Hide & Share: Landmark-based Similarity for Private KNN Computation

Antoine Boutet, Davide Frey, Rachid Guerraoui, Anne-Marie Kermarrec, Antoine Rault, François Taïani, Jingjing Wang

Hide & Share: Landmark-based Similarity for Private KNN Computation

Antoine Boutet*, Davide Frey†, Rachid Guerraoui‡, Anne-Marie Kermarrec†, Antoine Rault†, François Taïani§ and Jingjing Wang‡

*University of Saint-Etienne, France
Email: antoine.boutet@univ-st-etienne.fr

†INRIA Rennes, France
Email: firstname.lastname@inria.fr

‡EPFL, Switzerland
Email: firstname.lastname@epfl.ch

§University of Rennes 1, France
Email: francois.taiani@irisa.fr

***Abstract*—Computing k-nearest-neighbor graphs constitutes a fundamental operation in a variety of data-mining applications. As a prominent example, user-based collaborative-filtering provides recommendations by identifying the items appreciated by the closest neighbors of a target user. As this kind of applications evolve, they will require KNN algorithms to operate on more and more sensitive data. This has prompted researchers to propose decentralized peer-to-peer KNN solutions that avoid concentrating all information in the hands of one central organization. Unfortunately, such decentralized solutions remain vulnerable to malicious peers that attempt to collect and exploit information on participating users.**

In this paper, we seek to overcome this limitation by proposing *H&S* (Hide & Share), a novel landmark-based similarity mechanism for decentralized KNN computation. Landmarks allow users (and the associated peers) to estimate how close they lay to one another without disclosing their individual profiles.

We evaluate *H&S* in the context of a user-based collaborative-filtering recommender with publicly available traces from existing recommendation systems. We show that although landmark-based similarity does disturb similarity values (to ensure privacy), the quality of the recommendations is not as significantly hampered. We also show that the mere fact of disturbing similarity values turns out to be an asset because it prevents a malicious user from performing a profile reconstruction attack against other users, thus reinforcing users' privacy. Finally, we provide a formal privacy guarantee by computing an upper bound on the amount of information revealed by *H&S* about a user's profile.

Keywords*—*Data privacy, Nearest neighbor searches, Peer-to-peer computing, Recommender systems

I. Introduction

K-Nearest-Neighbor (KNN) algorithms provide a fundamental tool to mine and explore large amounts of data. In particular, they lie at the core of memory-based collaborative filtering (CF), a common technique for providing recommendation to users [1]. The use of KNN for memory-based CF has been particularly fruitful and has led to the recent emergence of peer-to-peer (P2P) recommenders based on highly decentralized KNN algorithms [2], [3].

Peer-to-peer KNN recommenders are particularly scalable, and have therefore been proposed as a way to address the scalability issues that characterize centralized recommenders. Distributing KNN computation across peers makes it possible to compute recommendation without requiring huge servers or data centers.

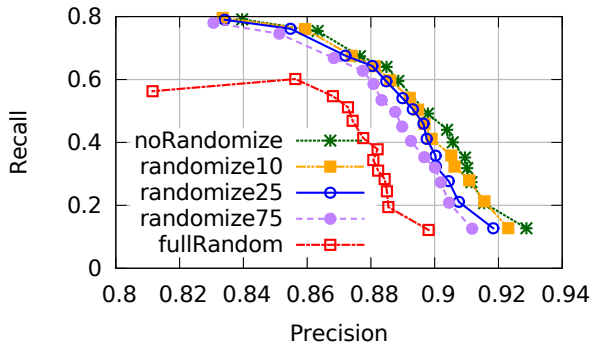

Fig. 1. Recommendation quality with different levels of randomization of user profiles. This quality is not significantly hampered by levels of randomization of up to 75%.

Peer-to-peer KNN recommenders also avoid the danger of prominent players acting as "Big Brothers". Deconcentrating data across peers makes it more difficult for content providers to access and possibly reuse the personal data for purposes other than recommendation. Yet, the peer-to-peer model introduces new privacy threats that do not come from a Big Brother but from the peers themselves. The decentralized KNN algorithms at the basis of most peer-to-peer recommenders [2], [3] require peers to exchange their interest profiles with other peers in order to compute similarity values. In doing so, they do not simply *risk* to share sensitive information; they systematically require users to share personal data with random other users. This makes it very easy for an attacker to learn about the interests of a large number of victims.

To address this challenge, we propose *Hide & Share* (*H&S*), a novel similarity mechanism for P2P KNN computation. *H&S* makes it possible to compute the KNN graph without requiring users to share their profile information with anyone else. *H&S* relies on a simple observation: user-centric KNN applications such as recommendations do not require

perfect knowledge. To illustrate this fact, Figure 1 depicts recommendation quality (quality increases towards the top and the right) with varying level of randomness injected into user profiles. The plot shows that randomness levels of up to 75% do not significantly hamper recommendation quality.

Based on this observation *H&S* trades-off precision in the computation of similarity for privacy. This allows it to gain significant protection in terms of privacy with a minimal impact on applications like recommendation. This makes *H&S* a perfect fit for decentralized CF systems.

H&S's key contribution lies in a novel *landmark-based* approximation technique as well as in a fair landmark-generation protocol. The landmarks of our solution allow two users to indirectly measure their similarity by comparing their own profiles with a set of randomly generated profiles (the landmarks). The similarity between a user's profile and a landmark acts as a coordinate in a coordinate system. Users then exchange vectors of coordinates and compute an approximation of their actual similarity. This preserves user privacy as users do not exchange their full profiles and landmark coordinates only reveal a limited amount of information about a user.

We present and evaluate *H&S* using real data traces. We also demonstrate formally its privacy guarantees by computing an upper bound on the amount of information leaked by *H&S*'s similarity approximation. Our results show that *H&S*'s KNN provides a reasonable trade-off between privacy and utility. *H&S* disturbs similarity values but it does not significantly hamper the quality of the resulting recommendations. Approximate similarity values constitute instead an asset towards privacy preservation as they effectively prevent adversaries from performing profile reconstruction attacks as we show in Section IV.

In the remainder of this paper, we first describe our system model in Section II before detailing our contribution in Section III. Then we evaluate *H&S* experimentally in terms of recommendation quality, privacy protection, and overhead in Section IV, and analyze its privacy guarantees in Section V. Finally, we discuss related work in Section VI and present our conclusions in Section VII.

II. System Model

We present *H&S* in the context of a user-based peer-to-peer recommender. To this end, we start by describing the operation of such a system, and highlighting the corresponding privacy risks. We then present our adversary model in Section II-C.

A. Decentralized User-based Collaborative-Filtering System

We consider a decentralized collaborative-filtering (CF) system similar to that of [3]. Each user controls a single peer which stores her full profile as a list of ratings for the items she has rated. Ratings may consist either of binary values or of discrete values within a range (e.g. 1 to 5). In the following, we consider binary ratings as in most existing decentralized solutions [3], [4], [5], [6], [7], [2], [8].

The system uses *asynchronous rounds* that are executed periodically by each peer. In each round, each peer attempts to select a better set of similar other nodes (its *neighbors*) according to some similarity metric: for example cosine

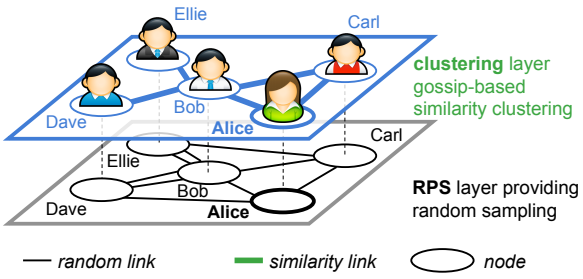

Fig. 2. Gossip-based distributed clustering

similarity [9]. Cosine similarity considers profiles as high-dimensional vectors in which each unique item is a dimension and values for each dimension correspond to ratings. It then evaluates the similarity of two profiles as the cosine of the angle between the two corresponding vectors.

$$cos(u_1, u_2) = \frac{u_1 \cdot u_2}{\|u_1\|\|u_2\|} \quad (1)$$

In what follows, we first describe how the neighbors of a peer are identified in this model (*Neighbor identification*), before moving on to the actual mechanism used for to recommend new items to users (*Recommendation*).

