Shilling Attacks against Privacy-Preserving Collaborative Filtering

Ihsan Gunes, Alper Bilge, Cihan Kaleli, and Huseyin Polat
Anadolu University/Computer Engineering Department, Eskisehir, Turkey
Email: {ihsang, abilge, ckaleli, polath}@anadolu.edu.tr

***Abstract*—Although collaborative filtering with privacy schemes protect individual user privacy while still providing accurate recommendations, they might be subject to shilling attacks like traditional schemes without privacy. There are various studies focusing on either proposing privacy-preserving collaborative filtering schemes or developing robust recommendation algorithms against shilling attacks. However, such studies fail to address preventing shilling attacks or providing privacy, respectively. We investigate a privacy-preserving memory-based collaborative filtering scheme with respect to shilling attacks. We study how to design random and bandwagon shilling attacks against such scheme and scrutinize the effects of them on the system in terms of robustness using some real data-based experiments. We show that it is still possible to create attacks to manipulate a database containing masked data. Our empirical results demonstrate that random and bandwagon attacks designed to manipulate the privacy-preserving collaborative filtering scheme affect the system's robustness. Thus, more attention should be given to designing shilling attacks against recommendation schemes with privacy and correspondingly developing robust algorithms and detection strategies.**

***Index Terms*—collaborative filtering, privacy, shilling, recommendation.**

I. Introduction

To help their customers while searching over the Internet to buy various products, many e-commerce sites utilize collaborative filtering (CF) schemes. CF is a relatively new concept used for filtering and prediction purposes. Users can obtain predictions about their daily activities including but not limited to books to read, music CDs to listen, restaurants to eat, sites to see, and so on with the help of CF systems.

Traditional user-based CF schemes work, as follows [1]: (*i*) *Data collection*: Users' preferences about various items are collected and an $n \times m$ user-item matrix (**D**) is created, where n and m represent number of users and items, respectively. (*ii*) *Neighbor selection*: An active user (a) sends her ratings vector (**A**) and a query for the item of which she is looking for a prediction, referred to as the target item (q), to the system. Similarity between a and each user in the database is computed using a similarity metric. The best similar k users are selected as neighbors. (*iii*) *Recommendation estimation*: A prediction is estimated for a on item q using a prediction algorithm based on **A** and those k users' data. The recommendation for a on item q (p_{aq}) is finally returned to a.

Recommendation systems can be subjected to various attacks. For example, malicious users or competing companies can create fake user profiles and insert them into the system's database. The aim of such attacks is to manipulate system's output (estimated predictions) in favor of the attackers. It has been shown that CF algorithms are vulnerable to such attacks, referred to as shilling attacks [2] and [3]. In shilling attacks, the attacker creates bogus profiles using as much information as possible about the CF scheme she intends to attack [4]. She then sends fake profiles to the system as being an authentic user. The attacker designs shilling profiles in such a way so that estimated predictions for specific items are biased in favor of her. The intend might be either increasing or decreasing the popularity of some items. Thus, the attacks are categorized as *push* or *nuke* attacks according to their intent. Push attacks focus on increasing the popularity of the target items, while nuke attacks aim to decrease the popularity of them [4].

Preserving privacy while performing various data mining applications have been receiving increasing attention. To provide recommendations without violating individual users' confidential data, various schemes have been proposed [5]-[8]. Such schemes make it possible to offer accurate predictions while preserving privacy. To protect confidential data in CF systems, randomization methods are widely utilized. Such methods randomly perturb original data so that CF systems are not able to learn truthful data while still performing recommendation services. Although there are various privacy-preserving collaborative filtering (PPCF) schemes, which protect confidentiality, they are not investigated with respect to shilling attacks. Such PPCF methods can also be subjected to profile injection or shilling attacks.

