Een privacyvriendelijk aanbevelingssysteem voor mobiele toestellen

Thorwald Frederik Lambrecht

Supervisor(s): Marleen Denert, Luc Martens, Toon De Pessemier

Abstract— **Dit artikel probeert het ideale privacyvriendelijke aanbevelingssysteem te creëren voor een mobiel toestel. In dit proces zal het trachten de grenzen van de wisselwerking tussen privacy, nauwkeurigheid en performantie bij aanbevelingssystemen te verleggen. Om dit te bereiken wordt een methode gebruikt op basis van homomorfische encryptie.**

Keywords— **Privacy, mobiel, aanbevelingssysteem**

I. Inleiding

OM gepersonaliseerde aanbevelingen toe te laten hebben aanbevelingssystemen privacygevoelige data nodig van hun gebruikers. Dit verplicht de *service provider* om deze data bij te houden en opent de poort voor de privacy inbreuken. Deze inbreuken kunnen het gevolg zijn van een oneerlijke provider of een naïeve gebruiker, maar ook van onvoldoende data bescherming tegenover aanvallen.

Er bestaan verschillende benaderingen om de privacy van de gebruiker te verbeteren. Eerst en vooral kan de gebruiker beter ingelicht worden over de grootteorde waarop zijn data wordt bijgehouden en de wijze waarop ze gebruikt wordt. Ten tweede zou de wet rond privacy strikter kunnen gemaakt worden. Een andere mogelijkheid is het gebruik van privacyvriendelijke algoritmes.

Het gebruik van bestaande algoritmes vermindert minstens de nauwkeurigheid van de aanbevelingen of de performantie. Rekening houdend met de mobiele setting, is het ook belangrijk om een werkwijze te vinden die niet veel processortijd of data overdracht nodig heeft op de client. Om uit te zoeken hoe een aanbevelingssysteem in een mobiele setting de wisselwerking best aanpakt, was er onderzoek nodig in de bestaande privacyvriendelijke oplossingen.

II. Onderzoek naar bestaande privacyvriendelijke methodes

Er bestaan verschillende methodes om de privacy te verbeteren in aanbevelingssystemen. Deze methodes omvatten werkwijzes met behulp van anonimisatie, randomisatie, aggregatie van gebruikersprofielen en cryptografische protocollen. Voor elk van deze mogelijkheden werd er een grondige analyse gedaan van minstens één voorbeeldoplossing en werd er een vergelijking gemaakt.

De oplossing gebaseerd op anonimisatie [1] maakt gebruik van agents die anoniem communiceren met elkaar. Ondanks het feit dat gedurende de aanvragen en de vergelijking van gebruikers onderling voor user-user collaborative filtering, de gebruikers anoniem blijven, garandeert anonimiteit geen privacy. Dit is bewezen door Narayan [2].

Het randomisatie algoritme gebruikt door Polat en Du [3] voorziet ook niet in volledige privacy voor de gebruiker, aangezien de server nog steeds de range van waarden weet waartussen een gebruiker zijn aparte waarden liggen. De methode verliest ook sterk aan nauwkeurigheid bij het gebruik van kleine datasets.

Het gebruik van aggregatie van gebruikersprofielen door Shokri et al. in [4] biedt de gebruiker de mogelijkheid om zijn voorkeuren te ontkennen maar toont de server nog steeds originele beoordelingen van de gebruiker. Deze werkwijze verliest boet wel niet aan nauwkeurigheid in.

De oplossing aan de hand van cryptografische protocollen met gebruik van een peer-to-peer relatie tussen clients [7] is wel privacyvriendelijk maar heeft een sociaal netwerk nodig waar gebruikers vaak online zijn. De methode vraagt ook te veel berekeningen aan de clientkant voor een mobiele setting.

De beste methode lijkt deze op basis van cryptografische protocollen met twee servers door Erkin et al [5]. Deze aanpak gebaseerd op een eerdere oplossing [6] biedt een hoog privacyniveau en vereist geen zware berekeningen aan de clientkant. Op basis van deze vaststelling werden de cryptografische protocollen van deze methode gekozen om te dienen als basis voor deze oplossing.