1) Neighbor identification: Peers use two gossip protocols to identify their KNN: a random-peer sampling (RPS) and a clustering protocol. The former maintains a continuously changing topology, while the latter converges to the KNN graph, as illustrated in Figure 2. Both protocols follow the same high-level behavior. In each protocol, each peer maintains a data structure, called *view*, consisting of a list of references to other peers: the peer's current neighbors in the corresponding protocol. Periodically, a peer p contacts another peer q from this list and sends it a subset of its own view—half of its view in the RPS protocol, and its entire view in the clustering protocol. Upon receiving such a subset, q merges the received subset with its own view. In the case of RPS, it keeps e random entries from the union of the two views. In the case of the clustering protocol, it keeps the e entries whose profiles are most similar to its own after combining its own clustering view, its own RPS view and the received clustering view. Then q replies by sending to p a subset of its view before the update, and p updates its view analogously. The clustering protocol provides each peer with a view that converges to its KNN. The

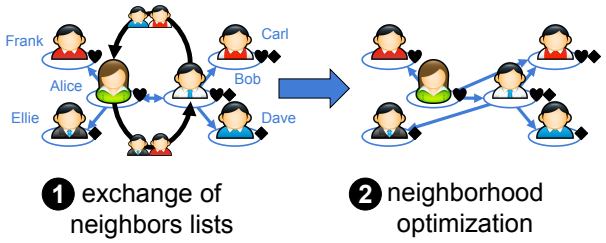

Fig. 3. Clustering mechanism for convergence to an optimal neighborhood. In this example, after exchanging their profiles, Alice and Bob modify their neighbors in order to be connected with the users who share the most their interests.

RPS provides resilience to churn and partitions and ensure that the process cannot get stuck into a local minimum.

Figure 3 exemplifies the operation of the clustering protocol. Alice and Bob are interested in hearts, though Bob prefers diamonds. After exchanging their respective list of neighbors, they keep the users which are closest to their interests. In this example, Alice replaces Ellie with Carl who likes hearts, and Bob replaces Alice with Ellie who likes diamonds. After a few cycles of this protocol, each peer's neighborhood view contains the corresponding KNN.

2) Recommendation: Peers use the KNN identified with the above protocol to recommend items to their users. In typical systems, each peer identifies the items that were found most interesting by its KNN and to which the peer has not yet been exposed. In the case of binary rating, these consist of the items that were *liked* by the largest number of KNN and to which the peer has not been exposed.

B. Privacy Risks

As suggested, the above protocols requires peers to share their profiles with each other in order to identify their KNN. This constitutes a major privacy risk: before convergence, both the RPS and the clustering protocol require peers to communicate with a large number of other peers, even with non similar ones. This means that a malicious non-similar peer can easily copy the profile of a target peer in order to forcibly enter its clustering view. In the rest of this paper, we remove this privacy threat by introducing *H&S*, a novel similarity mechanism that does not require peers to exchange their profile information.

Thanks to *H&S*, peers can identify their KNN without having to disclose any personal details to other peers. Once they identified their KNN, they do share their profile information with neighbors that are sufficiently stable to compute recommendations as described in Section II-A2. However, this does not constitute a significant privacy risk because peers identified as KNN already know that they have similar profiles. Learning the details of each other's profiles therefore does not add much to this knowledge. Conversely, a malicious peer that wanted to become a neighbor of a target node would not be able to clone the corresponding profile without being already similar to the target peer.

C. Adversary Model

In the rest of this paper, we consider a *curious adversary* model. Our adversary can only take a limited set of active actions to reach her goal, and can otherwise passively gather information. The goal of the adversary is to discover the profile of a chosen user (target) by a profile reconstruction attack, using information obtained during similarity computation. The adversary only controls one peer, i.e we assume there is no collusion between adversaries, and our adversary cannot forge peer identities (no sybil capacity). She also has no *a priori* knowledge regarding her target's interests. The active actions the adversary can take are: tap unencrypted communications; attempt to bias multi-party computations; compute her similarity with her target as many times as she want.

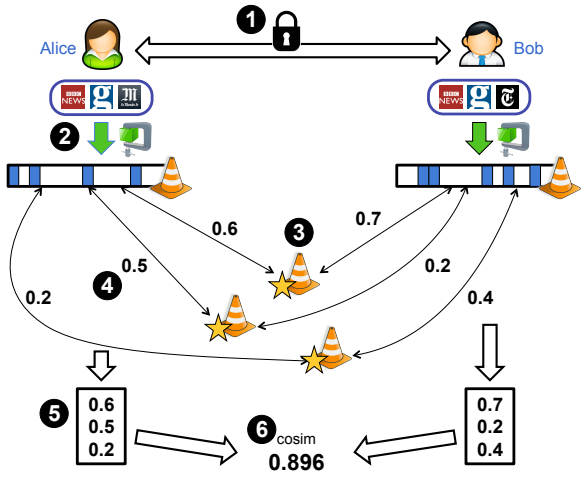

Fig. 4. Overview of the *H&S* similarity computation mechanism

III. The Hide & Share Landmark-based Similarity

We address the privacy issues in decentralized KNN computation by introducing *H&S* (Hide & Share), a novel mechanism for similarity computation. *H&S* relies on a simple observation: good recommendations do not require perfect neighborhoods. *H&S* therefore relaxes the precision of similarity computation, by exploiting randomly selected intermediate profiles (*landmarks*) with respect to which each peer positions itself. This allows peers to compute similarity scores they can exploit without exchanging clear-text profiles.

H&S landmarks take inspiration from reference points in geo-localization systems. For instance, two-dimensional geographic locations usually refer to *the Equator* and *the Greenwich meridian*: two landmarks that define their latitude and longitude. However, our landmarks also exhibit two important differences with respect to this geographic analogy.

First our landmarks are not fixed and set for the whole system; rather, each pair of peers randomly generates its own set of landmarks. This prevents cross-pair comparisons. Second, we use far fewer landmarks than there are dimensions in our system. This prevents a precise reverse computation of each peer's clear-text coordinates (i.e. its profile) from its landmark coordinates. Thanks to these differences, users can safely exchange their landmarks because they do not characterize their interests in any specific topic.

Figure 4 presents an overview of the operation of *H&S* by means of an example. Alice and Bob need to compute their similarity with each other. In a traditional system like the one described in Section II, Bob would send his profile to Alice and Alice would send hers to Bob. Each of them would then compute the similarity by applying Equation (1). With *H&S*, none of this happens. Rather, Alice and Bob follow these 6 steps. (1) They create a secure communication channel. (2) They each derive a compact version (Bloom filter) of his/her profile. (3) They agree on a set of random landmarks. (4) They each compute the similarity of his/her compact profile with each landmark. (5) They each gather these similarity values in a similarity vector. (6) They exchange each other's similarity vector and compute their final similarity

estimate. From a practical perspective, this translates into to two main components: a landmark generation mechanism, and a similarity estimation protocol. In the following we detail each of these two contributions.

A. Landmark Generation

H&S uses landmarks to estimate the similarity between two peers without requiring them to exchange their profiles with each other. To prevent adversaries from reconstructing profile information from these landmarks, the landmark generation mechanism must satisfy a set of requirements.

i *Computation confidentiality*: Only the two peers participating in the similarity computation may access the data they exchange. This includes landmark and similarity values.
ii *Independence of peer profiles*: Landmarks must be random and independent of the profiles of the peers that generate them.
iii *Fair landmark generation*: The choice of the landmarks must be fair. Neither of the two participating peers may bias the generated landmarks.
iv *Minimal information release*: An attacker should not be able to reconstruct a target profile by combining information from multiple landmark similarities, or by repeatedly computing its *H&S* similarity with the target.

In the following, we present our landmark generation mechanism by focusing on how it addresses each of these requirements. We detail the various steps in lines 1 through 18 of Algorithm 1.

1) Computation Confidentiality: Requirement (i) states that third-party peers should not be able to eavesdrop any communication between peers that are computing their similarity. To achieve this, *H&S* encrypts all the communication between two peers, including that relative to landmark generation.

Specifically, each peer maintains a public/private key pair. Peers exchange their public keys with each other by attaching them to the information transferred through the RPS and clustering protocols, similar to what was is done in [6]. In addition, we assume that peers may verify the authenticity of a public key by means of a certification authority or a web of trust [10], [7].