In this study, we study how to design shilling attacks against privacy-preserving k-nn recommendation algorithm proposed by [6]. We also investigate how random and bandwagon attacks affect robustness of the algorithm. On one hand, privacy-preserving k-nn prediction scheme is proposed to offer referrals with privacy [6]. However, it is not investigated with respect to shilling attacks even though they might be subjected to such attacks. On the other hand, researchers deeply study

Manuscript received October 11, 2012; revised December 13, 2012.

doi: 10.12720/joams.1.1.54-60

the *k*-nn CF algorithm in terms of shilling attacks [4] and [9]. However, they fail to protect individual users' privacy. Therefore, we investigate such algorithm with respect to both privacy and shilling attacks.

The paper is structured, as follows: In Section II, we discuss related studies and explain the differences between our study and them. We then briefly present preliminaries in Section Ⅲ. In Section Ⅳ, we study how to design random and bandwagon shilling attacks based on masked data for attacking the PPCF scheme. We present our real data-based expeirments and their empirical outcomes in Section V. We finally conclude the paper and present some future research directions in Section VI.

II. Related Work

The work conducted by [10] has inspired the studies about shilling attacks. Dellarocas [10] proposes a set of mechanisms, which eliminate or reduce negative effects of fraudulent behavior in online reputation reporting systems. Shilling or profile injection attack concept was first introduced by [11] and [12], where the authors argue vulnerabilities of recommender systems against shilling attacks to stimulate specific predictions. O'Mahony [13] discusses shilling attack strategies against CF schemes. The author shows that some statistical knowledge about the user-item matrix **D** is enough to design attacks against CF systems holding **D**. Lam and Riedl [14] propose that utilized recommender system algorithm, whether recommendation or prediction is generated, and detectability of attacks by system operators, and properties of items being attacked have effects on shilling attacks. Lam and Riedl [15] discuss privacy with respect to value of information and shilling attacks. In [16], the authors discuss the issues related to the following two questions: How can it be guaranteed that personal data collected for filtering purposes will never be leaked without users' permission? And how the customers are sure that the predictions they receive have not been modified? Mobasher *et al.* [17] and [18] discuss specific shilling attack types like random, average, bandwagon, and love/hate attacks. To design shilling attacks against CF systems, the attackers need some knowledge about the system they intend to attack [9] and [18]. The knowledge that might be needed can be mean rating for each item or user, standard deviation of the ratings of each item or user, ratings distribution, and so on. Some attacks might require very detailed knowledge about the CF system, referred to as the high-knowledge attacks; or require system independent knowledge, referred to as the low-knowledge attacks [18]. Informed attacks require high degree of domain knowledge to select appropriate items and ratings to design an attack profile [9].

Due to increasing popularity of privacy, various PPCF schemes have been proposed. Canny [5] proposes a scheme in which a community of users can compute a public "aggregate" of their data, which allows personalized singular value decomposition-based predictions to be computed by members of the community or by outsiders. Polat and Du [6] and [7] propose a privacy-preserving scheme to provide predictions based on a memory-based CF scheme. In [19], the authors discuss the effects of variably masked data on accuracy. In their scheme, each user independently masks her data using uniform or Gaussian distribution with a variable standard deviation. Yakut and Polat [20] propose a scheme to provide Eigentaste CF algorithm-based predictions while preserving privacy. To achieve confidentiality, the authors utilize randomized perturbation techniques (RPT). In [8], the authors study how to achieve binary ratings-based referrals while preserving privacy. Verhaegh *et al.* [21] propose to utilize encryption techniques to achieve individual privacy in memory-based central-server CF algorithms.

Some of the abovementioned studies focus on designing shilling attacks against CF systems and discuss their effects. However, they do not consider preserving privacy. The other group of the studies, on the other hand, studies how to protect privacy in recommendation algorithms. However, they do not focus on shilling attacks and their effects on such schemes. Hence, each group of the studies focuses on one aspect of CF schemes only. Unlike such studies, we focus on both preserving individual user confidentiality and designing shilling attacks against a privacy-preserving memory-based prediction algorithm. We propose designing random and bandwagon attacks on masked databases and discuss effects of such attacks on robustness.

