Faculteit Ingenieurswetenschappen en Architectuur

Een privacyvriendelijk aanbevelingssysteem voor mobiele toestellen

door

Thorwald Frederik Lambrecht

Interne promotor: Marleen Denert

Externe promotor: Luc Martens

Externe begeleider: Toon De Pessemier

Scriptie ingediend tot het behalen van de academische graad van
Master of Science in de industriële wetenschappen: informatica

Academiejaar 2014–2015

Voorwoord

Hier komt wat tekst.

Thorwald Frederik Lambrecht, mei 2014

Toelating tot bruikleen

Thorwald Frederik Lambrecht, mei 2014

Inhoudsopgave

Hoofdstuk 1

Inleiding

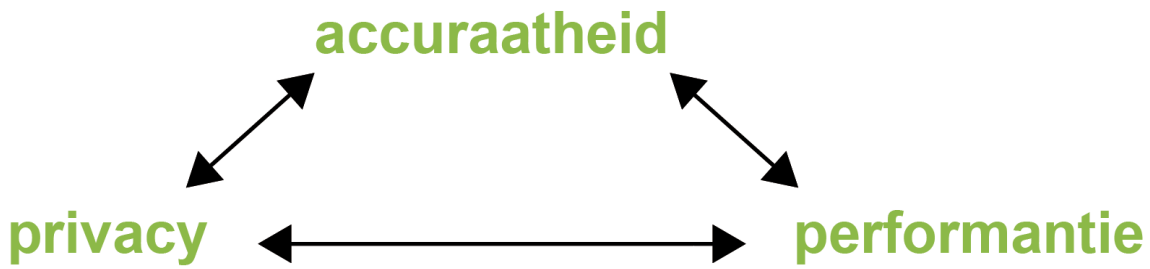
Een aanbevelingssysteem is een handig middel om zich een weg te banen in de enorme hoeveelheid content die sommige web applicaties aanbieden. Het laat ons toe snel interessante en persoonsgerichte items te vinden. Aanbevelingssystemen kunnen mensen nieuwe vrienden leren kennen of zorgen voor een boost in verkoopcijfers door gerichte marketing. Ze kunnen ook gebruikt worden om de gebruiker nieuwe dingen aan te leren. OWL [11] bijvoorbeeld zal gebruikers nieuwe shortcuts aangeven voor veelgebruikte commandos in een bepaald programma. Het kan ook een doel zijn om een community van gebruikers te creëren rond items. Een voorbeeld hiervan is Tripadvisor.com waar men niet beoogt om zoveel mogelijk reservaties te verkrijgen maar eerder een community uit te bouwen die elkaar helpt in het kiezen tussen hotels. Service providers die klassieke aanbevelingssystemen gebruiken houden vaak veel persoonlijke data bij op hun servers. De data van een persoon wordt bijgehouden in zijn gebruikersprofiel. Google heeft al aangegeven dat het alle data van één persoon van zijn verschillende services bijhoudt in één profiel.

Het bijhouden van deze gigantische hoeveelheden data brengt vragen met zich mee:

- Wordt deze data wel correct beheerd?
- Welke risico's zijn er voor onze privacy?
- Kan misbruik vermeden worden met alternatieve privacyvriendelijkere methodes?

Aanbevelingen zijn logischerwijs gezien beter of accurater naarmate het systeem beschikt over gedetailleerdere data. Er is dus in zekere zin een inherente wisselwerking tussen de privacy van de gebruiker en de accuraatheid van het systeem. In deze masterproef wordt gefocust op hoe we de verschillende klassieke algoritmes om aanbevelingen te berekenen kunnen aanpassen zodat

er minder risico is op privacy inbreuken in verband met persoonlijke voorkeuren of persoonlijke informatie in het algemeen. Jammer genoeg hebben de mogelijkheden die al beschreven zijn in de literatuur een negatieve impact op de performantie van het algoritme en / of de accuraatheid van de aanbevelingen.



Het doel van deze masterproef is een optimale oplossing te vinden voor deze drievoudige wisselwerking en dus het algoritme te vinden dat enerzijds privacyvriendelijk is maar waarvan anderzijds de performantie aanvaardbaar is en de aanbevelingen goed en persoonlijk gericht zijn. Ook wordt er rekening gehouden met de mobiele setting. Het algoritme moet bruikbaar zijn op een smartphone en dus rekening houden met eisen als CPU- of batterijgebruik.

Hiervoor bekijken we in hoofdstuk 2 in paragraaf 2.1 de klassieke aanbevelingssystemen en welke privacygevoelige data deze bijhouden. Vervolgens beschouwen we wat privacy betekent in een aanbevelingssysteemcontext en de risicos die voorkomen bij bestaande aanbevelingssystemen in paragrafen 2.2 tot en met 2.4. In hoofdstuk 3 worden de bestaande oplossingen bestudeerd en geanalyseerd en bekeken welke het best scoort op de drie hoofdpunten rekening houdend met een mobiele setting. De beste optie wordt als vertrekpunt genomen voor de uitwerking van een werkend privacyvriendelijk aanbevelingssysteem in hoofdstuk 4. We kozen aan clientkant voor een Androidapplicatie en aan de kant van de server kozen we voor Java.

Hoofdstuk 2

Privacyrisico's bij klassieke aanbevelingsystemen

2.1 Klassieke aanbevelingssystemen

De verschillende soorten aanbevelingssystemen worden besproken, elke soort krijgt een korte uitleg waarin de werking ervan aan bod komt. Voor meer informatie omtrent de precieze werking wordt doorverwezen naar de vakliteratuur.
Er wordt een onderscheid gemaakt tussen de volgende types.

2.1.1 Niet-gepersonaliseerde statistieken

Dit is de meest eenvoudige vorm van een aanbevelingssysteem. Statistieken van ratings van de community zoals het gemiddelde aantal sterren van een beoordeling of items met het grootste aantal "vind-ik-leuk"s zijn alomtegenwoordig. Andere statistieken die populair zijn, zijn productassociaties in de vorm van mensen die x leuk vonden, vonden ook y leuk. Externe community data, zoals bijvoorbeeld meest verkochte items, wordt ook gebruikt. Hoewel de niet-gepersonaliseerde statistieken duidelijk hun beperkingen hebben, zijn ze efficiënt en kunnen ze erg nuttig zijn.

2.1.2 Content-based recommenders

Het basisidee van inhoud-gebaseerde aanbevelingssystemen is om items te modelleren als vector in een meerdimensionale ruimte. Deze ruimte heeft als dimensies de relevante attributen van de items. De smaak van de gebruiker wordt ook voorgesteld als vector in deze ruimte, de *uservector*.

De uservector wordt opgesteld door zijn ratings van items die bepaalde attributen hebben. De interessante items worden meestal gevonden door die items te nemen waarvan de hoek tussen de itemvector en de uservector klein is.

2.1.3 Kennis-gebaseerde recommenders

De gebruiker die aanbevelingen vraagt zal zijn voorkeuren aan het systeem geven. De gebruiker kan op de aangeboden items feedback geven zodat het systeem zijn aanbevelingen kan aanpassen.

2.1.4 Collaborative Filtering recommenders

User-User Collaborative recommenders

Bij user-user collaborative recommenders wordt de smaak van gebruikers vergeleken op basis van hun gegeven ratings met een correlatiecoëfficiënt zoals deze van Pearson. Als een gebruiker aanbevelingen vraagt, worden de gebruikers met een gelijkaardige smaak bepaald. De score voor een item wordt berekend door het gewogen gemiddelde te nemen van de scores voor dat item van de gebruikers met gelijkaardige smaak.

Item-Item Collaborative recommenders

Om een score voor een item te vinden voor een gebruiker wordt eerst gekeken naar de items waar hij wel een rating voor heeft. Indien het gezochte item volgens andere gebruikers gelijkaardig is (een gelijkaardige score heeft voor hen) aan wel beoordeelde items, neemt men het gewogen gemiddelde van de ratings van deze items.

2.1.5 Andere aanbevelingssystemen

Demografic recommenders

Als een persoonlijke voorkeur niet gekend is wordt afgegaan op kenmerken als leeftijd, geslacht en land van herkomst om aanbevelingen te genereren op basis van een stereotype.

Social recommenders

Deze aanbevelingssystemen maken gebruik van de vriendschapsbanden van een sociaal netwerk omdat aangenomen wordt dat vrienden gelijkaardige interesses hebben.

Hybrid recommenders

Zoals de naam aangeeft worden hier verschillende aanbevelingssystemen samen gecombineerd. Dit heeft als doel betere aanbevelingen te maken en de nadelen van de onderliggende systemen weg te werken.

2.1.6 Singular Value Decomposition

Singular Value Decomposition of singulierewaardenontbinding is een wiskundige techniek die toelaat een $m \times n$ matrix A op te splitsen in een product van drie matrices respectievelijk een $m \times m$ matrix U, een $m \times r$ matrix $\sum$ met de singuliere waarden op de diagonaal en een $r \times r$ matrix V. Deze techniek kan uitgevoerd worden op een traditionele userratingmatrix om de smaken van de gebruikers niet vast te leggen als in scores op bepaalde trefwoorden maar eerder in een aantal dimensies die verschillende trefwoorden overstijgen. De singuliere waarden op de diagonaal van $\sum$ geven de belangrijkheid van deze bepaalde dimensie aan. Een deel van deze waarden is dicht bij 0 en dus kunnen deze dimensies verwaarloosd worden en de matrix verkleind worden naar k rijen en kolommen. De scores in de U en de V matrix op deze verwaarloosde dimensies kunnen ook weggelaten worden. Dit zorgt ervoor dat de data compacter kan worden bijgehouden en voor minder computationeel werk. Hierbij worden U de user-feature matrix en V de item-feature matrix genoemd.

