2018). *Building Applications with Homomorphic Encryption.* Opgeroepen op 12 februari, 2021, van https://homomorphicencryption.org/wp-content/uploads/2018/10/CCS-HE-Tutorial-Slides.pdf

Huynh, D. (2020, 19 juni). *Homomorphic Encryption intro: Part 1: Overview and use cases | by Daniel Huynh | Towards Data Science*. Opgeroepen op 15 februari, 2021, van https://towardsdatascience.com/homomorphic-encryption-intro-part-1-overview-and-use-cases-a601adcff06c

IBM. (z.d.). *Unlock value of sensitive data without decryption | IBM*. Opgeroepen op 15 februari, 2021, van https://www.ibm.com/security/digital-assets/fhe/unlock-value-of-sensitive-data-without-decryption

Ishai, Y., & Paskin, A. (2007). Evaluating Branching Programs on Encrypted Data. *Theory of Cryptography*. Opgeroepen op 9 februari, 2021, van https://link.springer.com/content/pdf/10.1007%2F978-3-540-70936-7_31.pdf

Jeckmans, A., Peter, A., & Hartel, P. (2013). Efficient privacy-enhanced familiarity-based recommender system. *Proceedings of the 18th European Symposium on Research in Computer Security, ESORICS 2013* (pp. 400-417). Springer. Opgeroepen op 19 februari, 2021, van https://research.utwente.nl/en/publications/efficient-privacy-enhanced-familiarity-based-recommender-system

Lauter, K., Kannepalli, S., Laine, K., & Moreno, R. C. (2021, 21 januari). *Password Monitor: Safeguarding passwords in Microsoft Edge - Microsoft Research*. Opgeroepen op 15 februari, 2021, van https://www.microsoft.com/en-us/research/blog/password-monitor-safeguarding-passwords-in-microsoft-edge/

Lauter, K., Naehrig, M., & Vaikuntanathan, V. (2011). *Can Homomorphic Encryption be Practical?* Redmond: Microsoft Research. Opgeroepen op 19 februari, 2021, van https://eprint.iacr.org/2011/405.pdf

Li, B., & Micciancio, D. (2020). *On the Security of Homomorphic Encryption on Approximate Numbers.* San Diego: University of California. Opgeroepen op 9 februari, 2021, van https://eprint.iacr.org/2020/1533.pdf

Lopez-Alt, A., Tromer, E., & Vaikuntanathan, V. (2013). On-the-Fly Multiparty Computation on the Cloud via Multikey Fully Homomorphic Encryption. *STOC 2012*. Opgeroepen op 9 februari, 2021, van https://eprint.iacr.org/2013/094.pdf

Masters, O., Hunt, H., Steffinlongo, E., Crawford, J., Bergamaschi, F., Dela Rosa, M. E., . . . Ferreira, D. G. (2019). *Towards a Homomorphic Machine Learning Big Data Pipelin for the Financial Services Sector.* Hursley: IBM Research. Opgeroepen op 15 februari, 2021, van https://eprint.iacr.org/2019/1113.pdf

MEGA. (z.d.). *MEGA*. Opgeroepen op 10 februari, 2021, van https://mega.io/

Moskvitch, K. (2020, 10 januari). *Top Brazilian Bank Pilots Privacy Encryption Quantum Computers Can't Break | by Inside IBM Research | Medium*. Opgeroepen op 12 februari, 2021, van https://ibm-research.medium.com/top-brazilian-bank-pilots-privacy-encryption-quantum-computers-cant-break-92ed2695bf14

Nederlands Jeugdinstituut. (2020, 9 maart). *Voortijdig schoolverlaten - Cijfers | NJi*. Opgehaald van https://www.nji.nl/nl/Databank/Cijfers-over-Jeugd-en-Opvoeding/Cijfers-per-onderwerp/Voortijdig-schoolverlaten

Pullman, J., Thomas, K., & Bursztein, E. (2019, 6 februari). *Google Online Security Blog: Protect your accounts from data breaches with Password Checkup*. Opgeroepen op

16 februari, 2021, van https://security.googleblog.com/2019/02/protect-your-accounts-from-data.html

Rahman, M. S., Khalil, I., Atiquzzaman, M., & Yi, X. (2020). Towards privacy preserving AI based composition framework in edge networks using fully homomorphic encryption. *Engineering applications of artificial intelligence, 94*. Opgeroepen op 10 februari, 2021, van https://www-sciencedirect-com.windesheim.idm.oclc.org/science/article/pii/S0952197620301512?via%3Dihub

Rivest, R. L., Adleman, L., & Dertouzos, M. L. (1978). On Data Banks and Privacy Homomorphisms. *Foundations of Secure Computation*. Opgeroepen op 9 feburari, 2021, van https://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=EA7BED6BE588981303D0EEC5C2EDDB0B?doi=10.1.1.500.3989&rep=rep1&type=pdf

Salter, J. (2020, 31 juli). *IBM completes succesful field trials on Fully Homomorphic Encryption*. Opgeroepen op 15 februari, 2021, van https://arstechnica.com/gadgets/2020/07/ibm-completes-successful-field-trials-on-fully-homomorphic-encryption/

Sander, T., Young, A. L., & Yung, M. (1999). *Non-Interactive CryptoComputing For NC1.* Focs1991.

Solomon, R. (2020, 2 juli). *An Intro to Fully Homomorphic Encryption for Engineers*. Opgeroepen op 4 februari, 2021, van https://blog.nucypher.com/an-engineers-guide-to-fully-homomorphic-encryption/

TechTarget Contributor. (2011). *What is homomorphic encryption? - Definition from WhatIs.com*. Opgeroepen op 5 februari, 2021, van https://searchsecurity.techtarget.com/definition/homomorphic-encryption

Thornton, A. (2020, 27 maart). *What is ElectionGuard? | Microsoft On the Issues*. Opgeroepen op 15 februari, 2021, van https://news.microsoft.com/on-the-issues/2020/03/27/what-is-electionguard/

Vaikuntanathan, V. (2012). Homomorphic Encryption: WHAT, WHY, and HOW. Toronto: University of Toronto. Opgeroepen op 8 februari, 2021, van https://www.cs.toronto.edu/~vinodv/Homomorphic-MCSS.pptx

Van Dijk, M., Gentry, C., Halevi, S., & Vaikuntanathan, V. (2009). Fully Homomorphic Encryption over the Integers. *Eurocrypt 2010*. Opgeroepen op 9 februari, 2021, van https://eprint.iacr.org/2009/616.pdf

Wikipedia contributors. (2020, 30 december). NTRU. *Wikipedia, The Free Encyclopedia*. Opgeroepen op 9 februari, 2021, van https://en.wikipedia.org/w/index.php?title=NTRU&oldid=997278152

Wikipedia contributors. (2021, 26 januari). Homomorphic encryption. *Wikipedia, The Free Encyclopedia*. Opgeroepen op 4 februari, 2021, van https://en.wikipedia.org/w/index.php?title=Homomorphic_encryption&oldid=1002934075

Wikipedia-bijdragers. (2019, 22 oktober). Homomorfe encryptie. Wikipedia, de vrije encyclopedie. Opgeroepen op 5 februari, 2021, van https://nl.wikipedia.org/w/index.php?title=Homomorfe_encryptie&oldid=54824375

Will, M. A., & Ko, R. K. (2015). A guide to homomorphic encryption. In R. Ko, & R. Choo, *The Cloud Security Ecosystem.* Opgeroepen op 4 februari, 2021, van https://learning.oreilly.com/library/view/the-cloud-security/9780128017807/B9780128015957000057.xhtml