III. Preliminaries

A. K-nn CF Algorithm with Privacy

The prediction algorithm proposed by [1] utilizes z-score normalization. If v_{uj} is user u's vote on item j, $\overline{v_u}$ is the mean vote for the user u, and σ_u is the standard deviation for the user u, then the z-score (z_{uj}) can be defined as $z_{uj} = (v_{uj} - \overline{v_u}) / \sigma_u$. The authors compute a weighted average of the z-scores, as follows:

$$p_{aq} = \overline{v_a} + \sigma_a \times \frac{\sum_{u \in U} w_{au} \times z_{uq}}{\sum_{u \in U} w_{au}} \quad (1)$$

where U is the set of neighbors and the similarity between a and user u (w_{au}) can be calculated, as follows:

$$w_{au} = \frac{\sum_{j \in J} (v_{aj} - \overline{v_a})(v_{uj} - \overline{v_u})}{\sigma_a \sigma_u} \quad (2)$$

where the summation over J is over the items for which both a and the user u rated; and σ_a and σ_u are the standard deviations of the a's and u's ratings, respectively.

Due to various privacy risks (like unsolicited marketing, price discrimination, profiling, and so on) posed by many CF schemes [22], users usually not willing to give their preferences about various products to online vendors. Polat and Du [6] and [7] propose the

following data disguising scheme to mask users' data while providing predictions using the *k*-nn algorithm:

- The server decides on distributions of perturbing data (uniform or Gaussian), data masking parameters (σ and μ), and methods to select the parameters, and let each user know.
- Each user u calculates the z-scores. Then, each user u creates m_r random values (r_{uj} values) drawn from chosen distribution, where m_r is the total number of rated items.
- Then, each user u adds those random values to her z-score values and generates the disguised z-scores $z'_{uj} = z_{uj} + r_{uj}$ for $j = 1, 2, \ldots, m_r$.
- Finally, each user u sends the disguised z-scores (z'_{uj} values) to the server, which creates the disguised user-item matrix (**D'**).

B. Shilling Attack

To manipulate the outcomes of CF schemes in favor of their advantages, attackers might create shilling attack profile depicted as in Table I, which is first defined by [2] and [17]; and they insert fake profiles into the attacked system's database. As seen from Table I, there are four sets of items in a typical attack profile [2] and [17]. A set of items, I_S, is determined by the attacker together with a particular rating function δ to form the characteristics of the attack. Another set of items, I_F, is selected randomly with a rating function θ to impede detection of an attack. A unique item (i_t) is targeted with a rating function, γ, to form a bias on. Remaining items are left unrated indicated as $I_\varnothing$, as seen from Table I.

We consider random and bandwagon attacks only, which happen to be low-knowledge attacks requiring limited amount of information. Moreover, they can be used as either push or nuke attacks. *Random attack* [17] operates through attack profiles with ratings to randomly chosen empty cells around system overall mean and r_{max} or r_{min} to target item for push and nuke attacks, respectively. Both of them are easy to implement.

In *bandwagon attacks*, an attacker generates profiles with high ratings to well-known popular items and the highest possible rating to the target item so that inserted fake profiles can easily be associated with respect to similarity to other users in the system and push the predictions to the target item. Bandwagon attack is also easy to implement [23] and [24]. To nuke the target item rather than to push its prediction value, the attacker can simply generate profiles on giving lowest possible ratings to target item, referred to as bandwagon nuke attack.

IV. Shilling Attack Profile Designs Against Perturbed Databases

PPCF applications collect masked preferences from users due to privacy concerns. Although researchers have proposed various shilling attack strategies against non-private CF schemes, their effects on privacy-preserving frameworks have not been studied. We investigate how to implement random and bandwagon attacks on masked data and scrutinize robustness of *k*-nn based PPCF algorithm.

TABLE I. General Form of an Attack Profile

I_S			I_F			$I_\varnothing$			
i_1^S	…	i_k^S	i_1^F	…	i_l^F	$i_1^\varnothing$	…	$i_v^\varnothing$	i_t
δ_1	…	δ_k	θ_1	…	θ_l	Ø	Ø	Ø	γ

The basic strategy for an attacker is to infiltrate as much fake profiles as possible into neighborhood by ensuring strong and positive correlation with a set of users in order to manipulate prediction values. However, while ensuring such high correlations, those shilling profiles need not to be recognized easily. Therefore, attack profile design shall balance trade-off between efficiency and detectability of such attacks. In this section, we describe modified versions of random and bandwagon attacks to manipulate perturbed collections.