2.1.7 Evaluatie aanbevelingssystemen

Een veelgebruikte basismetriek is Mean Absolute Error (MAE) of de gemiddelde absolute fout. Een rating voor een item wordt onzichtbaar gemaakt voor het systeem, waarna het systeem aangeeft welke score het zou geven op basis van alle andere bekende ratings. Het verschil tussen de vermomde score en de geraadde score is dus de fout. We berekenen het gemiddelde van de fouten voor elke rating en zo bekomen we het MAE. Een vergelijkbaar alternatief is de Mean Squared Error (MSE), die door kwadratering de grote fouten meer afstraft. De Root Mean Squared Error (RMSE) is hiervan de vierkantswortel zodat er een waarde bekomen wordt in de meer intuïtieve originele schaal.

$$MAE = \frac{\sum_{ratings}(prediction - rating)}{\#ratings} \tag{2.1}$$

$$MSE = \frac{\sum_{ratings}(prediction - rating)^2}{\#ratings} \tag{2.2}$$

$$RMSE = \sqrt{MSE} \tag{2.3}$$

Precision en *recall* zijn ook veelgebruikte metrieken bij *Information Retrieval* technieken. Ze focussen in tegenstelling tot de vorige metrieken niet op de nauwkeurigheid van een voorspellingswaarde. Precision en recall tonen respectievelijk hoeveel van de aanbevolen termen relevant zijn en het percentage van relevante items dat effectief wordt aanbevolen.

2.1.8 Privacygevoelige data

Veelgebruikte aanbevelingssystemen als content-based recommenders en collaborative filtering recommenders hebben toegang nodig tot persoonlijke voorkeuren om de berekeningen te kunnen doen. In de praktijk omvat dit expliciete data zoals commentaren, ratings of aankopen. Veel webapplicaties houden naast deze expliciete ook impliciete data bij zoals bezochte paginas of bijvoorbeeld hoe lang een gebruiker naar een bepaald filmpje kijkt op Youtube. Ook knowledge-based recommenders verzamelen deze voorkeuren. Demografic recommenders hebben kennis nodig over de kenmerken van de gebruiker en social recommenders moeten natuurlijk weet hebben van het sociale netwerk.

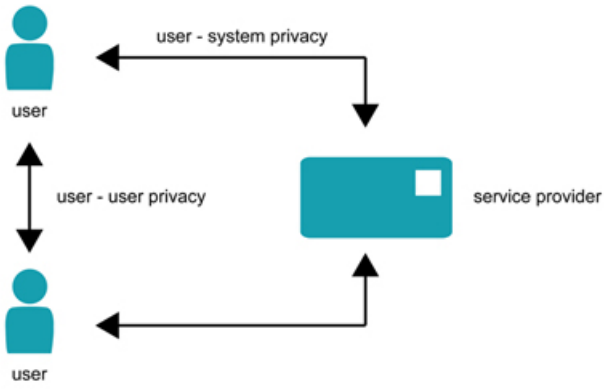
2.2 Privacy

Zoals eerder gezegd bestaat er een wisselwerking tussen privacy en accuraatheid van de aanbevelingen. Hier gaan we even in op wat men precies bedoelt met privacy. Privacy op het internet betekent privacy van informatie. In de literatuur verwijst men vaak naar de definitie van het Information Infrastructure Task Force (IITF).

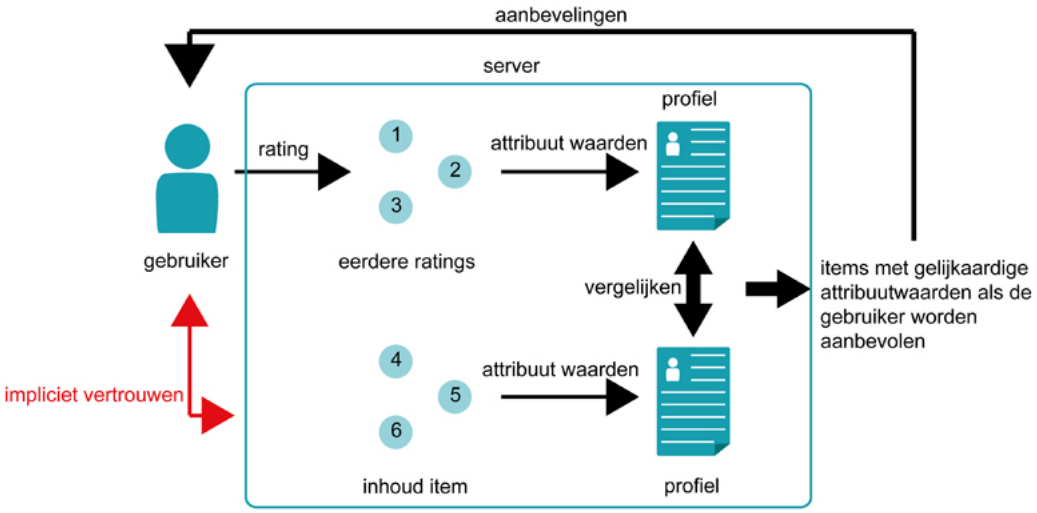
> "Privacy van informatie is het recht van een individu om controle uit te oefenen op de voorwaarden waaronder zijn persoonlijke informatie verzameld, gebruikt of bekendgemaakt wordt."
> – Information Infrastructure Task Force [8]

Informatie wordt door een individu altijd gedeeld binnen een bepaalde *scope*. Een scope wordt afgebakend door de grootte van het publiek, de manier waarop de informatie gebruikt mag worden en hoe lang dit mag. Privacy betekent in deze context dus de informatie in dezelfde scope te houden als vooropgesteld door de persoon die informatie verstrekte [8].
Privacy van aanbevelingssystemen kan worden opgedeeld in twee soorten. Er is *user-user privacy*, met betrekking tot de privacy tussen gebruikers onderling. Bij user-user privacy gaat het

om wat een gebruiker online allemaal kan te weten komen over het gedrag van andere gebruikers. Sommige websites tonen de ratings en persoonlijke voorkeuren publiek maar bij andere sites is de vooropgestelde scope veel kleiner en deze houden persoonlijke voorkeuren privé of onder vrienden.



In tegenstelling tot sociale netwerken ligt bij aanbevelingssystemen het grootste probleem bij *user-system privacy*. User-system privacy heeft betrekking tot de privacy kwesties tussen de gebruikers en de service provider. Een ander concept in privacy is *deniability of preferences*, wat betekent dat je als gebruiker de mogelijkheid hebt om je eigen voorkeuren te ontkennen. Er is dus geen zekerheid voor een partij om de link te leggen tussen jou en je ratings.



Architectuur van een klassiek content-based filtering systeem

Gebruikers hebben vaak een impliciet vertrouwen in de provider om verantwoord met hun data om te springen.

2.3 Betrouwbaarheid

De betrouwbaarheid van een aanbevelingssysteem leunt dicht aan bij de privacy. Hier kunnen we de volgende vragen stellen :

- Is het systeem voldoende beschermd tegen aanvallen van buitenaf?

Het aanbevelingssysteem moet allereerst bestand zijn tegen aanvallen van hackers die private informatie willen bemachtigen. In het verleden is er sprake geweest van buitenstaanders die ratings manipuleerden en extra gebruikersprofielen aanmaakten om extra informatie in te winnen over het gedrag van anderen. Een voorbeeld hiervan is een vroege aanval die de correlatiecoëfficient van Pearson misbruikte. De aanval maakte gebruik van het feit dat de coëfficient één is als twee gebruikers identiek dezelfde ratings hebben. Er werd met behulp van een beetje informatie over de gebruiker een valse account aangemaakt waarmee de coëfficient één werd, zo kon men informatie halen van de ratings van de gebruiker. Deze aanvallen kunnnen bemoeilijkt worden door een minder transparant algoritme te gebruiken dat bijvoorbeeld gebruik maakt van Singular Value Decomposition.
Daarbuiten moet het systeem ook voorkomen dat kwaadwillige gebruikers meerdere accounts aanmaken met als doel bepaalde items te promoten of in de vergetelheid te doen belanden.

- Beveelt het systeem de beste items wel aan?

Het kan dat een service provider enkel deze items aanbeveelt die in stock zijn of waar het systeem het meest geld op verdient en niet deze die het interessantst zijn voor de gebruiker.

2.4 Privacyrisico's bij aanbevelingssystemen

2.4.1 De gebruiker onderschat de omvang van de informatie die over hem wordt bijgehouden

Een gebruiker is zich niet altijd bewust van de omvang van de data die van hem wordt bijgehouden[8]. Dit is waarschijnlijk omdat weinig gebruikers de privacyvoorwaarden lezen vooraleer een applicatie te starten. Uit een studie [12] bij 274 studenten bleek bijvoorbeeld 55.1% de Facebook privacy policy niet gelezen te hebben. De reden hiervoor was grotendeels omdat de studenten het te veel moeite vonden (43.4%) of de policy moeilijk te begrijpen vonden(33,6%). Er is ook gewoonlijk weinig keuze, indien men de voorwaarden niet aanvaardt wordt de toegang tot de applicatie gewoonweg ontzegd.