Peers use their key pairs to establish a secure communication channel whenever they need to evaluate their similarity. To this end, they exploit an authenticated key agreement (AK) protocol [11] as shown in lines 1 and 2. A possible AK protocol consists of an authenticated variation of the elliptic curve Diffie-Hellman key agreement such as the one available in the NaCl cryptographic library [12].

2) Independence of peer profiles: Requirement (ii) states that landmarks consist of *randomly generated* profiles that are independent of the profiles or of the choices of participating peers. However, as we discussed in Section II, profiles consist of lists of item-score pairs, where the items belong to an unbounded or at least very large universe. This would make it difficult, if not impossible to generate random landmarks. To circumvent this problem, *H&S* replaces traditional profiles with *compact profiles* (step 2 in Figure 4).

A *compact profile* consists of a Bloom filter [13] and contains only the items considered as liked by the corresponding peer. A Bloom filter provides a compact representation of a set in the form of an array of n bits. To add an item to the set, the bloom filter applies h hash functions to the item to obtain h bit positions in the array and sets these positions to 1. To query for the presence of an item, the filter uses the same hash functions and checks if all the bits at the h indexes have a value of 1.

Compact profiles carry slightly less information than full profiles. First, Bloom filters can return false positives even though they never return false negatives. Second, compact profiles cannot distinguish between disliked items and items to which the user has not been exposed. This does not constitute a problem: the like status of items proves sufficient to describe the interests of peers, and the effect of false positives may actually be beneficial in terms of privacy. Compact profiles also reduce Equation (1) to counting the number of common bits between the two bloom filters.

Given a user or peer, $p \in \{1, 2, \ldots, N\}$, we denote her compact profile as $\vec{c_p} \in \mathbb{Z}_2^n$. Lines 10 through 18 of Algorithm 1 show how peers use compact profiles to generate random landmarks. Let L be a system parameter specifying the number of landmarks to generate and let PRNG be a pseudo-random number generator whose code is available to all peers (for example MRG32k3a [14] or Mersenne Twister [15]). Two peers, say $p1$ and $p2$, may generate a set of landmarks by first generating a common random seed (lines 10 to 13 in Algorithm 1). Then, each of them saves this seed (line 14), along with a timestamp, and uses it to initialize the PRNG (line 15). Finally Each of the two peers independently uses the PRNG to generate the L landmarks: $\{M_i\}$ with $i \in \{0, 1, \ldots, L\}$ (lines 16-18). Each generated landmark consists of a vector of bits of the same size as a compact profile, with a few random bits (around 5%) set to 1, while other bits are set to 0. This proportion of set bits mimics that of compact profiles, which are usually sparse.

3) Fair Landmark generation: Requirement (iii) states that the choice of the landmarks must be fair. To achieve this, peers agree on their common seed using a bit-commitment scheme like Blum's coin-flipping protocol [16]. Blum's protocol operates as follows. Both $p1$ and $p2$ flip a coin. They set the output of the protocol to 1 if they obtain the same result, and to 0 otherwise. To exchange their coin-flip results without cheating, $p1$ and $p2$ employ a bit-commitment scheme. After flipping its coin, $p1$ sends $p2$ a commitment on its result ($f(concatenate(result, nonce))$). Then $p2$ reveals its result to $p1$, and $p1$ reveals its result as well as the nonce it used for the commitment to $p2$. $p2$ cannot cheat because it is the first to send its result. $p1$ cannot cheat because $p2$ can then check its result against the initial commitment.

Blum's protocol does not provide an unbiased coin, which is impossible in the two-party case [17], but a weaker fairness guarantee that suffices for our application. This guarantee holds as long as a malicious party does not abort the protocol before it ends. Since the two peers in our protocol use a secure channel, if $p2$ aborts, $p1$ can deduce that $p2$ is trying to bias the result.

4) Minimal information release: Requirement (iv) states that attackers should not be able to reconstruct a target profile by combining information from multiple landmarks or by

Algorithm 1 *H&S* landmark-based similarity computation protocol between peers $p1$ and $p2$, as executed by $p1$

1: $session\_key \leftarrow AK(key_{p1}, pub\_key_{p2})$
2: $secure\_channel \leftarrow connect(p2, session\_key)$
3: **if** $p2$ is known **then**
4: $s \leftarrow load\_seed(p2)$
5: **if** s is not older than th_L **then**
6: $seed \leftarrow s$
7: **goto** 15
8: **end if**
9: **end if**
10: **for all** i s.t. $0 \leq i < 32$ **do**
11: $r \leftarrow rand\_bit()$
12: $seed[i] \leftarrow coin\_flip(r, secure\_channel)$
13: **end for**
14: $save\_seed(p2, seed, timestamp(now))$
15: $prng \leftarrow init\_prng(seed)$
16: **for all** i s.t. $0 \leq i < L$ **do**
17: $\vec{M}_i \leftarrow generate\_lm(prng)$
18: **end for**
19: **for all** i in $0 \leq i < L$ **do**
20: $\sigma_{p1}[i] \leftarrow cosine(\vec{c}_{p1}, \vec{M}_i)$
21: **end for**
22: $send(\vec{\sigma}_{p1}, secure\_channel)$
23: $\vec{\sigma}_{p2} \leftarrow receive(secure\_channel)$
24: $similarity \leftarrow cosine(\vec{\sigma}_{p1}, \vec{\sigma}_{p2})$
25: **return** $similarity$

repeatedly computing their similarity with the target. To satisfy the first part of this requirement, *H&S* similarity uses a small number of landmarks with respect to what would be required to reconstruct the original profile. In Section IV-D, we show that this does not significantly impact the ability to provide good recommendations.

To satisfy the second part of this requirement, *H&S* peers do not generate new landmarks each time they meet. Rather they only do so if their latest common set of landmarks is older than a threshold, th_L. To achieve this, they verify the timestamp associated with their latest saved common seed. If the timestamp is newer than the threshold, then they reuse the seed, otherwise they generate a new random seed.

B. Similarity approximation

We conclude the description of our protocol by presenting how *H&S* approximates the similarity between two peers using its randomly generated landmarks. Let $\{M_1, \ldots, M_L\}$ be a set of common landmarks known to peers $p1$ and $p2$. First, each of the two peers independently computes its similarity with each of these landmarks (step 4 in Figure 4 and lines 19-21 in Algorithm 1). This consists in applying Equation (1) to its own profile and each of the landmarks. Both $p1$ and $p2$ then store the results of these computations in a similarity vector (respectively $\vec{\sigma}_{p1}$ and $\vec{\sigma}_{p2}$) as shown in step 5 in Figure 4 and on line 20 in Algorithm 1. Second, $p1$ and $p2$ exchange their similarity vectors with each other. This consists of lines 22 and 23 in Algorithm 1. Finally (step 6 and line 24), $p1$ and $p2$ compute their *H&S* similarity by applying Equation (1) to their own similarity vector and to the one they have received (note that $cos(\vec{A}, \vec{B}) = cos(\vec{B}, \vec{A})$).

TABLE I. CHARACTERISTICS OF THE TRACES IN TERMS OF NUMBER OF USERS, NUMBER OF ITEMS, NUMBER OF RATINGS AND RATING RANGE.

	# users	# items	# ratings	Rating range
ML-100k	943	1,682	100,000	$[1:5]$ (integers)
ML-1M	6,040	3,900	1,000,000	$[1:5]$ (integers)
Jester-1-1	24,983	100	1,810,455	$[-10:10]$ (continuous)

IV. EVALUATION

We evaluate *H&S* by applying it in the context of a gossip-based decentralized recommendation system. Using publicly available traces, we evaluate the quality of its recommendations, its ability to protect privacy, and the overhead it implies.

A. Methodology

1) Simulator: We use our own simulator written in Java. The simulator takes as input a trace from a recommendation system, consisting of user-item matrix of ratings, split into a training set and a test set. The training set (80% of the ratings) allows peer neighborhoods to converge, while the test set (the remaining 20%) provides the ground truth to evaluate the relevance of recommendations. The simulator operates in two steps. First it uses the training set to simulate the convergence of the clustered overlay, then it generates r recommendations for each peer using the converged overlay and compares the results with the ratings in the test set.