A. Modified Random Attack Model

Random attack model is easy to implement and requires low knowledge compared to the other models [25]. A random attack can be performed to a push or nuke a target item's prediction value. In regular random attack model, filler items are chosen randomly among all but target items and assigned random ratings drawn from a distribution based on overall distribution of user ratings. Additionally, the set of selected items is empty. Random attack model can be defined, as follows:

- The set of selected items is empty ($I_S = \varnothing$).
- Filler items (I_F) are randomly selected from $I - \{i_t\}$ set, where the density of filler items is a predetermined value.
- Assigned rating value for each filler item is drawn from a normal distribution with system's overall rating mean and standard deviation.
- Target item (i_t) is assigned to r_{max} or r_{min} for push and nuke attacks, respectively.

The above steps follow the same strategy to produce shilling profiles against an unmasked database. However, since the server collects masked preferences from all users, attack profiles must also be disguised as genuine users do, as well. Such perturbation can be performed as explained in Section III. Although random perturbation protocol allows performing aggregate algebraic operations on masked data with minor flaw, it principally alters like/dislike properties of individual ratings. The key aspect in producing modified random attack profiles is to preserve push/nuke characteristics of target item's rating value after perturbation. Thus, a positive *z*-score value for a pushed item must also remain positive after perturbation and similarly negative *z*-score values for nuked items are required to be masked via negative random numbers. Accordingly, the rest of the modified random attack profile design can be described, as follows:

- The attacker calculates the z-scores for each shilling profile fp prior to perturbation.
- According to predefined distribution (uniform or Gaussian) and data masking parameters (σ and μ), the attacker generates fp_r random values, where fp_r

is the total number of ratings in each shilling profile including the target item's rating.

- For each shilling profile fp, the attacker selects one of the random numbers, r_t, to mask the target item. Such selection is made randomly among positive or negative random numbers for push and nuke attack strategies, respectively.
- After masking target item's rating with r_t, the attacker adds remaining disguising random numbers onto z-scores to obtain each masked modified random attack profile, fp′.

After producing masked random attack profiles, the attacker submits such fake profiles to the server. Note that the effects of how many random attack profiles to inject into the system's database (*attack size*) and how much ratings to insert into the shilling profiles (*filler size*) can be determined experimentally. Such effects can be measured by obtained shift in system's output. On the other hand, to avoid easy detection of attacks, the number of inserted shilling profiles must remain reasonable and each profile must be generated according to general characteristics of genuine users' rating patterns.

B. Modified Bandwagon Attack Model

Bandwagon attack is also a low-knowledge attack. It can be considered as an extension on random attack to increase efficiency, where an additional set of selected items are also included in the attack design. Such selected items comprise of highly popular items of which can be determined easily from the best seller or top ranked products lists. Since those popular items are much likely to get high valued ratings, they are also rated with the highest possible rating in shilling profiles. This way the attacker aims to increase the likelihood of leaking into the neighborhood by presenting high correlation with users over popular items. However, in privacy-preserving environment, ratings are disguised randomly jeopardizing resembling a high similarity over popular items. Due to characteristics of perturbation protocol, individual ratings are irreversibly disguised and consequently even popular items can be seemed as disliked. Thus, unlike regular bandwagon attack, we propose to rate relatively high ratings to popular items in modified bandwagon attack model, which will increase the likelihood of being similar to other disguised rating values. Modified bandwagon attack model can be characterized, as follows:

- The set of selected items (I_S) comprises of intensely highly rated products.
- In each profile fp, items in I_S are rated with relatively high ratings.
- Filler items (I_F) are randomly selected from $I - \{i_t \cup I_S\}$ set, where the density of filler items is a predetermined value.
- Assigned rating value for each filler item is drawn from a normal distribution with system's overall rating mean and standard deviation.
- Target item (i_t) is assigned to r_{max} or r_{min} for push and nuke attacks, respectively.
- Profiles are disguised similarly as in modified random attack model. Popular items' ratings are also randomly disguised because authentic users also disguise their ratings in the same way.