2.4.2 Onterecht vertrouwen in de service provider

Het verkopen van data

De informatie over de ratings en voorkeuren van gebruikers is erg interessant voor marketingdoeleinden. De verkoop van deze gegevens aan derde partijen ligt vaak niet in de lijn der verwachting van de gebruikers. Om de privacy te beschermen wordt deze data vaak geanonimiseerd. Toch biedt deze anonimisatie geen volledige bescherming. Zo hebben Narayanan en Schmatikov in [13] de geanonimiseerde Netflix database kunnen deanonimiseren. Met een heel beperkte kennis van een gebruiker slaagden ze erin zijn records uit de databank te halen. Uit deze data konden ze politieke voorkeuren en andere gevoelige informatie afleiden.

Data buiten de verwachte scope [8]

De gebruiker kan ervan uitgaan dat bepaalde informatie enkel zichtbaar is voor een bepaald publiek, terwijl dit niet het geval is. Er is ook geen garantie voor gebruikers dat het personeel van het aanbevelingssysteem geen kijkje neemt in hun persoonlijke data.

Informatie op het internet is moeilijk te verwijderen. De service provider vermoeilijkt het zelf soms aangezien er commerciële waarde aan deze gegevens hangt. Het kan dus zijn dat data langer aanwezig is dan de gebruiker wil.

2.5 Preventie van inbreuken op de privacy

Nu er inzicht verkregen is welke risico's de gebruiker loopt kan er onderzocht worden wat er kan doen gebeuren inbreuken op zijn privacy te voorkomen.

2.5.1 De gebruiker informeren

Hoofdstuk 3

Onderzoek Bestaande Privacyvriendelijke Methodes

3.1 Bruikbare Encryptiesystemen

3.1.1 Homomorfe encryptie

Homomorfe encryptie is een encryptietechniek die toelaat om bewerkingen op cijferteksten uit te voeren die overeenkomen met bewerkingen op de onderliggende data van deze cijferteksten. Er wordt een onderscheid gemaakt tussen additief en multiplicatief homomorf. Additieve cryptosystemen bevatten een operatie op twee cijferteksten die overeenkomt met de som van de data. De volgende formules [6] tonen deze additieve homomorfe eigenschap bij additieve systemen op basis van vermenigvuldiging. De encryptie $\mathcal{E}$ en decryptie $\mathcal{D}$ van de berichten m_1 en m_2 gebeuren logischerwijs met de bij elkaar horende publieke en private sleutel.

$$\mathcal{D}(\mathcal{E}(m_1).\mathcal{E}(m_2)) = m1 + m2 \tag{3.1}$$

Als gevolg hiervan kan ook de vermenigvuldiging berekend worden van de data met een niet geëncrypteerd getal.

$$\mathcal{D}((\mathcal{E}(m))^a) = m.a \tag{3.2}$$

De gebruikte systemen, Paillier en DGK, ondersteunen beide de additieve homomorfe eigenschap op basis van vermenigvuldiging.

Voor de volledigheid is dit de formule voor multiplicatieve homomorfe systemen op basis van vermenigvuldiging.

$$\mathcal{D}(\mathcal{E}(m_1).\mathcal{E}(m_2)) = m1.m2 \tag{3.3}$$

Een Paillier cryptosysteem

De encryptie in een Pailliersysteem is op de volgende manier gedefinieerd [6]:

$$\mathcal{E}(m, r) = g^m.r^n mod n^2 \tag{3.4}$$

Hier is n het product van p en q, twee grote priemgetallen. De publieke sleutel wordt gevormd door (n,g) en de private sleutel door (p,q). Het gebruik van de randomwaarde r zorgt ervoor dat het Pailliersysteem semantisch veilig is. Dit betekent dat elke encryptie van dezelfde plaintext nooit resulteert in dezelfde ciphertext. Deze eigenschap komt later van pas in de privacyvriendelijke oplossing. Zoals eerder vermeld ondersteunt het Pailliercryposysteem de additieve homomorfe eigenschap op basis van vermenigvuldiging.

Een Damgard, Geisler en Kroigaard cryptosysteem (DGK)

Het DGKsysteem ondersteunt net als Paillier de additieve homomorfe eigenschap op basis van vermenigvuldiging en is net als Paillier semantisch veilig. Voor hetzelfde veiligheidsniveau heeft het echter een kleinere berichtgrootte en het is dus efficiënter dan Paillier.

3.2 Bestaande Privacyvriendelijke Methodes

3.2.1 Op basis van anonimisatie

Agent-Based aanpak door Cisse en Albayrak [3]

Deze werkwijze spitst zich toe op een Information Filtering (IF) systeem.

Het IF systeem wordt opgesplitst in drie entiteiten: de user, de provider en de filter entiteit. De entiteiten worden in de praktijk door agents vertegenwoordigd. In tegenstelling tot sommige applicaties, waar de provider en de filter door dezelfde partij worden vertegenwoordigd, is dit bij deze werkwijze niet vereist. Dit zorgt voor een meer generieke oplossing.

Het systeem houdt per entiteit rekening met het respectievelijk privacy aspect. Informatie die mogelijks gelinkt kan worden aan een gebruiker kan niet permanent opgevraagd worden door

andere entiteiten. De enige informatie die de provider permanent prijsgeeft zijn de aanbevelingen zelf. Het filteralgoritme kan niet door externe bronnen geraadpleegd worden. Private data van de entiteiten (uitgezonderd de aanbevelingen) wordt enkel tijdelijk beschikbaar gemaakt voor een andere entiteit. Dit wordt mogelijk gemaakt doordat bepaalde entiteiten de communicatie van andere entiteiten kunnen controleren.

Het basisidee is dat de user en provider entiteiten hun data sturen naar de filter entiteit. De gebruikersdata bestaat uit zijn ratings. De data van de provider bestaat uit een enorme hoeveelheid items, enkel relevante items worden naar de filter gestuurd. Deze filter entiteit berekent de aanbevelingen en verwijdert private data van beide entiteiten. De communicatie tussen de filter entiteit en de user en provider entiteiten gebeurt via een relay entiteit. Die zorgt ervoor dat de zoekopdrachten naar de provider anoniem gebeuren en dus niet gelinkt kunnen worden aan de filterentiteit die tijdelijk de persoonlijke data bevat.

In dit artikel kan de op agents gebaseerde aanpak enkel gebruikt worden voor content-based filtering of knowledge-based filtering. Om ook user-user collaborative filtering toe te laten kan een match-maker module geïntegreerd worden, beschreven in [2]. Hier heeft de provider entiteit de extra taak om de gelijkenissen tussen gebruikers te bepalen. De privacy van de gebruiker wil men garanderen door met pseudoniemen te werken aan de kant van de provider en anonieme communicatie van de provider naar de user toe.

Belangrijke kenmerken :

- Berekeningen van de aanbevelingen gebeuren in de vertrouwde filter entiteit.

Voordelen :

- Houdt ook rekening met de privacy van de provider en het gebruikte algoritme.

Nadelen:

- Anonimisatie van de zoekopdrachten door middel van relay entiteiten bij het kennis-gebaseerde aanbevelingssysteem zou kunnen leiden tot reïdentificatie van de gebruikers. Dit kan ook het geval zijn bij het gebruik van pseudoniemen in het user-user collaborative filtering systeem.

- De filter agent moet worden vertrouwd door de gebruiker en mag niet samenvallen met de provider.

- Het aanmaken van agents en bijhorende communicatieplatformen creëert extra overhead wat de performantie benadeelt. Een aanbeveling met dit systeem duurt ongeveer 10 maal zo lang als een normale aanbeveling.

3.2.2 Op basis van verstoring door randomisatie

Randomisatie aanpak op basis van Singular Value Decomposition door Polat en Du [14]

Deze werkwijze focust enkel op de privacy van de gebruiker. De gebruiker zal zijn gegevens vermommen of eerder verstoren en deze verstoorde data naar de server sturen. De verstoring gebeurt op een manier zodat de server niet de correcte persoonlijke data van de user kent maar eerder de range kent waartussen de waarden liggen. Toch kan de server op deze waarden Collaborative Filtering alsnog uitvoeren. Dit is mogelijk omdat SVD gebaseerd is op geaggregeerde gegevens en niet op individuele waarden.

Men voegt de verstoring toe in drie stappen:

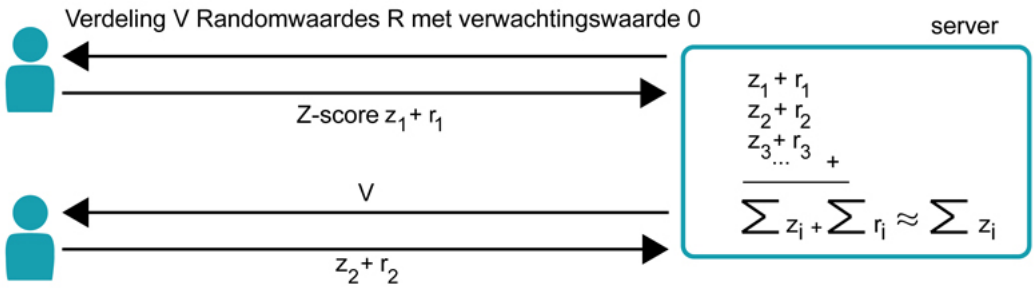
1. De server beslist welke verdeling men zal gebruiken om de data te verstoren en laat dit de gebruikers weten. Mogelijkheden zijn bijvoorbeeld een uniforme verdeling of een Gaussverdeling.

2. Elke gebruiker berekent Z-scores [1] voor de waarden van zijn rating vector vult de lege cellen met nullen.

3. De gebruiker genereert per score een randomwaarde op basis van de verdeling aangegeven door de server. Hij voegt deze aan de score toe. Het geheel van de ratings wordt per gebruiker dan doorgestuurd naar de server, deze berekent hiermee de vermomde user-item matrix A.