2) Datasets: Table I outlines the characteristics of the three traces we use. ML-100k[1] and ML-1M[1] are traces from the MovieLens [18] online movie-recommendation service. They contain 100,000 and 1,000,000 ratings respectively. Jester-1-1[2] is a trace for the Jester [19] online joke-recommendation service. It is the first third of Jester's dataset-1.

3) Evaluation metrics: We evaluate recommendation quality in terms of precision and recall. The former evaluates whether peers like the recommendations they receive. The latter evaluates if recommendations cover all the interests expressed by the ground truth in the test set.

To evaluate *H&S*'s ability to protect privacy we consider both neighborhood quality, and a privacy metric. Neighborhood quality evaluates how much the neighborhoods provided by *H&S* resemble the optimal neighborhoods, that is those obtained with the standard cosine similarity metric. Specifically, for each user we measure the average of the cosine similarities with all the peers in its *H&S* view, and we normalize it by the average cosine similarity with the peers in the optimal neighborhood obtained using an exhaustive search procedure. Let u be a user with full profile, $profile_u$, and let n_u and N_u be respectively u's *H&S* neighborhood and u's optimal neighborhood. Then we compute u's neighborhood quality as follows.

$$quality(u) = \frac{\frac{1}{k}\sum_{p \in n_u} cos(profile_u, profile_p)}{\frac{1}{k}\sum_{p \in N_u} cos(profile_u, profile_p)}$$

Neighborhood quality provides a first indication of privacy: lower quality implying better privacy. To obtain a more precise

[1]MovieLens datasets are available at: http://grouplens.org/datasets/movielens/

[2]Jester datasets are available at: http://eigentaste.berkeley.edu/dataset/

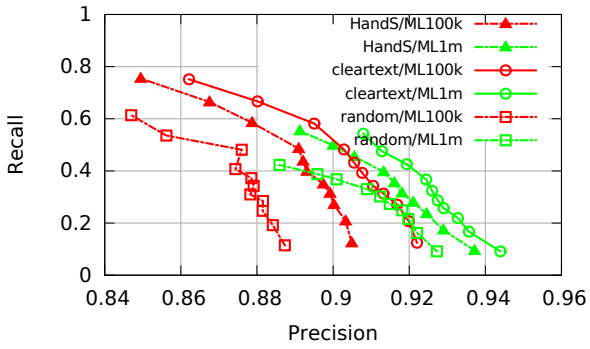

Fig. 5. Recommendation quality expressed as precision and recall with a varying number of recommendations r, using the MovieLens datasets.

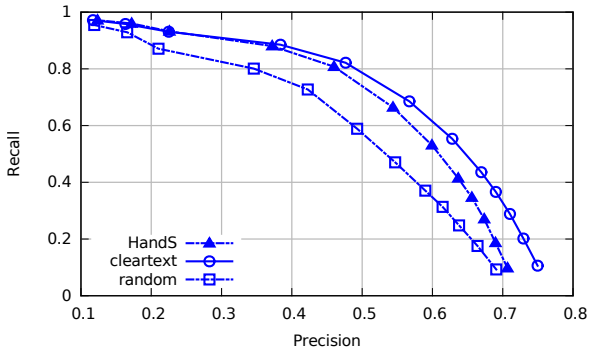

Fig. 6. Recommendation quality expressed as precision and recall with a varying number of recommendations r, using the Jester dataset.

privacy evaluation, we also define *set score*. This metric measures the success rate of the adversary in the context of a profile reconstruction attack. Let G be the set of items that the adversary guesses as liked by the target, and let P be the set of items actually liked by the target. We then define set score as follows, with $\triangle$ as the symmetric difference of two sets.

$$setScore(G, P) = \frac{|G \triangle P| - |G \cap P|}{|G \cup P|}$$

A set score of 1 (adversary's failure) indicates that all the guessed items are wrong (highest privacy), while a set score of -1 (adversary's success) indicates the adversary guessed exactly the target's liked items (no privacy).

Finally, we evaluate overhead by comparing the bandwidth consumption and the storage space required by a *H&S*-based recommendation system with those required by a standard implementation like that of the reference model described in Section II.

4) Default parameters: The subsequent results correspond to simulations using neighborhood and RPS view sizes of 10 peers. Compact profile sizes depend on the dataset used: 660 and 1473 bits for ML-100k and ML-1M respectively (roughly 40% of the number of items), and 99 bits for Jester. When the number of landmarks is not explicitly mentioned, *H&S* uses 50 landmarks. This represents a good trade-off between recommendation quality and privacy. For all the metrics except set score, we plot values averaged over all the peers.

B. Recommendation quality

We evaluate the quality of recommendations providing by an *H&S*-based system using precision and recall [9].

$$precision(user) = \frac{\|recommendedItems \cap likedItems\|}{\|recommendedItems\|}$$

$$recall(user) = \frac{\|recommendedItems \cap likedItems\|}{\|likedItems\|}$$

We consider an item as liked when its rating is greater than or equal to a dataset-dependent threshold ($rating \geq 3$ for MovieLens and $rating \geq 0.0$ for Jester). Using user-dependent threshold values such as the average rating, the median rating, or the half of the rating range for each peer results in similar or lower precision/recall values. This suggests that users tend to use the available rating range similarly.

Peers recommend the r most liked item in their neighborhoods, not including those they have already rated. We check whether a recommended item is liked by looking at the rating given to this item by the recipient in the test set.

Figures 5 and 6 show precision and recall values for several values of r. The former shows the results with the MovieLens datasets, and the latter shows the results with the Jester dataset. For each dataset, we compare the results of the *H&S*-based system (triangle-shaped) with a lower bound (square-shaped) and a cleartext baseline (circle-shaped). The lower bound consists of a CF system that uses completely random neighbors. The baseline consists of the reference model with full profiles in cleartext, as described in Section II. The absolute values of recall and precision are quite high even with random neighborhoods because we do not consider items for which a user has no rating in the original dataset as potential recommendations. More generally, absolute values of precision and recall depend on the predictability and regularity of the dataset, and their acceptable levels depend on the application.

Figure 5 shows consistent results by the *H&S*-based system across the two MovieLens datasets. *H&S* provides a reasonable quality of recommendations: it never suffers from a degradation of more than 50% with respect to the cleartext baseline. Moreover the higher the value of r, the closer the quality remains to that of the cleartext baseline.

Figure 6 shows a similar behavior of the *H&S*-based system with the Jester datastet. Recall reaches almost a value of 1 because the dataset only contains 100 items. This characteristic is also the cause of the maximum precision values being lower than those of the MovieLens datasets. As the test set does not contain many items, we consider that a recommended item without rating in this set is disliked by the recipient, instead of ignoring it as done otherwise. Although this approach is pessimistic, it allows us to make a sufficient number of recommendations.

We showed that *H&S* preserves the quality of recommendation, being only slightly worse than the cleartext baseline. In the following, we show that it achieves this while protecting the privacy of users.

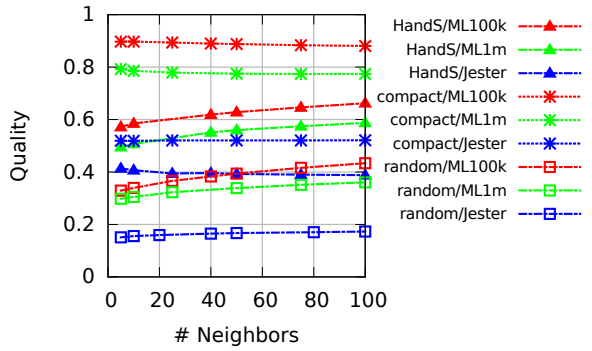

Fig. 7. Effect of compact profiles and the *H&S* similarity on neighborhood quality. The *H&S* similarity is the main source of perturbation of neighborhood quality.

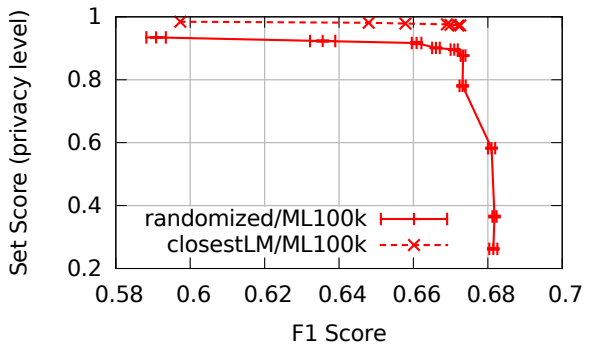

Fig. 8. Trade-off between recommendation quality and privacy for the *H&S*-based system and a system with perturbation-based privacy.