After producing modified bandwagon attack profiles, the attacker submits them into the system's database. Effects of attack size and filler size can be determined experimentally. Although the number of popular items is an important factor for modified attack, it is reasonable to keep it constant at small values to obstruct detectability.

V. Evaluation

We conducted real data-based experiments to assess modified attacks' effects on the system with respect to some controlling parameters. Researchers define two controlling parameters; *filler size* and *attack size*, for performing successful shilling attacks. *Filler size* refers to the number of unrated cells chosen to be filled with fake ratings while creating the attacking profile [26]. *Attack size* can be measured as a percentage of the pre-attack user count [2]. Hence, we conducted different sets of trials to show how shilling attacks affect the PPCF scheme with varying values of the parameters.

A. Data Set and Evaluation Criteria

In the experiments, we utilized MovieLens Public (MLP) data set, which is collected by GroupLens (http://movielens.umn.edu/). It consists of 100K ratings collected from 943 users for 1,682 movies. The ratings are discrete 5-star numeric ratings. To evaluate the effects of the shilling attacks, we used *prediction shift* [25] metric, which is the average change in the predicted rating for the attacked item before and after the attack.

B. Experimental Results and Discussion

In all experiments, we followed *one-but-all* experimentation methodology in which at each of the iterations, one of the users is considered as the active user *a* and the rest of the set is taken as the training set. The proposed attacks target the different sets of 50 movies for push and nuke attacks. Such 50 movies were selected randomly within different ranges to represent characteristics of data set according to the attack model. Since it is unreasonable to try to push an item's popularity whose rating is already high or similarly nuke an unpopular item, we basically selected items with low averages to push and high averages to nuke. Table II shows the statistics of the 50 target movies, where cell values represent how many of these movies fall into the specified group.

TABLE II. Statistics of Target Movies

Ratings	*Pushed Items*		*Nuked Items*	
	1-2	2-3	3-4	4-5
1 - 50	30	15	12	18
51 - 150	-	3	5	6
151 - 250	-	1	2	3
250 and up	-	1	1	3
1 - 50	30	15	12	18
51 - 150	-	3	5	6

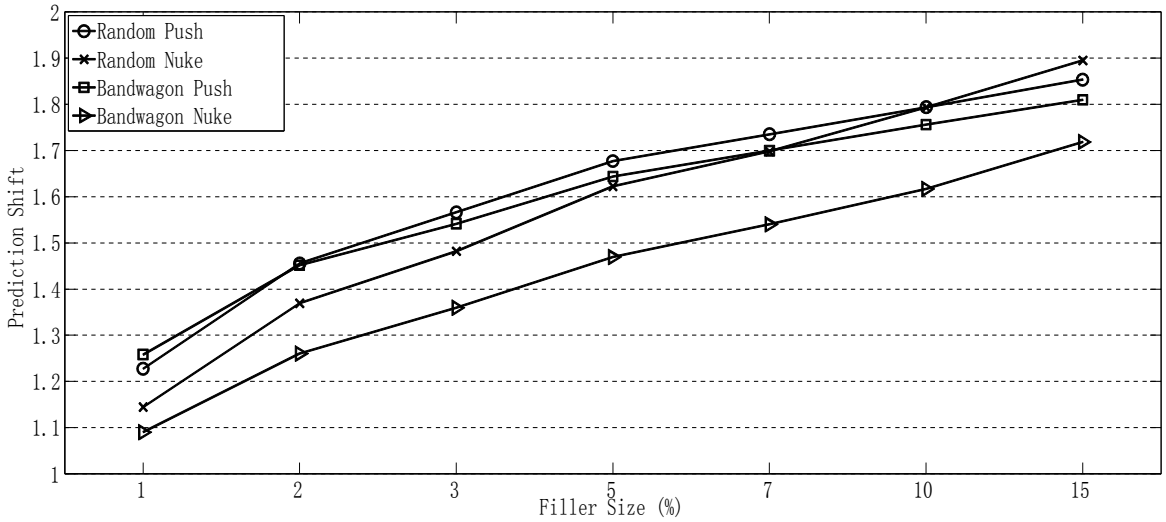

Figure 1. Prediction shift with varying filler size.