Om aan Singular Value Decomposition te doen heb je een user-feature matrix U en een item-feature matrix V nodig. De item-feature matrix V bereken je op basis van $A\hat{T}A$. De waarden hiervan worden verkregen door het scalair product te nemen van de rijen van de matrix $A\hat{T}$ en

[1] De Z-score of gestandaardiseerde vorm van een stochastische variabele X met verwachtingswaarde μ en standaardafwijking σ is de afwijking van zijn verwachtingswaarde, uitgedrukt in eenheden van de standaardafwijking. In formulevorm: $Z = \frac{X-\mu}{\sigma}$ De Z-score betekent een gestandaardiseerde waarde die zich met andere Z-scores laat vergelijken.



Figuur 3.1: Het basisprincipe bij een oplossing met randomisatie: bij een som over een grote groep gebruikers zal de som over de randomwaardes naar 0 gaan omdat de verwachtingswaarde van hun verdeling 0 is.

de kolommen van matrix A. Als men deze som van producten over een groot aantal gebruikers neemt dan zal de waarde van de totale random verstoring naar 0 neigen. Dit komt omdat de random waarde gekozen is uit een verdeling met verwachtingswaarde 0. Uit de eigenwaarden van $A\hat{T}A$. kan men dan de matrix V bepalen. De user-feature matrix U kan op een gelijkaardige manier benaderd worden met een som over de items. Deze aanpak wordt duidelijk nauwkeuriger naarmate er meer gebruikers en items zijn aangezien de impact van de willekeurig gekozen waarden uit de verdeling meer naar 0 zal neigen bij grotere sommaties.

Hoe meer verstoring, hoe meer privacy de gebruiker heeft maar ook hoe hoger de MAE-waarde en dus hoe slechter de aanbevelingen. Uit tests op de MovieLens en Jester dataset lijkt een keuze van randomwaarden uit de Gaussverdeling met standaarddeviatie 1 de beste resultaten te geven. Bij 100 gebruikers in de MovieLens dataset is de MAE zonder verstoring ongeveer 0.71 en met verstoring 0.78. Omdat de MovieLens database met een ratingsysteem werkt van 1 tot 5 sterren geeft het kleine verschil van 0.07 in de MAE aan dat de kwaliteit van de aanbevelingen niet veel inboet door de aangebrachte verstoring. De resultaten zijn bij nog meer gebruikers zoals verwacht zelfs nog beter. De Jester dataset gaf gelijkaardige resultaten.

Belangrijke kenmerken :

- Berekeningen van de aanbevelingen gebeuren op de server.

Voordelen :

- Geen berekeningen aan client-zijde.

Nadelen:

- Biedt geen volledige privacy, de provider heeft nog steeds een idee in welke range de ratings liggen afhankelijk van de gebruikte verstoring.
- Werkt minder nauwkeurig met kleinere datasets.

3.2.3 Op basis van verstoring door aggregatie

Aanpak door het gedistribueerd aggregeren van offline profielen door Shokri et al. [15]

In tegenstelling tot de oplossing met randomisatie worden in deze methode niet de aparte ratings verstoord maar eerder de volledige profielen. De service provider berekent dan aanbevelingen op basis van deze verstoorde profielen.

Een gebruiker heeft drie verschillende profielen. Het eerste profiel wordt bijgehouden aan de kant van de gebruiker en is enkel opgesteld met ratings geleverd door de gebruiker zelf. Dit profiel wordt nooit opgevraagd door de service provider. Een tweede profiel, het offline profiel genoemd, is een kopie van het eerste profiel uitgebreid met ratings uit offline profielen van andere gebruikers. Het online profiel dat bijgehouden wordt aan de kant van de server is eigenlijk een kopie van het offline profiel en haalt periodiek updates op van de client. Op basis van dit profiel zal de server zijn aanbevelingen berekenen.

Men kan gebruikers en items voorstellen als nodes in een graaf en ratings als verbindingen met gewicht tussen item en gebruiker. De client kan op een arbitraire manier contact met andere clients leggen. Bij een contact delen clients ratings van hun offline profielen. Hierbij kan de client vanzelfsprekend zijn eigen ratings behouden. Omdat deze interactie enkel tussen clients gebeurt heeft de server geen weet van welke clients onderling ratings delen. Hij weet dus ook niet welke ratings van het offline profiel van de gebruiker zelf zijn en welke van een andere client afkomstig zijn. Aangezien de clients onderling enkel gegevens van hun offline profiel delen geldt hetzelfde tussen gebruikers en is er dus ook een graad van user-user privacy. Het kan dan ook gebeuren dat de server items aanbeveelt die de gebruiker al beoordeeld heeft. Het is aan de clientapplicatie zelf om hierin te schiften.

Er moet vooraf bepaald worden welke ratings gedeeld worden tussen twee gebruikers en hoeveel. Shokri et al. geven hier verschillende opties voor. Alle ratings, een vast aantal ratings of het aantal ratings laten afhangen van de gelijkenis van twee gebruikers zijn mogelijkheden. De gelijkenis tussen twee gebruikers wordt privacyvriendelijk berekend met een methode van Lathia et al. [10]. Welke ratings worden gebruikt kan willekeurig gekozen worden of er kan een

voorkeur gegeven worden aan de items die het minst geratet zijn over het hele systeem. Een onderzoek van Narayanan en Shmatikov [**?**] geeft aan dat een rating van een item, dat weinig beoordeeld is over alle users heen, sneller tot identificatie van een gebruiker kan leiden. De optie om net deze ratings te delen tussen gebruikers heeft dus een positieve impact op de privacy.

De privacymaat wordt berekend op basis van de gelijkenis tussen de verstoorde online graaf en de originele graaf rekening houdende met aantal ratings per item. Als er gegevens met zes gebruikers per jaar uitgewisseld worden kan dit leiden naar een significante privacywinst van 0.68 de gebruiker naar de provider toe. bij een accuraatheidsverlies van 2% .

Belangrijke kenmerken :

- Deze methode is dan wel privacyvriendelijker dan de verstoring met randomisatie maar heeft toch zijn beperkingen.

Voordelen :

- In tegenstelling tot de oplossing met randomisatie heeft hier de service provider geen idee welke items door de user zelf beoordeeld zijn.
- Weinig accuraatheidsverlies

Nadelen:

- Ratings van een offline profiel die lange tijd gelijk blijven hebben meer kans van die user zelf te zijn.
- Het systeem krijgt de originele ratings van de gebruiker.

3.2.4 Met behulp van cryptografische protocollen zonder server

Hier is het uitgangspunt dat er geen vertrouwen meer nodig is in een provider indien de provider niet betrokken is bij het berekenen van aanbevelingen. Men werkt dus op een *peer-to-peer* basis. De privacy van de gebruikers onderling wordt gerespecteerd door middel van secure multi-party computation.

Een voorbeeld is uitgewerkt door Hoens et al.

Aanpak op basis van cryptografische protocollen zonder server met behulp van een sociaal netwerk door Hoens et al. [7]

Deze methode benut de vriendschapsrelaties op sociale netwerken. Men stelt dat aanbevelingen berekenen aan de hand van een sociaal netwerk betere resultaten levert dan een algemene aanpak

omdat er gelijkenissen zijn tussen de smaak van een persoon en de smaak van zijn netwerk (vrienden, vrienden van vrienden,..). Een score voor een item wordt berekend door een gewogen gemiddelde te bepalen van de ratings van het netwerk van die persoon. Een optie is om hierbij de ratings van de gebruikers die dichter bij de gebruiker staan meer gewicht te geven. Tijdens de berekening van de score mag een gebruiker geen private informatie over het ratingsgedrag van een andere gebruiker te weten komen.

De vrienden van een gebruiker worden in deze context bekeken als zijn onmiddellijke kinderen in de boom. Eerst worden zijn onmiddellijke kinderen in de boom naar hun rating gevraagd om er een gewogen gemiddelde van te bepalen. Zij berekenen op hun beurt een rating op basis van een gewogen gemiddelde van hun kinderen enzovoort. Dit gebeurt tot de gewenste diepte bereikt is. De gebruiker kan eventueel zelf instellen hoe diep er mag worden afgedaald in de vriendschapsboom. Dit systeem heeft als nadeel dat bij n interactieronde elke gebruiker online moet zijn op het sociaal netwerk.

Bij dit proces mag enkel de vragende gebruiker het voor hem berekende eindresultaat te weten komen. Andere tussenresultaten mogen noch door de aanvrager zelf of zijn netwerk leesbaar zijn. Om dit te bereiken worden verschillende cryptografische protocollen gebruikt.

Homomorfische encryptie lijkt een logische keuze om de optelling/vermenigvuldiging te doen van de verschillende waarden vereist voor het berekenen van het gewogen gemiddelde. Daarnaast bestaat een (t,n)-threshold encryption, een encryptie die ervoor zorgt dat er medewerking van minstens t partijen nodig is om een waarde te decrypteren. Elke partij krijgt hierbij een deel van de sleutel. Omdat geen enkele partij volledig vertrouwd wordt zal het genereren van de sleutel ook moeten verdeeld worden tussen de partijen. Om aan homomorfische en threshold encryptie te voldoen kiest men ervoor gebruik te maken van het Paillier cryptosysteem. Voor de deling uit te voeren bij de berekening van het gewogen gemiddelde wordt een nieuw protocol gebruikt.