III. De privacyvriendelijke oplossing

We besloten om een native androidapplicatie te maken en de servers werden geschreven in Java. Ze communiceren allemaal onderling met het HTTP-protocol. Als testdatabank werd de MovieLens databank gebruikt met 100.000 beoordelingen, 943 personen en 1682 films. De oplossing in [5] maakt gebruik van twee servers, een aanbevelingsserver en een tweede server die wordt ingezet door een betrouwbare derde partij. De clientapplicatie stuurt de beoordelingen en voorkeuren van de gebruikers naar de aanbevelingsserver, ze zijn geëncrypteerd met de publieke Paillierssleutel van de tweede server. Dit hoeft niet elke keer te gebeuren als een gebruiker een item beoordeelt. Om een voorspelling te doen van een beoordelingswaarde voor een item is er communicatie nodig tussen de twee servers op basis van hun Paillier en DGK cryptosysteem. Zo wordt het aantal en de som van de geëncrypteerde ratings van gebruikers met gelijkaardige voorkeuren geteld zodat de client het gemiddelde kan bepalen. De gelijkaardigheid tussen twee gebruikers wordt bepaald door de gekende Pearson-correlatie, die hier ook deels door de client wordt berekend. Om de berekeningen van de Pearsoncorrelatie aan de serverkant te maken worden ingewikkelde protocollen zoals een vermenigvuldigingsprotocol en een drempelprotocol gebruikt.

Er zijn verschillende beslissingen die moeten genomen worden tijdens het implementeren van deze protocollen, vooral aan de clientkant: In tegenstelling tot [5] zou de gebruiker zijn

geëncrypteerde waarden best niet stuk per stuk opsturen als hij een item beoordeelt. Dit zou aan de service provider laten weten welke items de persoon geratet heeft. Een mogelijkheid is om een aantal willekeurige items te kiezen en de beoordelingen hiervan mee te sturen. Indien deze items niet beoordeeld zijn wordt een nulwaarde geëncrypteerd en meegestuurd. Toch kan hiermee nog privacy lekken indien alle random gekozen items dezelfde eigenschappen hebben. Dit kan worden vermeden door deze items op een slimme manier te kiezen, maar dit laat wel zijn offsets (zie later) en voorkeuren berekend op een oude versie van het gemiddelde. Om optimale privacy in onze applicatie te bereiken stuurt de gebruiker zijn ratings en voorkeuren, samen ook zijn profiel genoemd, in één keer naar de server. Hier berekent hij voor alle items zonder ratings een geëncrypteerde nulwaarde die hij meegeeft met zijn profiel.
Voor deze oplossing is het ook nodig om per rating uit een profiel, een extra geëncrypteerde bit wordt meegegeven. Deze geëncrypteerde bit $q_{U_x,i}$ is n als de gebruiker x item i heeft beoordeeld en nul indien niet. Dit maakt het mogelijk om het aantal effectief ingevulde gebruikte ratings te bepalen in formule (1). De encryptie van de beoordelingen wordt ook best berekend op het moment dat de gebruiker zijn profiel stuurt. Indien dit gebeurt onmiddelijk na een beoordeling en pas later het hele profiel wordt opgestuurd kan, over tijd, de aanbevelingsserver vergelijken welke geëncrypteerde waarden veranderd zijn. In het origineel artikel [5] wordt een voorspellingswaarde van een rating berekend door het gemiddelde te nemen van de beoordelingen van mensen met een gelijkaardigheidsscore boven een bepaalde drempelwaarde. Toch laten de protocollen een verbetering in deze wijze toe door voorspellingswaarden te genereren gebaseerd op een aangepaste versie van een formule die vaak wordt gebruikt in het user-user collaborative filtering:

$$p_{U_1,i} = \bar{r}_{U_1} + \frac{\sum_{j=2}^{n}(r_{(U_j,i)} - \bar{r}_{U_j}).s_{(U_1,U_j)}.q_{(U_j,i)}}{\sum_{j=2}^{n} s_{(U_1,U_j)}.q_{(U_j,i)}} \quad (1)$$

Hier staat $r_{(U_x,i)}$ voor de beoordelingswaarde van user x voor item i, $\bar{r}_{U_x}$ de gemiddelde beoordelingswaarde van een gebruiker over alle items heen en $p_{U_x,i}$ voor de voorspellingswaarde voor gebruiker x, item i. De waarde $s_{(U_1,U_j)}$ staat voor het bit dat is berekend door het drempelprotocol en het is 1 als de gebruiker een gelijkheidswaarde heeft boven de drempelwaarde. In de plaats van de beoordeling zelf zoals in [5] wordt deze keer het verschil van de beoordeelde waarde $r_{(U_j,i)}$ en de gemiddelde waarde $\bar{r}_{U_1}$ van de gebruiker gestuurd. Deze waarde moet geconverteerd worden naar een positieve integer voor het gebruik van de cryptosystemen. Dit kan op een makkelijke manier gebeuren $result = ((r_{(U_j,i)} - \bar{r}_{U_j}) + 5) * 1000$. Eenmaal de som is genomen over alle gelijkaardige gebruikers, kan de client de conversie omkeren door te delen met 1000 en dan 5 af te trekken per gelijkaardige gebruiker.

IV. Resultaten

De voorspellingswaarden van de verbeterde methode (lichte lijn) tonen signifant betere scores op nauwkeurigheid dan de methode uit [5] (de donkere lijn).