All target items were attacked individually for all users in the system. We employed modified random and bandwagon attack strategies for pushing/nuking the target items and we observed *prediction shift* to demonstrate relative change on predicted values for different attack models. Since overall system mean (μ) and standard deviation (σ) are required for generating I_F in shilling profiles, we set μ at 3.6 and σ at 1.1, as suggested by [18]**.** To form I_S in the modified bandwagon attack, we filled the selected items in I_S with rating values 4 and 5 randomly. The probability of being 4 for any item in I_S is 0.8 while it is 0.2 for being 5. According to aim of the attack, we set i_t at 5 or 1 for pushing or nuking, respectively. To disguise the shilling profiles, we generated random numbers from a Gaussian distribution with zero mean and 1.1 standard deviation. After generating attack profiles for both attack types, we disguised them, as explained in Section 4. We estimated predictions before and after injecting such profiles; and measured *prediction shift*. We kept number of neighbors utilized in prediction process at 60 as it is shown to be reasonable for CF systems [2]. All experiments were repeated 100 times due to randomization in perturbation process and average results were presented.

To investigate the effects of the modified random and bandwagon attacks on *k*-nn prediction algorithm with privacy, we first performed experiments with varying *filler size*. This is the number of ratings for the filler items added to fill out the attack profile, and thus, it is directly related to the effect of the attack. To magnify such impact of manipulation, we kept *attack size* at 15%, which is the largest value we tried. In the experiments, we varied *filler size* from 1% to 15%. We displayed the results in Fig. 1.

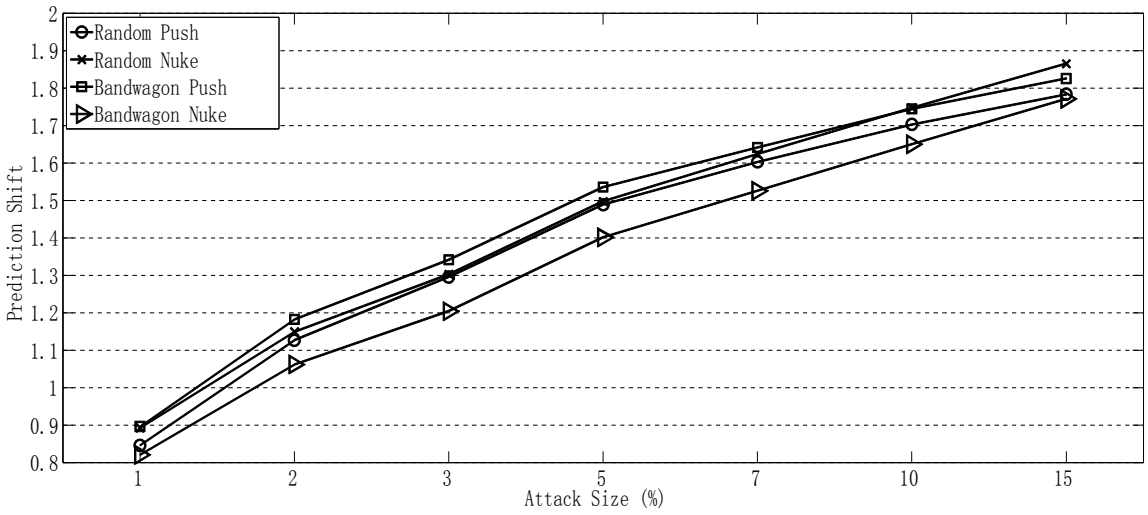

Figure 2. Prediction shift with varying attack size.