Belangrijke kenmerken :

- Enkel bruikbaar voor Collaborative filtering
- Berekeningen gebeuren bij de gebruiker zelf

Voordelen :

- Gebruikt vriendschapsrelaties van sociale netwerken

- Provider kan helemaal niets weten van private data gebruikers

Nadelen:

- Niet elk aanbevelingssysteem beschikt over een achterliggend sociaal netwerk.
- Er gebeuren veel berekeningen bij de gebruiker. Dit is bij een mobiel toestel een belangrijk minpunt aangezien intensief cpugebruik liever vermeden wordt.
- De oplossing gaat ervan uit dat andere gebruikers ook online zijn, wat voor vertragingen kan zorgen.

3.2.5 Met behulp van cryptografische protocollen met server

Aanpak met behulp van cryptografische protocollen door Erkin et al [4].

Deze aanpak voert de verschillende stappen in het collaborative filteringproces privacyvriendelijk uit met behulp van verschillende protocollen. Om deze protocollen te kunnen uitvoeren genereert elke gebruiker eerst een sleutelpaar voor een Paillier en een DGK-cryptosysteem. Een eerste stap in dit proces is het bepalen van gelijkaardige gebruikers. Dit gebeurt aan de hand van de gekende Pearson-correlatie op ratings van items in een bepaalde range R. De Pearson-correlatie berekent gelijkenis tussen twee gebruikers aan de hand van de cosinus tussen hun smaakvectoren. De ratings in de range R noemt men de voorkeurenvector . De Pearson-correlatie (3.5) kan in 2 delen worden opgedeeld[6]:

$$sim_{A,B} = \frac{\sum_{j=0}^{R-1}(v_{(A,i)} - \bar{v}_A).(v_{(B,i)} - \bar{v}_B)}{\sqrt{\sum_{j=0}^{R-1}(v_{(A,j)} - \bar{v}_A)^2 . \sum_{j=0}^{R-1}(v_{(B,j)} - \bar{v}_B)^2}} \tag{3.5}$$

$$sim_{A,B} = \sum_{i=0}^{R-1} C_{A,i}.C_{B,i} \tag{3.6}$$

$$C_{X,i} = \frac{(v_{(X,i)} - \bar{v}_X)}{\sqrt{\sum_{j=0}^{R-1}(v_{(X,j)} - \bar{v}_X)^2}} \tag{3.7}$$

Zo kunnen twee gebruikers apart de C-waarden berekenen. De gebruiker die aanbevelingen vraagt encrypteert zijn voorkeurenvector met zijn public key. Dit stuurt hij naar de server die dit op zijn beurt doorstuurt naar een andere gebruiker. Door middel van het Paillier cryptosysteem kan deze de waarden van zijn voorkeurenvector respectievelijk vermenigvuldigen met de waarden voor de zelfde items van de voorkeurenvector van de aanvrager. Van al deze producten wordt

de som genomen (4.1) zoals in de Pearsoncorrelatie. De som geeft de gelijkenis weer tussen de twee gebruikers die nog steeds is geëncrypteerd met de publieke sleutel van de aanvrager. Deze waarde wordt dan teruggestuurd naar de server. De server voegt de ontvangen gelijkeniswaarden allemaal samen in een vector en voert hierop dan een aangepast protocol met de aanvrager. Dit aangepast protocol maakt gebruik van een DGK cryptosysteem in plaats van een Paillier cryptosysteem voor efficiëntieredenen. Het resultaat is een gencrypteerde vector van enen en nullen die aangeeft of de gelijkeniswaarde tussen de aanvrager en de respectievelijke gebruiker boven een bepaalde ondergrens ligt. De server stuurt deze naar de gebruiker zodat hij zijn ratings kan vermenigvuldigen met deze waarde en terugsturen. Hierop maakt de server de som van de ratings over alle gelijkaardige gebruikers en stuurt deze naar de aanvrager die de som decrypteert en er een gemiddelde van berekent.

Met behulp van dit protocol worden enkel de ratings van gelijkaardige gebruikers gebruikt. Geen enkele entiteit kan de gelijkenis tussen twee gebruikers bepalen.

Belangrijke kenmerken :

- Berekeningen van de aanbevelingen gebeuren door de server en de client.

Voordelen :

- Dankzij diverse cryptografische protocollen is dit een volledig privacyvriendelijke oplossing in een statische setting

Nadelen:

- Bij een dynamische userdatabase kan deze oplossing gegevens van de gebruikers lekken. Als een zelfde gebruiker tweemaal aanbevelingen vraagt met de tweede maal één gebruiker meer kan hij afleiden of deze extra gebruiker een gelijkaardige smaak heeft als hijzelf, afhankelijk van of deze gebruiker een invloed heeft op zijn aanbevelingen of niet. Men spreekt over een "*new group attack*"[9].
- Cryptografie zorgt voor een overhead.

3.2.6 Conclusie onderzoek

Nu er van de belangrijkste soorten oplossingen zijn doorlicht, kan er besloten worden in welke richting we verderwerken. Een eerste optie met behulp van randomisatie leek op het eerste zicht

erg interessant omdat deze heel weinig berekeningen aan de clientkant vraagt, wat natuurlijk een pluspunt is bij mobiele toestellen. Helaas werkt dit systeem niet goed bij lage gebruikersaantallen en kan het informatie lekken aan de service provider afhankelijk van de gekozen verdeling. Daar heeft de verstoring met behulp van aggregatie geen last van. Deze biedt qua privacy wel deniability of preferences aan de user maar geeft de provider wel toegang tot originele beoordelingen van de gebruiker. Als Narayan en Schmatikov in [13] anonieme profielen kunnen linken met users met maar een beperkte kennis over deze gebruiker, lijkt het maar een kleine stap om uit het aangegeven verstoorde profiel de juiste ratings bij een gebruiker te plaatsen. Zeker indien er maar wordt geaggregeerd met een beperkt aantal andere gebruikersprofielen zoals in [15] beschreven. Hoewel de oplossing op basis van een sociaal netwerk erg privacyvriendelijk is, is ze niet echt flexibel. Ze vereist dat andere gebruikers online zijn en ook actief meewerken aan het protocol. Door het *peer-to-peer* character van deze werkwijze zijn de clients verplicht om alle berekeningen zelf te doen. Het is duidelijk dat deze eigenschappen in een mobiele setting niet gewenst zijn. Bij de oplossing met anonimisatie heeft de clientapplicatie daarentegen weinig werk. Het stuurt de beoordelingen rechtstreeks naar de filterentiteit, die moet vertrouwd worden. De anonimisatie die gebeurt in het user-user collaborative filtering systeem of de kennisgebaseerde versie voldoet niet om volledig privacyvriendelijk te zijn zoals aangegeven in 2.4.2. De meest privacyvriendelijke oplossing lijkt de methode met behulp van cryptografische protocollen en een server. Het systeem werkt er volledig privacyvriendelijk in een statische setting. Uit verder onderzoek blijkt dat dezelfde auteur ook een paper uitbracht in november 2013 [1] die een erg gelijkaardige oplossing aanbiedt die ook privacyvriendelijk is in een dynamische setting. Deze methode heeft als extra voordeel dat de gebruikersapplicatie relatief weinig berekeningen hoeft te doen. Het enige minpunt is dat de berekeningen aan de serverkant erg zwaar zijn door de cryptografische protocollen. We besluiten voor deze werkwijze te kiezen en deze verder te optimaliseren in een mobiele setting in hoofdstuk 4.

Hoofdstuk 4

Privacyvriendelijk aanbevelingssysteem voor mobiele toestellen

4.1 Inleiding

De keuze voor vertrekpunt van onze oplossing viel op de werkwijze van Z. Erkin, T. Veugen en R.L. Lagendijk beschreven in "Privacy-preserving recommender systems in dynamic environments [1]". Deze laat user-user collaborative filtering toe op een privacyvriendelijke manier. Kort samengevat gaat deze een voorspelling van een beoordeling doen van een gebruiker A door de gemiddelde rating te nemen van gebruikers die een gelijkenis hebben met A die hoger ligt dan een bepaalde drempelwaarde. De basis van de werkwijze uit [1] is gebleven, maar er werden delen aangepast en eigen accenten gelegd. Net als het besproken systeem in paragraaf 3.2.5 maakt dit systeem gebruik van cryptografische protocollen met behulp van een Paillier en een DGK-cryptosysteem. Technieken gebaseerd op cryptografie hebben het voordeel dat de privacy bewijsbaar is als de gebruikersdata wordt geëncrypteerd. De service provider kan dan door de homomorfische eigenschappen van de cryptosystemen zijn taken verrichten zonder toegang te krijgen tot deze data. Buiten de gebruikersvoorkeuren worden ook de gelijkeniswaarden tussen gebruikers en de uiteindelijke aanbevelingen afgeschermd van de service provider door encryptie. Om problemen te voorkomen in een dynamische setting zoals in 3.2.5 wordt ook de informatie verminderd waarover de provider beschikt bij herhaaldelijke aanvragen van aanbevelingen. Om dit te verwezenlijken wordt er gebruik gemaakt van een tweede server, een server die onafhanke-

lijk moet zijn van de server van de service provider. Deze tweede server wordt in dit naslagwerk ook vernoemd als controle server. De controle server gedraagt zich als privacy provider. Deze rol kan bijvoorbeeld vervuld worden door de overheid of een ander commercieel bedrijf. Het gebruik ervan zou opgelegd kunnen worden door de overheid in bepaalde privacygevoelige sectoren zoals bijvoorbeeld een systeem gebaseerd op medische data. Er wordt vanuit gegaan dat beide servers optreden in een "*semi-honest*"model, wat betekent dat ze betrouwbaar genoeg zijn om het onderling protocol in volgorde uit te voeren.
Beide servers beschikken over een Paillier-sleutelpaar, hiernaast heeft de controleserver ook een DGK-sleutelpaar. Zoals eerder vermeld werd er gekozen om de clientapplicatie uit te werken in Android en deze communiceert met de Javaservers via het HTTP-protocol. We gebruiken de MovieLens database, de standaard voor het testen van aanbevelingssystemen.