C. Neighborhood quality

In order to evaluate the extent to which neighborhoods are different from the optimal neighborhoods, we use the neighborhood quality measure as defined in Section IV-A3.

Figure 7 shows the evolution of neighborhood quality with the size of neighborhoods. For each dataset, it compares the *H&S*-based system (triangle-shaped) with a CF system using random neighbors as a lower bound (square-shaped) and a variant of our system model using compact profiles (star-shaped). Our reference model from Section II by definition achieves a neighborhood quality of 1 and compact profiles provide neighborhoods that are almost identical in the ML datasets. In the case of Jester, they lower neighborhood quality by 50% because the Jester dataset contains only a few items. This makes it more sensitive to the collisions in the Bloom filters.

H&S similarity has a more significant impact on neighborhood quality than compact profiles. Yet, *H&S*'s neighborhood still retain their utility in terms of recommendation as we showed in Section IV-B. Because landmarks are randomly generated, some of them might be "far" from the two users comparing themselves, thus giving little information about the users' similarity. Moreover, a set of landmarks is not necessarily linearly independent. The lower quality of *H&S*-generated neighborhoods is in fact an asset in terms of privacy. Because of this mix of neighbors with various levels of similarity, the adversary cannot infer her target's interests just by looking at her target's neighbors.

D. Privacy

We evaluate the privacy offered by *H&S* by running a profile reconstruction attack against it. This attack consists in trying to discover the liked items in a targeted peer's profile using information obtained during similarity computation. We quantify the resilience of *H&S* to such attacks with the set score defined in Section IV-A3.

The adversary makes her guess in two steps: *(1)* she tries to infer her target's compact profile, then *(2)* she tries to deduce the items forming this profile. We consider for *(1)* that the adversary uses the closest landmark to her target as her guessed profile. For *(2)*, we consider that the adversary knows all the items in the system, so she includes in her guessed set all the items matching the guessed profile.

We compare our *H&S*-based system with a perturbation-based privacy technique. When using this technique, peers compute their similarity by the usual profile exchange, but they add random noise to their profile to protect their privacy. For the sake of comparison, peers implement this technique by using compact profiles and randomizing a certain percentage of bits in the profile.

Figure 8 compares our *H&S*-based system and a recommendation system using randomized compact profiles, in terms of the trade-off between recommendation quality and privacy. We use set score for the latter and F1 score, the harmonic mean of precision and recall, for the former. We obtain different values of this trade-off by varying the number of landmarks from 2 to 100 for the *H&S* system, and by varying the number of randomized bits in profiles from 5% to 100% for the perturbation-based system. Set score values are averages over 100 different adversaries and 200 different targets, i.e. 20,000 different sets of landmarks. F1 score values correspond to $r = 30$ recommendations.

We observe that the *H&S*-based system provides an excellent level of privacy in any case. It also provides a recommendation quality on par with the best values of the other system, starting from 25 landmarks. However, the increase in recommendation quality does not grow as fast as increase in the number of landmarks.

The recommendation system using randomized compact profiles preserves an almost optimal recommendation quality with up to 75% of randomized bits. Although it achieves reasonable privacy ($setScore = 0.8$ approximately) starting from 50% of randomized bits, it never reaches the privacy levels offered by the *H&S*-based system. Even 100% of randomized bits does not yield a set score of 1 because the attacker tries to match all item signatures against the randomized profile. In general, a fully randomized compact profile will contain more bits with value 1 than a landmark. This will cause the attacker to identify more potentially matching items.

With these basic strategies for the profile reconstruction attack, we showed that *H&S* provides improved privacy to users without sacrificing recommendation quality, and without obvious flaws.

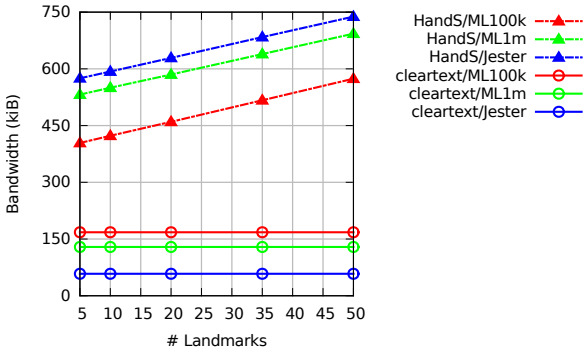

Fig. 9. Average bandwidth consumption of a peer per gossip cycle. The *H&S*-based system consumes roughly twice to seven times more bandwidth than our system model with 5 to 50 landmarks.

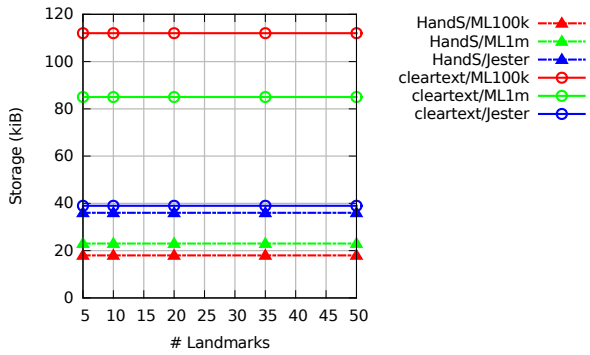

Fig. 10. Average storage space needed for a peer. The *H&S*-based system needs less storage space because peers only store the seed used to generate landmarks.

E. Overhead

We evaluate the overhead caused by *H&S* to peers in terms of bandwidth consumption and storage space.

Overall, *H&S* incurs the most part of its overhead when two peers compute their similarity for the first time because they have to generate a seed using the bit commitment scheme and store this seed. So we measure in our simulations the average number of similarity computations with new peers. The other parameters influencing *H&S*'s overhead are the sizes of the RPS and neighborhood views, and the number of landmarks.

The main factors impacting bandwidth consumption are the exchange of coordinate vectors and the bit commitment scheme. The main factor impacting storage space is the need to store seeds.

Figure 9 compares the *H&S*-based system (triangle-shaped) and the reference model (circle-shaped) in terms of the average bandwidth consumption of a peer per gossip cycle. Bandwidth consumption of the *H&S*-based system increases linearly with the number of landmarks used. It consumes roughly twice to seven times more bandwidth than the reference model, but the absolute values remain reasonable (up to 700KiB per cycle). Moreover, it can probably be improved as a bit commitment protocol with $O(1)$ bits of communication per committed bit exists [20].

Figure 10 compares the *H&S*-based system and the reference model in terms of the average storage space needed by a peer. The *H&S*-based system needs less storage space because peers only store the seed used to generate landmarks instead of storing the profiles of peers in their neighborhood and RPS views as done by standard systems. Still, we observe that the required storage space is tiny compared to the storage capacity of modern devices (computers, smartphones, tablets, *etc*).

1) Computational overhead: We observe that the computational overhead of *H&S* is negligible from the point of view of peers. The most computationally intensive elements of the *H&S* similarity are *(1)* the authenticated key agreement (AK) protocol, *(2)* the generation of random bits (bit commitment scheme and mostly landmark generation), *(3)* the cosine similarity computations with landmarks.

(1) Executing cryptographic primitives incurs negligible cost on modern devices. It is similar to accessing a website with HTTPS: the end user does not perceive a difference between accesses over HTTP and HTTPS. *(2)* Efficient PRNGs such as Mersenne Twister can generate millions of bit/second on modern devices and *H&S* only needs a few thousands of bits to generate landmarks during a gossip cycle, which lasts several seconds at least. *(3)* Cosine similarity is cheap to compute on binary vectors such as landmarks and compact profiles. This confirms the applicability of our approach.

V. Privacy Guarantee

We now analyze *H&S* from an information theoretical viewpoint and compute an upper bound on the amount of information leaked during our landmark-based similarity computation. We carry out our analysis from the point of view of an attacking peer, a, that seeks to obtain information about another peer, p.

During the protocol, a, and p share three pieces of information: the common seed they agree upon, the landmarks, $\{M_1, \ldots, M_L\}$, they generate using this seed, and the similarity vectors $\vec{\sigma}$ containing their similarity with respect to $\{M_1, \ldots, M_L\}$. The first two of these items do not depend on the profile of p and thus do not disclose any information. So we concentrate our analysis on the information that $\vec{\sigma}_p$ may leak about the corresponding compact profiles.