Although perturbation protocol disguises ratings, modified attacks are still effective on shifting produced predictions, as seen from the figure. With increasing *filler size*, the attacks become more effective. For push attacks, predictions are uplifted from 1.6 to 1.9 for *filler size* being 15%, which are significant in a 1-5 rating scale. Similarly, nuke attacks show a drop of about 1.7 to 1.9. Thus, it can be concluded that having more rating values in profiles has a positive effect on obtaining high similarity values and consequently successful in leaking

into neighborhood of active user. Surprisingly, for this set of experiments, random attack model is more successful than bandwagon attack. However, the difference between *prediction shift* values is insignificant. The reason for this phenomenon can be enlightened, as follows. Due to the data masking scheme, popular items are not supposed to have all high ratings after disguising. Therefore, having ratings to popular items are not very effective in PPCF schemes as they are in non-private CF environment.

We also experimented on varying *attack sizes*. Since the previous experiments show that the larger the *filler size*, the higher the prediction shift, we set *filler size* at 15% for these experiments; and we varied *attack size* from 1% to 15%. We demonstrated the results in Fig. 2.

As seen from the figure, *attack size* has also positive effect on triggering shifts on prediction values. Actually, this is obvious as the likelihood of shilling profiles to present in the neighborhood increases. However, unlike its effect on non-private schemes, the shift seems linear as *attack size* grows. Bandwagon attack again performs not better than random attack due to the disguising protocol, as explained before. It simply operates as a random attack due to random perturbation.

VI. Conclusions and Future Work

We examined the effects of inserting maliciousness into PPCF databases to manipulate predictions. We presented the modified versions of two low-knowledge shilling attack models, namely random and bandwagon attacks, to be utilized against PPCF schemes. We explained how to integrate them in masked databases by employing random perturbation protocol. According to obtained results, the proposed attack models demonstrate that *k*-nn PPCF algorithm is vulnerable to shilling attacks.

We will investigate the robustness of other PPCF algorithms including model-based ones. In addition, other shilling attack models such as love/hate and segmented attacks can be used to manipulate produced predictions. Also, effects of high-knowledge attacks should be compared against low-knowledge ones.

Acknowledgment

This work was supported by the Grant 111E218 from TUBITAK.

References

[1] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. T. Riedl, "An algorithmic framework for collaborative filtering," in *Proc. 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Berkeley, CA, USA, 1999, pp. 230-237.

[2] R. Bhaumik, C. A.Williams, B. Mobasher, and R. D. Burke, "Securing collaborative filtering against malicious attacks through anomaly detection," presented at 4th Workshop on Intelligent Techniques for Web Personalization, Boston, MA, USA, 2006.

[3] B. Mobasher, R. D. Burke, R. Bhaumik, and J. J. Sandvig, "Attacks and remedies in collaborative recommendation," *IEEE Intelligent Systems*, vol. 22, no. 3, pp. 56-63, 2007.

[4] B. Mobasher, R. D. Burke, R. Bhaumik, and C. A. Williams, "Towards trustworthy recommender systems: An analysis of attack models and algorithm robustness," *ACM Transactions on Internet Technology*, vol. 7, no. 4, pp. 23-60, 2007.

[5] J. Canny, "Collaborative filtering with privacy via factor analysis," in *Proc. 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Tampere, Finland, 2002, pp. 238-245.

[6] H. Polat and W. Du, "Privacy-preserving collaborative filtering using randomized perturbation techniques," in *Proc. 3rd IEEE International Conference on Data Mining*, Melbourne, FL, USA, 2003, pp. 625-639.

[7] H. Polat and W. Du, "Privacy-preserving collaborative filtering," *International Journal of Electronic Commerce*, vol. 9, no. 4, pp. 9-35, 2005.

[8] C. Kaleli and H. Polat, "Providing private recommendations using naïve Bayesian classifier," *Advances in Soft Computing*, vol. 43, pp. 168-173, 2007.

[9] R. D. Burke, M. P. O'Mahony, and N. J. Hurley, "Robust collaborative recommendation," *Recommender Systems Handbook*, Springer, New York, USA, 2011, pp. 805-835.