4.2 Werking

4.2.1 Diagram van het hoofdprotocol

Het volgend sequentiediagram geeft de communicatie weer tussen de clientapplicaties en de twee servers als een gebruiker aanbevelingen vraagt. De paragrafen waar de onderdelen ervan besproken worden staan links aangegeven. Functies, protocollen en taken staan in het zwart aangegeven, attributen in het blauw. De berekeningen op de aanbevelingsserver gebeuren in het gesloten Pailliersysteem getekend met publieke sleutel S2 zodat de server geen persoonlijke data kan lezen.

Alvorens het aanvragen van aanbevelingen slaat een gebruiker zijn beoordelingen en voorkeuren op en stuurt ze dan samen in zijn profiel naar de recommenderserver (4.2.2). Eens user 1 aanbevelingen vraagt is de eerste stap de gelijkenisfactor berekenen tussen user 1 en de andere users Ux. Dit gebeurt door de aanbevelingsserver in paragraaf 4.2.3. Daarna gaat de server met het thresholdprotocol (4.2.4) na of deze geëncrypteerde waarden boven een drempelwaarde liggen. Nu bezit de recommender een geëncrypteerde bit per gebruiker die aangeeft of dit wel (1) of niet (0) het geval is. Hierop starten de recommender en de privacy provider een nieuw protocol waarbij de controleserver een deel van de gebruikers selecteert die kunnen meedoen aan het protocol (4.2.5). De resultaatbits hiervan geven aan of de ratings van respectievelijke gebruiker zullen worden meegeteld in het berekenen van de aanbevelingen. Het multiplicationprotocol (4.2.6) vermenigvuldigt deze gebruikerbits met zijn overeenkomstige ratings. Daarna wordt de

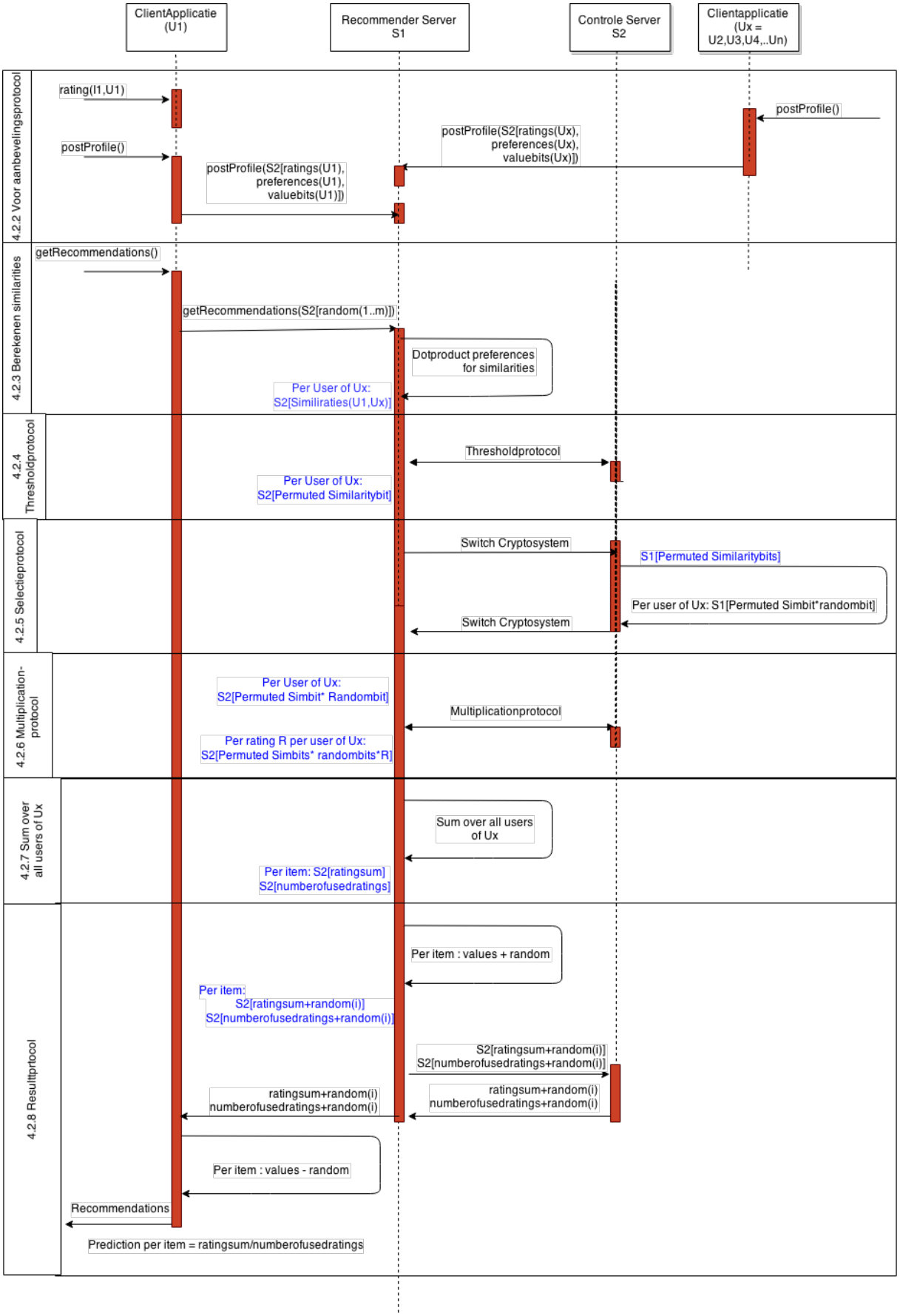
som van de ratings per item genomen over alle gebruikers heen in paragraaf 4.2.7 en wordt het resultaat privacyvriendelijk naar de gebruiker gebracht (4.2.8).

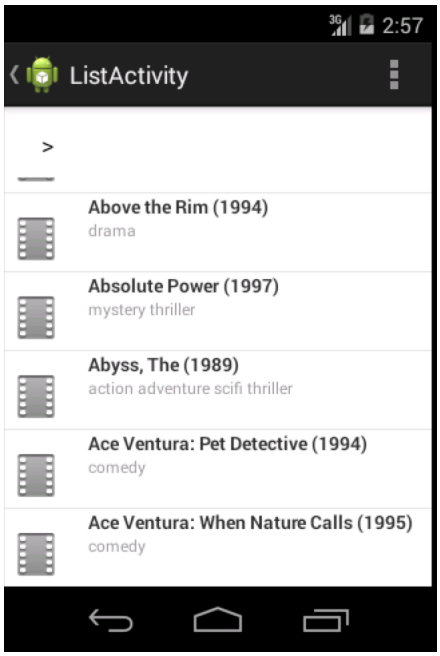
4.2.2 Voor een aanbevelingsaanvraag

Voor een aanbevelingsaanvraag zal de gebruiker verschillende items beoordelen. De ratings worden lokaal bijgehouden in de Androidapplicatie in een SQLiteDatabase. Ze worden daarna voor het sturen van het profiel geëncrypteerd met de publieke sleutel van het Pailliersysteem van de controleserver en naar de recommenderserver gestuurd.

Om de gelijkenis te berekenen tussen twee gebruikers wordt er dezelfde methode gebruikt als in 3.2.5 en wordt de Pearson-coëfficient dus opgesplitst (4.1).
Per gebruiker moeten dus de *preferences* of C-waarden $C_{X,i}$ bepaald worden. De preferences worden bepaald op basis van de ratings van een bepaalde deelverzameling van de items. Deze deelverzameling wordt het best bepaald door items te nemen die door het gros van de gebruikers beoordeeld zijn. Om deze lijst dynamisch te bepalen zou de recommenderserver moeten weten hoeveel gebruikers een bepaald item geëvalueerd hebben. Dat kan niet dynamisch bepaald worden gezien het volledig privacyvriendelijk karakter van dit systeem. Hierdoor mag de server niet weten welke gebruiker welk item beoordeeld heeft, zelfs al weet de server de waarde van de beoordeling niet. Dit zou contact kunnen verraden tussen een user en een item, bijvoorbeeld dat een persoon naar een restaurant geweest is. Het systeem zou met deze data zelfs een smakenprofiel van een gebruiker opzetten. Een oplossing voor dit probleem is de deelverzameling vast kiezen en eventueel bij het eerste gebruik aan de gebruiker vragen deze lijst te evalueren, zonder hem te verplichten een bepaald item te raten. Deze deelverzameling zou dan best opgesteld worden uit items waarover de mening van de gebruikers verdeeld is en niet diegene die de meeste users goed of slecht vinden. Om deze complicaties te vermijden gebruiken we in onze testapplicatie de ratings van alle items als preferences.