A. Conditional Entropy as a Measure of Information Leakage

We start our analysis by obtaining a first expression for the amount of information leaked by our landmark-based similarity computation. From the attacker's perspective, we define C as the random variable for p's compact profile, with realization $\vec{c}$. Let $\vec{\sigma}$ be the vector of similarity values between $\vec{c}$ and each of the landmarks in the landmark matrix. According to the definition of cosine similarity, we have $\sigma_i = \cos(\vec{c}, M_i) = \frac{\vec{c}M_i}{||\vec{c}|| \cdot ||M_i||} \forall i \in \{1, ..L\}$.

Let us now define an adjusted similarity vector $\vec{v} = \{v_1, ..., v_n\}$, such that $v_i = \sigma_i \cdot ||M_i||$. Then, $v_i = \frac{\vec{c}}{||\vec{c}||} \cdot M_i$. The goal of an attacker is to guess $\vec{c}$ based on the knowledge of M and $\vec{\sigma}$. But knowledge of M and $\vec{\sigma}$ implies knowledge of $\vec{v}$, while knowledge of $\frac{\vec{c}}{||\vec{c}||}$ implies knowledge of $\vec{c}$ because $\vec{c}$ is a binary vector. We can therefore analyze the case of an attacker that tries to guess $\frac{\vec{c}}{||\vec{c}||}$ based on $\vec{v}$ and M.

To this end, we define W as the random variable for p's normalized compact profile, with realization $\vec{w} = \frac{\vec{c}}{||\vec{c}||}$. We also define V as the random variable for the corresponding adjusted similarity vector with realization $\vec{v}$, and Mt as the random variable for the landmark matrix with realization M.

We can then express the uncertainty about W given V and Mt through the conditional entropy $H(W|V, Mt)$. Such uncertainty corresponds to the amount of information protected from the adversary. According to the definition of conditional entropy, we have:

$$H(W|V, Mt) = \sum_{\vec{w},\vec{v},M} p(\vec{w}, \vec{v}, M) \log \frac{p(\vec{v}, M)}{p(\vec{w}, \vec{v}, M)}. \quad (2)$$

We can then express $p(\vec{w}, \vec{v}, M)$ as follows.

$$\begin{aligned} p(\vec{w}, \vec{v}, M) &= p(\vec{v}|\vec{w}, M)p(\vec{w}, M) \\ &= \begin{cases} 1 \cdot p(\vec{w}, M) & \text{if } \vec{v} = \vec{w}M \\ 0 & \text{if } \vec{v} \neq \vec{w}M \end{cases} \end{aligned} \quad (3)$$

This allows us to rewrite Equation (2).

$$\begin{aligned} H(W|V, Mt) &= \sum_{\vec{w},\vec{v},M,\ \text{s.t. } p(\vec{w},\vec{v},M)\neq 0} p(\vec{w})p(M) \log \frac{p(\vec{v}, M)}{p(\vec{w})p(M)} \\ &= \sum_{\vec{w},\vec{v},M,\ \text{s.t. } p(\vec{w},\vec{v},M)\neq 0} p(\vec{w})p(M) \log \frac{p(\vec{v}|M)}{p(\vec{w})} \\ &= \sum_M p(M) \sum_{\vec{w}} p(\vec{w}) \sum_{\vec{v}=\vec{w}M} \log \frac{p(\vec{v}|M)}{p(\vec{w})}. \end{aligned} \quad (4)$$

We can split Equation (4) into two parts, using the fact that $log(\frac{a}{b}) = log(a) - log(b)$. Let $H(W|V, Mt) = J + K$, we have

$$J = \sum_M p(M) \sum_{\vec{w}} p(\vec{w}) \sum_{\vec{v}=\vec{w}M} \log p(\vec{v}|M) \quad (5)$$

$$K = \sum_M p(M) \sum_{\vec{w}} p(\vec{w}) \sum_{\vec{v}=\vec{w}M} -\log p(\vec{w}) \quad (6)$$

Because there is only one $\vec{v}$ such that $\vec{v} = \vec{w}M$, we can write K as

$$\begin{aligned} K &= \sum_M p(M) \cdot \sum_{\vec{w}} p(\vec{w})(-\log p(\vec{w})) \\ &= -\sum_{\vec{w}} p(\vec{w}) \log p(\vec{w}) \\ &= H(W). \end{aligned} \quad (7)$$

So we have $H(W|V, Mt) = H(W) + J$. The quantity $\mathcal{L} = -J$ represents the amount of leaked information, that is the amount of information that the adversary, a, can learn about p's compact profile. Equation (5) provides a first expression for this amount of information. In the following, we refine this expression and present a way to compute an upper bound for it.

B. Leaked Information and the Landmark Matrix

We now identify a relationship between the amount of leaked information and the number of non-zero rows in the landmark matrix, M. We start by taking a closer look at the term $p(\vec{v}|M)$ from Equation (5). We expand it as follows.

$$\begin{aligned} p(\vec{v}|M) &= \sum_{\vec{w}\ \text{s.t. } p(\vec{v},\vec{w}|M)\neq 0} p(\vec{v}, \vec{w}|M) \\ &= \sum_{\vec{w}\ \text{s.t. } p(\vec{v},\vec{w}|M)\neq 0} p(\vec{v}|\vec{w}, M)p(\vec{w}) \\ &= \sum_{\vec{w}\ \text{s.t. } p(\vec{v},\vec{w}|M)\neq 0} p(\vec{w}) \\ &= \sum_{\vec{w}\ \text{s.t. } \vec{v}=\vec{w}M} p(\vec{w}). \end{aligned} \quad (8)$$

The first line follows from the law of total probability while the third and fourth result from the same observations on $p(\vec{v}|\vec{w}, M)$ as in Equation (3).

To solve the final sum in Equation (8), we define $S(\vec{v}, M) = \{\vec{c}|\vec{v} = \frac{\vec{c}}{||\vec{c}||}M, \vec{c} \in \mathbb{Z}_2^n\}$ as the set of all compact profiles that have the same adjusted similarity vectors given a set of landmarks. To evaluate the cardinality of $S(\vec{v}, M)$, we observe that $\forall \vec{c} \in S(\vec{v}, M)$, $||\vec{c}||$ belongs to one of the values $0, \sqrt{1}, \ldots, \sqrt{n}$. The worst case w.r.t. information leakage occurs when all vectors in $S(\vec{v}, M)$ have the same norm $\sqrt{wt}$, wt being the hamming weight of one such vector. Obviously, $\vec{v} \times \sqrt{wt}$ produces an integer vector. Moreover, $\vec{v} \times \sqrt{wt}$ must be a sum of some of the non-zero rows of M, or in other words, a linear combination of the non-zero rows of M. Then an even worse case occurs when not only all vectors have one same norm, but also only one such linear combination exists: in this case, $S(\vec{v}, M)$ is smallest.

Let $k(\vec{v}, M)$ be the number of 1's in the coefficients of such a unique linear combination, and let $j(\vec{v}, M) = wt - k(\vec{v}, M)$ be the number of remaining 1's, those that correspond to zero rows of M. We can then compute the size of $S(\vec{v}, M)$ as $\binom{n-D(M)}{j(\vec{v},M)}$ where $D(M)$ is the number of non-zero rows of M. In the general case, we will therefore have the following lower bound on $|S(\vec{v}, M)|$.

$$|S(\vec{v}, M)| \geq \binom{n - D(M)}{j(\vec{w}, M)} \quad (9)$$

where with a slight abuse of notation we write $j(\vec{w}, M)$ to mean $j(\vec{w}M, M)$. Then, because we assume that all compact profiles are equally likely ($p(\vec{w}) = \frac{1}{2^n}$), we can simplify Equation (8) into Inequality (10).

$$p(\vec{v}|M) = p(\vec{w})|S(\vec{v}, M)| \geq \frac{1}{2^n}\binom{n - D(M)}{j(\vec{w}, M)} \quad (10)$$

This allows us to compute an upper bound on the amount of leaked information.