De MAE verbeterde van rond 0.82 ster to ongeveer 0.74 ster, dit is zeer dicht bij de MAE 0.7146 van de privacy-

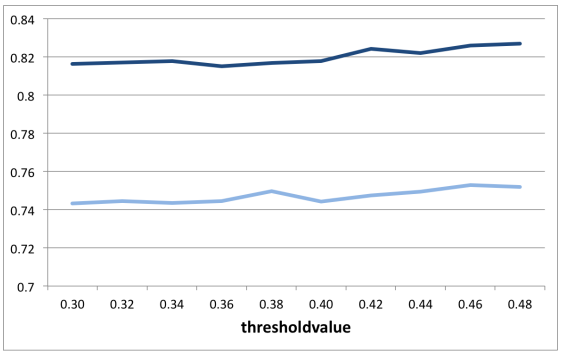

Fig. 1. MAE resultaten over 10.000 predictions berekend op verschillende drempelwaarden.

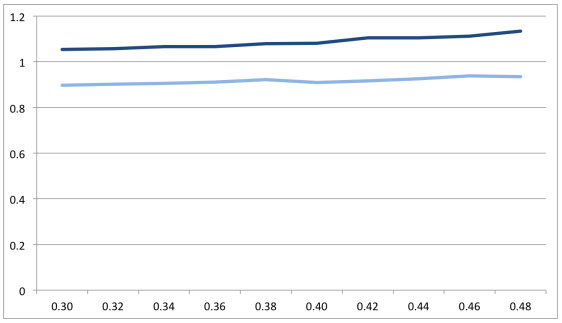

Fig. 2. MSE resultaten over 10.000 predictions berekend op verschillende drempelwaarden.

onvriendelijke oplossing op identiek dezelfde database door [3]. Sinds de privacy nog verbeterd is ten opzichte van de originele methode behaalt het een zeer hoog privacyniveau. Het uploaden van het profiel aan de clientkant is $O(N+S)$ met N als het aantal items en S als het aantal voorkeuren. Voor elk item gebeuren er 2 encrypties en voor elke voorkeur 1 encryptie. Deze encryptie gebeurde op het testtablet [1] in 1,52 seconden gemiddeld over 10 keer. De servers verrichten zware berekeningen, de testserver[2] had 7 minuten 26 seconden nodig om voorspellingswaarden te genereren voor alle 1682 items voor n gebruiker, als er gebruik wordt gemaakt van 30 preferences. Deze berekeningen kunnen echter nog geoptimaliseerd woord met het gebruik van een protocol op een lager niveau, want het was niet de beste keuze om het HTTP-protocol te gebruiken voor de communicatie tussen de twee servers. Ook is er een mogelijkheid om de gebruiker minder waarden naar de server te laten sturen zoals besproken. Dit zou een impact hebben op de nauwkeurigheid van het algoritme maar zou het werk aan de kant van de servers verminderen en de grootte van de database op de aanbevelingsserver beperken.

V. Conclusion

Deze oplossing garandeert een zeer hoog niveau van privacy door het gebruik van Paillier en DGK encryptie en het feit dat de server niet weer welke items beoordeeld zijn door de gebruiker. Hoge nauwkeurigheid wordt ook behaald zonder al te veel

[1] Samsung Galaxy Tab4 (7.0) Wi-Fi on Android 4.4.2
[2] MacBook Pro 4 GB RAM 2.4 GHZ i7 processor

berekeningen aan de clientkant. Deze eigenschappen maken deze methode ideaal voor het gebruik als aanbevelingssysteem voor mobiele toestellen. De servers zelf doen zware berekeningen maar deze kunnen nog verder geoptimaliseerd worden.

REFERENCES

[1] Cissée, Richard and Albayrak, Sahin *An Agent-based Approach for Privacy-preserving Recommender Systems*, Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems,2007

[2] Narayanan, Arvind and Shmatikov, Vitaly *Robust De-anonymization of Large Sparse Datasets*, Proceedings of the 2008 IEEE Symposium on Security and Privacy, 2008

[3] Polat, Huseyin and Du, Wenliang *SVD-based Collaborative Filtering with Privacy*, Proceedings of the 2005 ACM Symposium on Applied Computing,2005

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

[5] Zekeriya Erkin and Thijs Veugen and R.L. Lagendijk *Privacy-Preserving Recommender Systems in Dynamic Environments*, IEEE Workshop on Information Forensics and Security, 2013

[6] Zekeriya Erkin and Michael Beye and Thijs Veugen and Reginald L. Lagendijk *Efficiently computing private recommendations*, ICASSP,2011

[7] Hoens, T. Ryan and Blanton, Marina and Chawla, Nitesh V. *A Private and Reliable Recommendation System for Social Networks*, Proceedings of the 2010 IEEE Second International Conference on Social Computing, 2010