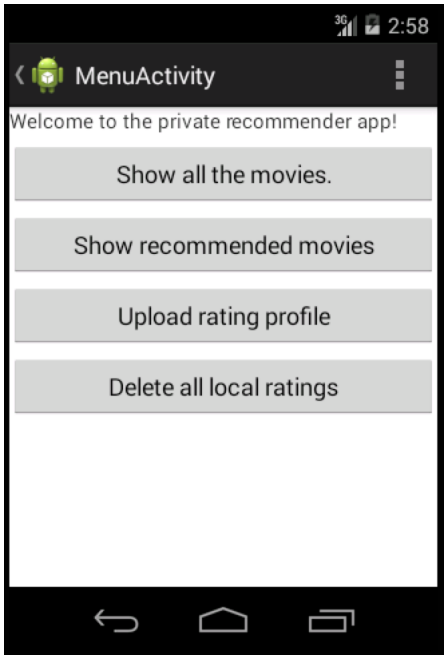
De $\bar{v}_X$ en $\sqrt{\sum_{j=0}^{M-1}(v_{(X,j)} - \bar{v}_X)^2}$ waarden zijn voor elke C-waarde gelijk. Dit betekent dat je de C-waarde eventueel ook aan de kant van de server zou kunnen berekenen en voor de deling in het Pailliersysteem gebruik maken van een protocol als [16]. Dit lijkt niet zo efficiënt en er werd besloten de berekening van de C-waarden te doen op de client en geëncrypteerd mee te sturen in het profiel. Naast de lijsten van de ratings en de preferences geven we ook nog een lijst geëncrypteerde bits mee die aangeven of een item wel (1) of niet (0) beoordeeld is door


ClientApplicatie (U1)
Recommender Server S1
Controle Server S2
Clientapplicatie (Ux = U2,U3,U4,..Un)
4.2.2 Voor aanbevelingsprotocol
rating(I1,U1)
postProfile()
postProfile(S2[ratings(U1), preferences(U1), valuebits(U1)])
postProfile()
postProfile(S2[ratings(Ux), preferences(Ux), valuebits(Ux)])
4.2.3 Berekenen similarities
getRecommendations()
getRecommendations(S2[random(1..m)])
Dotproduct preferences for similarities
Per User of Ux: S2[Similiraties(U1,Ux)]
4.2.4 Thresholdprotocol
Thresholdprotocol
Per User of Ux: S2[Permuted Similaritybit]
4.2.5 Selectieprotocol
Switch Cryptosystem
S1[Permuted Similaritybits]
Per user of Ux: S1[Permuted Simbit*randombit]
Switch Cryptosystem
4.2.6 Multiplication-protocol
Per User of Ux: S2[Permuted Simbit* Randombit]
Multiplicationprotocol
Per rating R per user of Ux: S2[Permuted Simbits* randombits*R]
4.2.7 Sum over all users of Ux
Sum over all users of Ux
Per item: S2[ratingsum] S2[numberofusedratings]
4.2.8 Resultprotocol
Per item : values + random
Per item: S2[ratingsum+random(i)] S2[numberofusedratings+random(i)]
S2[ratingsum+random(i)] S2[numberofusedratings+random(i)]
ratingsum+random(i) numberofusedratings+random(i)
ratingsum+random(i) numberofusedratings+random(i)
Per item : values - random
Recommendations
Prediction per item = ratingsum/numberofusedratings



Figuur 4.1: De gebruiker krijgt de lijst met films te zien uit de MovieLensdatabase

Figuur 4.2: De gebruiker beoordeelt een film.



Figuur 4.3: Het menu van de testapplicatie. De gebruiker kan zijn profiel uploaden wanneer hij wil.

deze gebruiker. Dit lijkt redundante informatie maar dit kunnen we gebruiken als we moeten optellen hoeveel (ingevulde) ratings er werden opgeteld. Dat komt omdat als we het profiel naar de server sturen ook "ratings"moeten doorsturen van items die de persoon nog niet beoordeeld heeft. Dit om te verbergen welke items de persoon geëvalueerd heeft. Dit is een uitbreiding op de theoretische paper [1] waar er vanuit gegaan wordt dat alle ratings ingevuld zijn. We sturen ook best de waarden van alle items mee, anders zitten er C-waarden in de databank op de server berekend op verourde $\bar{v}_X$ en $\sqrt{\sum_{j=0}^{M-1}(v_{(X,j)} - \bar{v}_X)^2}$ waarden. Deze manier geeft ook de garantie op de hoogste privacy.

We geven ook de gebruiker de keuze wanneer hij zijn profiel uploadt en hij kan ook zijn lokale ratings leegmaken. Dit geeft hem meer controle over zijn profiel en laat hem toe om zijn profiel eerst leeg te maken en dan up te loaden. Zo heeft de aanbevelingsserver geen enkele geëncrypteerde beoordeling meer staan van deze gebruiker, tenzij hij natuurlijk backups nam. Het bespaart ook data-overdracht en processortijd als de upload niet bij elke rating gebeurt.

Nu moet er de keuze gemaakt worden wanneer de encryptie gebeurt. De encryptie gebeurt optimaal op het moment dat het profiel wordt gestuurd naar de aanbevelingsserver en niet alvorens de rating in de databank wordt geplaatst. Dit is aan de ene kant logisch aangezien

ook de C-waarden dan het best berekend worden met de laatste waarden, maar zorgt er ook voor dat alle geëncrypteerde waarden telkens anders zijn zodat de server niet kan zien welke waarden veranderd zijn ten opzichte van een vorige rating dankzij de semantische veiligheid van het Paillier-cryptosysteem.

4.2.3 Berekenen van gelijkeniswaarden tussen gebruikers

Het berekenen van gelijkeniswaarden gebeurt zoals eerder aangegeven door de Pearson-coëfficient. Aangezien de geëncrypteerde C-waarden hiervoor al op de clientapplicatie worden berekend hoeven we enkel nog het scalair product te nemen van deze waarden met de formule $sim_{A,B} = \sum_{i=0}^{R-1} C_{A,i}.C_{B,i}$ uit paragraaf 3.2.5. Geëncrypteerd met de publieke sleutel van de controleserver wordt dit (naar [1]):

$$[sim_{A,B}]_2 = \prod_{i=0}^{R-1} [C_{A,i}]_2 \otimes [C_{B,i}]_2 \tag{4.1}$$

De som wordt omgezet in een product dankzij de homomorfe eigenschappen van het Paillier-systeem (3.1.1). Het product tussen twee geëncrypteerde waarden uitrekenen, hier aangegeven met $\otimes$, is niet zo eenvoudig. Hiervoor wordt dezelfde werkwijze gebruikt als in het multiplicatieprotocol (4.2.6).

4.2.4 Thresholdprotocol

Dit protocol uit Privacy-Preserving Face Recognition [5] laat toe om twee geëncrypteerde waarden $[a]_2$ en $[b]_2$ te vergelijken met als resultaat een geëncrypteerde bit die $[1]_2$ is als $a \geq b$ als en $[0]_2$ is als $a < b$. We gebruiken dit om een gelijkeniswaarde sim tussen de gebruikers te verlijken met een drempelwaarde d. De drempelwaarde kiest de recommender zelf afhankelijk van de dataset en encrypteert hij dus met de publieke Paillierssleutel van de controleserver.

Het aanbevelingssysteem berekent eerst $[z]_2$

$$[z]_2 = [2^l + sim - d]_2 = [2^l]_2.[sim]_2.[d]_2 \tag{4.2}$$

Als we het binair bekijken geeft de grootste bit van $[z]_2$ inderdaad de verhouding van sim ten opzichte van d aan. Deze bit is de l-de bit (geteld vanaf rechts beginnend bij 0) en wordt z_l genoemd. De waarde van z_l kan als volgt berekend worden.

$$z_l = 2^{-l}.(z - (zmod2^l)) \tag{4.3}$$

Hier worden eerst alle bits behalve de meest linkse op 0 gezet door $z - (zmod2^l)$ en wordt de bit dan naar rechts geschoven door te vermenigvuldigen met 2^{-l}. De aanbevelingsserver beschikt

niet over $[zmod2^l]_2$ en zal dus hulp moeten vragen aan de controleserver, die de waarde van z niet mag kennen. De waarde van z wordt dus eerst vermomd met een randomwaarde r.

$$[c]_2 = [z + r]_2 = [z]_2.[r]_2 \tag{4.4}$$

De c-waarde wordt dan naar de controleserver gestuurd die c uitpakt en $cmod2^l$ berekent. Hij stuurt $cmod2^l$ op twee manieren terug. Eń maal geheel Pailliergeëncrypteerd en éénmaal opgesplitst in bits DGKgeëncrypteerd (zie straks). Omdat $[c]_2 = [z + r]_2 = [z]_2.[r]_2$ geldt dat:

$$zmod2^l = (cmod2^l - rmod2^l)mod2^l \tag{4.5}$$

Dus afhankelijk van of $cmod2^l < rmod2^l$ moeten we bij $cmod2^l - rmod2^l$ wel of niet 2^l bijtellen. Dit is het zogenaamde millionairsprobleem van Yao waarbij twee partijen beschikken over een getal dat ze niet aan de andere partij bekend willen maken maar ze willen wel weten welke het grootste is. Hiervoor worden de e_i-bits (i staat voor de plaats van de bit) berekend, waarvoor we de bits van $cmod2^l$ en $rmod2^l$ gebruiken. Er wordt voor dit subprotocol een DGKsysteem gebruikt voor efficiëntieredenen. De recommender kiest een s-waarde 1 of -1 die aangeeft in welke richting de vergelijking gebeurt. De DGKencryptie geven we aan met dubbele haakjes [[]].

$$[[e_i]] = [[sim_i - r_i + s + 3. \sum_{j=i+1}^{l-1} (sim_j \oplus r_j)]] \tag{4.6}$$

Eens we alle e_i-bits berekend hebben kunnen we de γ-bit bepalen. Deze is 0 als er n e_i-bit 0 is en anders 1.

- Als s=1 is en $\gamma = 1$ dan is $sim > r$, is $\gamma = 0$ dan is $sim < r$.
- Als s=-1 is en $\gamma = 1$ dan is $sim < r$, is $\gamma = 0$ dan is $sim > r$.