$$\begin{aligned}\mathcal{L} &\leq -\sum_{M} p(M) \sum_{\vec{w}} \frac{1}{2^n} \log \frac{1}{2^n} \binom{n - D(M)}{j(\vec{w}, M)} \\ &\leq \sum_{M} p(M)(n - \frac{1}{2^n} \sum_{\vec{w}} \log \binom{n - D(M)}{j(\vec{w}, M)}) \\ &= n - \frac{1}{2^n} \sum_{M} p(M) \sum_{\vec{w}} \log \binom{n - D(M)}{j(\vec{w}, M)})\end{aligned} \tag{11}$$

Let us define $S(D(M)) = \sum_{\vec{w}} \log \binom{n-D(M)}{j(\vec{w},M)})$. $S(D(M))$ sums over all possible $\vec{w}$ and thus depends only on M. Since $0 \leq k(\vec{w}, M) \leq D(M)$, $S(D(M))$ is lowerbounded by $T(D(M))$:

$$\begin{aligned}T(D(M)) = &\sum_{wt=D(M)+1}^{n-1} \binom{n}{wt} \\ &\log(\min(\binom{n - D(M)}{wt - D(M)}, \binom{n - D(M)}{wt}))\end{aligned} \tag{12}$$

To further simplify $\mathcal{L}$, let $d \in [0, nm]$ be the number of 1's in the matrix M, and let $N(d, D(M))$ be the number of M matrices with d 1's spread across $D(M)$ non-zero rows. Finally, let $p(M_d)$ be the probability of a matrix with d 1's. Then Inequality (13) decomposes the summation in the last line of Inequality (11) as follows. The outer sum considers all the matrices with i non-zero rows. The inner sum considers all the matrices with d 1's (at least i and no more than $i\,m$, m being the number of columns).

$$\begin{aligned}\sum_{M} p(M)T(D(M)) &\geq \sum_{i=1}^{n} \sum_{d=i}^{i\,m} N(d, i)p(M_d)T(i) \\ &= \sum_{i=1}^{n} T(i) \sum_{d=i}^{i\,m} N(d, i)p(M_d) \\ &= \sum_{i=1}^{n} T(i)p(D(M) = i) \\ \mathcal{L} &\leq n - \frac{1}{2^n} \sum_{D(M)} p(D(M))T(D(M))\end{aligned} \tag{13}$$

The last two lines follow because $\sum_{d=i}^{i\,m} N(d, i)p(M_d) = p(D(M) = i)$ is the probability of having a matrix with i non-zero rows.

C. Upper Bound on Information Leakage

We conclude our analysis by computing the value of the bound on information leakage in the test configurations of Section IV. To this end, let η be the probability of an element in the M matrix's being 1, and let $\vec{r}$ be a row vector in matrix M. We can compute the probability of having a non-zero row vector in M as follows.

$$\rho_1 = p(\vec{r} \neq \vec{0}) = 1 - (1 - \eta)^m \tag{14}$$

TABLE II. F1 SCORE AND UPPER BOUNDS ON LEAKED INFORMATION IN THE CONFIGURATIONS OF SECTION IV.

	n	$\mathcal{L}$	F1 score
ML-100k, $m = 25$	660	660	0.6690
ML-100k, $m = 10$	660	505	0.6602
ML-100k, $m = 7$	660	399	0.6567
ML-100k, $m = 5$	660	338	0.6480
ML-100k, $m = 3$	660	283	0.6360

This allows us to rewrite Equation (13) to obtain:

$$\mathcal{L} \leq n - \frac{1}{2^n} \sum_{D(M)} \binom{n}{D(M)} (\rho_1)^{D(M)} (1-\rho_1)^{n-D(M)} T(D(M)). \tag{15}$$

For the configuration of Section IV, we obtain $\eta = 0.05$. Depending on the dataset and number of landmarks, this leads us to the values of F1 score and leaked-information bound shown in Table II. The results show that 5, and 3 landmarks allow *H&S* to provide good similarity scores while leaking respectively no more than 51% and 43% of the information in the compact profiles. Also, while the value of $\mathcal{L}$ for $m = 25$ may seem bad, it does not mean that 25 landmarks leak all the information in the compact profiles, but only that the bound is not tight.

VI. RELATED WORK

Social networks in general, and collaborative filtering (CF) in particular, have attracted a number of efforts to propose decentralized implementations [21], [4], [22], [5], [23]. These approaches differ in (i) the mechanisms they use to connect users, and (ii) the level of privacy they provide.

A. Decentralized implicit social networks

Users may be connected through some explicit connections (as in [23]), which they declare and control (e.g. by "friending" or following other users, as in traditional centralized social networks), or through some implicit overlays. PeerSoN [23] for instance provides an explicit social networks, and relies on an external DHT infrastructure (e.g. OpenDHT [24]) to store information and publications related to users, even when users are disconnected, thus providing temporal uncoupling in users' communication.

H&S is based on this second type of approaches, which organize users in an implicit overlay based on their similarity. This implicit overlay can then be used for item recommendations [5], query extension [8], search [22], and news propagation [2], [3]. These works exploit the local neighborhoods constructed around each users to construct recommendations, or/and route queries and news to nodes and communities most likely to contain information related to them or have an interest in them. These approaches construct in practice distributed KNN graphs [25], [5]. They are therefore closely related to top-k processing algorithms [26], in particular when these approaches only exploit local information, thus lending themselves to decentralized implementations.

B. Privacy protection in peer-to-peer social networks

The means of privacy protection in decentralized social networks depend on the nature of the connections created

between users. In explicit P2P social networks, i.e. in networks in which overlay links mirror social relationships, the main focus lies in the integrity and confidentiality of the peer-to-peer links. PeerSoN [23] for instance relies on public/private key pairs to allow each user to control access to her private data. Safebook [27] extends this approach by relying on a *trusted identification service* (TIS), and routing mechanism combining onion-routing and real-life trusted relationships (*matryoshkas*) to hide a user's node id.

This type of approaches cannot however be directly transposed to implicit overlay networks, in which users (through their associated node) might interact with many other users they do not know or trust. One solution is to inject noise in users' profiles [28], [29] to distort and thus protect users' private information. Unfortunately, other works have shown that these scheme are weak and still allow attackers to reconstruct missing information [30], [31].

[32] provides two-party protocols to compute the cosine similarity of a document private to one party, with a corpus of documents private to the other party without revealing any document to each other. The most efficient of these protocols uses Paillier's partially homomorphic cryptosystem to compute the cosine similarity in a privacy-preserving way. This provides strong privacy guarantees in the considered security model which is, unlike *H&S*, the semi-honest (aka honest but curious) adversary.

In [33], authors describe a Private Neighbor Collaborative Filtering (PNCF) algorithm which guarantees differential privacy of neighborhoods. They also adapt elements of differential privacy so that the specific requirements of CF are met, thus retaining utility of recommendations. Although this PNCF algorithm is designed for the centralized setting, it should be possible to make a decentralized version.

Private Profile Matching (PPM) [34], [35], [36] and private set intersection [37] are two related domains of research which may serve as bricks for decentralized KNN computation. However, to our knowledge, existing protocols in these domains rely on techniques such as homomorphic encryption that provide stronger privacy guarantees at the cost of much higher computational complexity (e.g. [38] reports computation times of several seconds for a similarity operation between a pair of users). This makes these solutions impractical for recommendation systems because distributed KNN computation requires users to frequently recompute their similarities with a large number of other users. *H&S* strikes a different balance than these approaches and provides an efficient solution with reasonable privacy.

VII. Conclusion

We have presented *Hide & Share* (*H&S* for short), a novel peer-to-peer similarity computation protocol for decentralized KNN computation. We have demonstrated both formally and experimentally that *H&S* protects the privacy of users in the context of a peer-to-peer recommender. This protection is provided while preserving the system's effectiveness. *H&S* introduces *landmarks*, random data structures that are shared between peers comparing their profiles. It leverages a combinations of cryptography and security mechanisms to ensure that these landmarks cannot be diverted to attack the privacy of users.

We have shown using three real-world datasets that *H&S* maintains a strong level of privacy while providing recommendations close to that of an open system with no particular privacy protection mechanism. We have also shown using preliminary attacks that the *Hide & Share* mechanism performs better than a randomization scheme. Finally we have proposed an upper bound on the amount of information leaked by our scheme.