Het aanbevelingssysteem stuurt de e_i-bits gepermuteerd naar de controleserver. Die decrypteert ze en berekent de γ-bit, hij weet niet wat deze betekent aangezien hij de waarde van s niet kent. Hij encrypteert de γ-bit met zijn publieke Pailliersleutel en stuurt deze terug naar de recommenderserver. De recommenderserver kent de waarde van s en weet dus wat deze geëncrypteerde γ-bit betekent. Hij kan dus $[\lambda]_2$ berekenen de bit die aangeeft of $cmod2^l < rmod2^l$ en dus of 2^l moet opgeteld worden bij $cmod2^l - rmod2^l$ in 4.5. Als $s = 1$ heeft $\lambda = 1$ de omgekeerde bitwaarde als γ. De omgekeerde bitwaarde kan berekend worden door $\gamma \oplus 1$. De XOR functie wordt nagebootst met $[a \oplus b]_2 = [a + b - 2ab]_2 = [a]_2 * [b]_2 * [a]_2^(- 2b)$ Als $s = -1$ is $\lambda = \gamma$. Hierdoor heeft de recommender dus genoeg info om 4.5 te berekenen en daarna 4.3.

4.2.5 Selectie maken van gebruikers

Na het meerdere keren aanroepen van het aanbevelingssysteem, zou het systeem of een gebruiker kunnen afleiden de data van welke gebruikers welke invloed heeft op de aanbevelingen. Dit heet een new group attack en is eerder al vernoemd in 3.2.5. Om een new group attack tegen te gaan wordt er door de controleserver een selectie van gebruikers gemaakt die kunnen deelnemen aan het protocol. De recommender kent deze selectie dus niet. De recommender zou daarvoor de gelijkenisbits naar de controleserver moeten sturen. De bits zijn nog geëncrypteerd met de publieke sleutel van de controleserver, die deze bits niet mag kennen. Om dit op te lossen worden er randomgetallen $[\rho_i]_1$ bij de geëncrypteerde simbits geteld. Het randomgetal wordt zelf ook meegestuurd, geëncrypteerd met de publieke sleutel van de recommenderserver. Zo kan de controleserver $[sim_i + \rho_i]_2$ decrypteren met zijn eigen private sleutel. Deze som encrypteert hij dan weer met de publieke sleutel van de recommenderserver. Op die manier kan hij de randomwaarde hiervan aftrekken met het Pailliersysteem $[sim_i + \rho_i - \rho_i]_1$ en de originele gelijkenisbits verkrijgen. De controleserver maakt nu een tabel aan met bits met evenveel 1-waarden als gebruikers die hij wil betrekken in het protocol. In [1] wordt een waarde $n/2$ voorgesteld voor voldoende privacy. De server vermenigvuldigt deze bits met de gelijkenisbits en stuurt het resultaat terug naar de aanbevelingsserver. Hierbij wordt nogmaals gewisseld van cryptosysteem zoals bij het heensturen van de bits.

4.2.6 Multiplicatieprotocol

Hier vermenigvuldigt de recommenderserver de ratings en bitvalues van de gebruiker met hun resultaatbits uit 4.2.5 die aangeven of hun beoordeling worden opgenomen in het protocol. De vermenigvuldiging van twee geëncrypteerde waarden zit niet standaard in het Pailliercryptosysteem. Dit wordt opgelost door dit eenvoudig subprotocol [6]. De aanbevelingsserver bepaalt per vermenigvuldiging eerst twee random getallen r_1 en r_2 die hij aftrekt van de respectievelijke bits en zo $[sim_x - r_1]_2$ en $[rating_y - r_2]_2$ bepaalt. Hij stuurt deze naar de controleserver die deze decrypteert, vermenigvuldigt, weer encrypteert met zijn publieke sleutel en terugstuurt. Zo kan de recommenderserver het resultaat van de vermenigvuldiging berekenen door:

$$[sim_x.rating_y]_2 = [(sim_x - r_1).(rating_y - r_2) + sim_x.r_2 + rating_y.r_1 - r_1.r_2]_2 \tag{4.7}$$

$$[sim_x.rating_y]_2 = [(sim_x - r_1).(rating_y - r_2)]_2.[sim_x]_2^{r_2}.[rating_y]_2^{r_1}.[-r_1.r_2]_2 \tag{4.8}$$

Deze berekeningen zijn allemaal mogelijk in het Pailiercryptosysteem.

4.2.7 Som over alle gebruikers heen

In dit deel worden de waarden van de ratings en bitvalues (beide vermenigvuldigd met de resultaatbit) samengeteld per item over alle gebruikers heen. Op deze manier beschikt de recommenderserver over een geëncrypteerde som van de ratings die worden meegeteld en het geëncrypteerde aantal ratings dat wordt meegeteld.

4.2.8 Resultaatprotocol

Figuur 4.4: De gebruiker krijgt een lijst met aanbevelingen en een voorspelling van het aantal sterren dat hij aan deze film.

Het resultaatprotocol zorgt ervoor dat de twee berekende waarden per item uit 4.2.7, de som der ratings en het aantal, op een privacyvriendelijke manier bij de aanvrager terechtkomen. Bij het aanvragen van aanbevelingen gaf de clientapplicatie randomwaarden mee. Deze worden bij de waarden geteld en naar de controleserver gestuurd die ze decrypteert en terugstuurt naar de recommender, die op zijn beurt ze gewoon teruggeeft aan de gebruiker. De gebruiker trekt er zijn gegenereerde randomwaarden van af en verkrijgt de originele waarden. Hierop bepaalt hij de gemiddelde rating van gelijkaardige gebruikers door ze eenvoudigweg te delen.

De clientapplicatie is wel verantwoordelijk om de items die reeds door de gebruiker zelf zijn beoordeeld uit de resultaten te filteren.

Bibliografie

[1] *Privacy-Preserving Recommender Systems in Dynamic Environments*, 11/2013 2013.

[2] Richard Cissée. *An agent-based approach for privacy-preserving information filtering.* PhD thesis, 2009.

[3] Richard Cissée and Sahin Albayrak. An agent-based approach for privacy-preserving recommender systems. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '07, pages 182:1–182:8, New York, NY, USA, 2007. ACM.

[4] Zekeriya Erkin, Michael Beye, Thijs Veugen, and Reginald L. Lagendijk. Efficiently computing private recommendations. In *ICASSP*, pages 5864–5867. IEEE, 2011.

[5] Zekeriya Erkin, Martin Franz, Jorge Guajardo, Stefan Katzenbeisser, Inald Lagendijk, and Tomas Toft. Privacy-preserving face recognition. In *Proceedings of the 9th International Symposium on Privacy Enhancing Technologies*, PETS '09, pages 235–253, Berlin, Heidelberg, 2009. Springer-Verlag.

[6] Zekeriya Erkin, Thijs Veugen, Tomas Toft, and Reginald L. Lagendijk. Generating private recommendations efficiently using homomorphic encryption and data packing. *IEEE Transactions on Information Forensics and Security*, pages 1053–1066, 2012.

[7] T. Ryan Hoens, Marina Blanton, and Nitesh V. Chawla. A private and reliable recommendation system for social networks. In *Proceedings of the 2010 IEEE Second International Conference on Social Computing*, SOCIALCOM '10, pages 816–825, Washington, DC, USA, 2010. IEEE Computer Society.

[8] ArjanJ.P. Jeckmans, Michael Beye, Zekeriya Erkin, Pieter Hartel, ReginaldL. Lagendijk, and Qiang Tang. Privacy in recommender systems. In Naeem Ramzan, Roelof van Zwol,

Jong-Seok Lee, Kai Clver, and Xian-Sheng Hua, editors, *Social Media Retrieval*, Computer Communications and Networks, pages 263–281. Springer London, 2013.

[9] Dmitry Kononchuk, Zekeriya Erkin, Jan CA van der Lubbe, and Reginald L Lagendijk. Privacy-preserving user data oriented services for groups with dynamic participation. In *Computer Security–ESORICS 2013*, pages 418–442. Springer Berlin Heidelberg, 2013.

[10] Neal Lathia, Stephen Hailes, and Licia Capra. Private distributed collaborative filtering using estimated concordance measures. In *Proceedings of the 2007 ACM Conference on Recommender Systems*, RecSys '07, pages 1–8, New York, NY, USA, 2007. ACM.

[11] Frank Linton, Deborah Joy, Hans peter Schaefer, and Andrew Charron. Owl: A recommender system for organization-wide learning, 2000.

[12] Ikmal Hisham Md Tah Mohd. Ikhsan Md. Raus and Saadiah Yahya. Personal information disclosure in facebook: Theawareness of uitm pahang students. International Journal of Future Computer and Communication, 2013.

[13] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *Proceedings of the 2008 IEEE Symposium on Security and Privacy*, SP '08, pages 111–125, Washington, DC, USA, 2008. IEEE Computer Society.

[14] Huseyin Polat and Wenliang Du. Svd-based collaborative filtering with privacy. In *Proceedings of the 2005 ACM Symposium on Applied Computing*, SAC '05, pages 791–795, New York, NY, USA, 2005. ACM.

[15] Reza Shokri, Pedram Pedarsani, George Theodorakopoulos, and Jean-Pierre Hubaux. Preserving Privacy in Collaborative Filtering through Distributed Aggregation of Offline Profiles. In *The 3rd ACM Conference on Recommender Systems (RecSys)*. ACM, 2009.

[16] Thijs Veugen. Encrypted integer division, 2010.