Although recommendation has been our primary focus in this paper, the applicability of *H&S* is not limited to recommendation services. In the near future, we would like to investigate how our proposal might be applicable to other services, such as search, news propagation, and decentralized differential privacy. We also plan to improve our upper bound, and further investigate the properties of *H&S* under stronger attack models than those presented here, in particular if we assume that attackers have some (limited) collusion ability, or have some prior knowledge of items and users distribution in the system.

Acknowledgments

This work was partially funded by the Region of Brittany, France, by the French ANR project SocioPlug (ANR-13-INFR-0003), by the DeSceNt project granted by the Labex Comin-Labs excellence laboratory (ANR-10-LABX-07-01), and by the Google Focused Research Award Web Alter-Ego.

References

[1] M. Ekstrand, J. Riedl, and J. Konstan, *Collaborative Filtering Recommender Systems*. Now Publishers, 2011.

[2] A. Boutet, D. Frey, R. Guerraoui, A. Jégou, and A.-M. Kermarrec, "WHATSUP: A decentralized instant news recommender," in *Parallel and Distributed Processing Symposium, International*. IEEE Computer Society, 2013, pp. 741–752.

[3] R. Baraglia, P. Dazzi, M. Mordacchini, and L. Ricci, "A peer-to-peer recommender system for self-emerging user communities based on gossip overlays," *Journal of Computer and System Sciences*, vol. 79, no. 2, pp. 291–308, 2013.

[4] M. Bertier, D. Frey, R. Guerraoui, A.-M. Kermarrec, and V. Leroy, "The gossple anonymous social network," in *Proceedings of the ACM/IFIP/USENIX 11th International Conference on Middleware*. Springer-Verlag, 2010, pp. 191–211.

[5] S. Voulgaris and M. v. Steen, "Epidemic-style management of semantic overlays for content-based searching," in *Euro-Par 2005 Parallel Processing*, no. 3648. Springer Berlin Heidelberg, 2005, pp. 1143–1152.

[6] A. Boutet, D. Frey, A. Jégou, A.-M. Kermarrec, and H. B. Ribeiro, "Freerec: An anonymous and distributed personalization architecture," in *The First International Conference on Networked Systems*. Springer Berlin Heidelberg, 2013, pp. 58–73.

[7] D. Frey, A. Jégou, and A.-M. Kermarrec, "Social market: Combining explicit and implicit social networks," in *Stabilization, Safety, and Security of Distributed Systems*. Springer Berlin Heidelberg, 2011, no. 6976, pp. 193–207.

[8] M. Bertier, R. Guerraoui, V. Leroy, and A.-M. Kermarrec, "Toward personalized query expansion," in *Proceedings of the Second ACM EuroSys Workshop on Social Network Systems*. ACM, 2009, pp. 7–12.

[9] C. J. v. Rijsbergen, *Information Retrieval*, 2nd ed. Butterworths, 1979.

[10] U. Kuter and J. Golbeck, "SUNNY: a new algorithm for trust inference in social networks using probabilistic confidence models," in *Proceedings of the 22d National Conference on Artificial Intelligence*, vol. 2. AAAI Press, 2007, p. 1377–1382.

[11] S. Blake-Wilson and A. Menezes, "Authenticated diffe-hellman key agreement protocols," in *Selected Areas in Cryptography*. Springer Berlin Heidelberg, 1999, no. 1556, pp. 339–361.

[12] D. J. Bernstein, "NaCl: Networking and cryptography library." [Online]. Available: http://nacl.cr.yp.to

[13] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM*, vol. 13, no. 7, p. 422–426, 1970.

[14] P. L'Ecuyer, "Good parameters and implementations for combined multiple recursive random number generators," 1998.

[15] M. Matsumoto and T. Nishimura, "Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator," *ACM Transactions on Modeling and Computer Simulation*, vol. 8, no. 1, pp. 3–30, 1998.

[16] M. Blum, "Coin flipping by telephone a protocol for solving impossible problems," *ACM Special Interest Group on Algorithms and Computation Theory News*, vol. 15, no. 1, pp. 23–27, 1983.

[17] R. Cleve, "Limits on the security of coin flips when half the processors are faulty," in *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing*. ACM, 1986, pp. 364–369.

[18] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, "An algorithmic framework for performing collaborative filtering," in *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 1999, pp. 230–237.

[19] K. Goldberg, T. Roeder, and C. Perkins, "Eigentaste: A constant time collaborative filtering algorithm," *Information Retrieval*, vol. 4, pp. 133–151, 2001.

[20] M. Naor, "Bit commitment using pseudo-randomness," *Journal of Cryptology*, vol. 4, p. 151–158, 1991.

[21] V. Leroy, B. B. Cambazoglu, and F. Bonchi, "Cold start link prediction," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2010, pp. 393–402.

[22] X. Bai, M. Bertier, R. Guerraoui, A.-M. Kermarrec, and V. Leroy, "Gossiping personalized queries," in *Proceedings of the 13th International Conference on Extending Database Technology*. ACM, 2010, pp. 87–98.

[23] S. Buchegger, D. Schiöberg, L.-H. Vu, and A. Datta, "Peerson: P2p social networking: early experiences and insights," in *Proceedings of the Second ACM EuroSys Workshop on Social Network Systems*. ACM, 2009, pp. 46–52.

[24] S. Rhea, B. Godfrey, B. Karp, J. Kubiatowicz, S. Ratnasamy, S. Shenker, I. Stoica, and H. Yu, "Opendht: A public dht service and its uses," in *Proceedings of the 2005 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*. ACM, 2005, pp. 73–84.

[25] M. Jelasity and O. Babaoglu, "T-man: Gossip-based overlay topology management," in *Proceedings of the 3rd International Workshop on Engineering Self-Organising Applications*. Springer Berlin Heidelberg, 2005, pp. 1–15.

[26] W. Dong, C. Moses, and K. Li, "Efficient k-nearest neighbor graph construction for generic similarity measures," in *Proceedings of the 20th international conference on World wide web*. ACM, 2011, pp. 577–586.

[27] L. A. Cutillo, R. Molva, and T. Strufe, "Safebook: A privacy-preserving online social network leveraging on real-life trust," *IEEE Communications Magazine*, vol. 47, no. 12, pp. 94–101, 2009.

[28] R. Agrawal and R. Srikant, "Privacy-preserving data mining," in *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*. ACM, 2000, pp. 439–450.

[29] H. Polat and W. Du, "Svd-based collaborative filtering with privacy," in *Proceedings of the 2005 ACM Symposium on Applied Computing*. ACM, 2005, p. 791–795.

[30] Z. Huang, W. Du, and B. Chen, "Deriving private information from randomized data," in *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*. ACM, 2005, p. 37–48.

[31] H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar, "On the privacy preserving properties of random data perturbation techniques," in *Third IEEE International Conference on Data Mining*. IEEE Computer Society, 2003, pp. 99–106.

[32] M. Murugesan, W. Jiang, C. Clifton, L. Si, and J. Vaidya, "Efficient privacy-preserving similar document detection," *The VLDB Journal*, vol. 19, no. 4, pp. 457–475, 2010.

[33] T. Zhu, G. Li, Y. Ren, W. Zhou, and P. Xiong, "Differential privacy for neighborhood-based collaborative filtering," in *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. ACM, 2013, p. 752–759.

[34] B. Schoenmakers and P. Tuyls, "Private Profile Matching," in *Intelligent Algorithms in Ambient and Biomedical Computing*. Springer Netherlands, 2006, no. 7, pp. 259–272.

[35] M. Li, N. Cao, S. Yu, and W. Lou, "FindU: Privacy-preserving personal profile matching in mobile social networks," in *2011 Proceedings IEEE INFOCOM*, 2011, pp. 2435–2443.

[36] X. Liang, X. Li, K. Zhang, R. Lu, X. Lin, and X. Shen, "Fully Anonymous Profile Matching in Mobile Social Networks," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 9, pp. 641–655, 2013.

[37] E. D. Cristofaro and G. Tsudik, "Practical Private Set Intersection Protocols with Linear Complexity," in *Financial Cryptography and Data Security*. Springer Berlin Heidelberg, 2010, pp. 143–159.

[38] R. Zhang, Y. Zhang, J. Sun, and G. Yan, "Fine-grained private matching for proximity-based mobile social networking," in *2012 Proceedings IEEE INFOCOM*, 2012, pp. 1969–1977.