[10] C. Dellarocas, "Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior," in *Proc. 2nd ACM Conference on Electronic Commerce*, Minneapolis, MN, USA, 2000, pp. 150-157.

[11] M. P. O'Mahony, N. J. Hurley, and G. C. M. Silvestre, "Towards robust collaborative filtering," *Lecture Notes in Computer Science*, vol. 2464, pp. 87-94, 2002.

[12] M. P. O'Mahony, N. J. Hurley, and G. C. M. Silvestre, "Promoting recommendations: An attack on collaborative filtering," in *Proc. 13th International Conference on Database and Expert Systems Applications*, Aix-en-Provence, France, 2002, pp. 494-503.

[13] M. P. O'Mahony, "Towards robust and efficient automated collaborative filtering," Ph.D Dissertation, University College Dublin, Ireland.

[14] S. K. Lam and J. T. Riedl, "Shilling recommender systems for fun and profit," in *Proc. 13th International Conference on World Wide Web*, New York, NY, USA, 2004, pp. 393-402.

[15] S. K. Lam and J. T. Riedl, "Privacy, shilling, and the value of information in recommender systems," in *Proc. User Modeling Workshop on Privacy-Enhanced Personalization*, Edinburgh, UK, 2005, pp. 85-92.

[16] S. K. Lam, D. Frankowski, and J. T. Riedl, "Do you trust your recommendations? An exploration of security and privacy issues in recommender systems," *Lecture Notes in Computer Science*, vol. 3995, pp. 14-29, 2006.

[17] B. Mobasher, R. D. Burke, R. Bhaumik, and J. J. Sandvig, "Attacks and remedies in collaborative recommendation," *IEEE Intelligent Systems*, vol. 22, no. 3, pp. 56-63, 2007.

[18] B. Mobasher, R. D. Burke, R. Bhaumik, and C. A. Williams, "Towards trustworthy recommender systems: An analysis of attack models and algorithm robustness," *ACM Transactions on Internet Technology*, vol. 7, no. 4, pp. 23-60, 2007.

[19] H. Polat and W. Du, "Effects of inconsistently masked data using RPT on CF with privacy," in *Proc. ACM Symposium on Applied Computing*, Seoul, Korea, 2007, pp. 649-653.

[20] I. Yakut and H. Polat, "Privacy-preserving Eigentaste-based collaborative filtering," In *Proc. 2nd International Conference on Advances in Information and Computer Security*, Nara, Japan, 2007, pp. 169-184.

[21] W. Verhaegh, A. van Duijnhoven, P. Tuyls, and J. Korst, "Privacy protection in memory-based collaborative filtering," *Lecture Notes in Computer Science*, vol. 3295, pp. 61-71, 2004.

[22] L. F. Cranor, "'I didn't buy it for myself' Privacy and e-commerce personalization," in *Proc. ACM Workshop on Privacy in the Electronic Society*, Washington, DC, USA, 2003, pp. 111-117.

[23] M. P. O'Mahony, N. J. Hurley, and G. C. M. Silvestre, "Recommender systems: Attack types and strategies," in *Proc. 20th National Conference on Artificial Intelligence*, Pittsburgh, PA, USA, 2005, pp. 334-339.

[24] Z. Cheng and N. J. Hurley, "Robust collaborative recommendation by least trimmed squares matrix factorization," in *Proc. 22nd IEEE International Conference on Tools with Artificial Intelligence*, Arras, France, 2010, pp. 105-112.

[25] R. D. Burke, B. Mobasher, and R. Bhaumik, "Limited knowledge shilling attacks in collaborative filtering systems," in *Proc. Workshop on Intelligent Techniques for Web Personalization*, Edinburgh, UK, 2005.

[26] N. J. Hurley, Z. Cheng, and M. Zhang, "Statistical attack detection," in *Proc. 3rd ACM International Conference on Recommender Systems*, New York, NY, USA, 2009, pp. 149-156.

Mr. Gunes received his BSc and MSc degrees in Computer Engineering Department from Kocaeli University and Anadolu University, respectively. He is currently a PhD candidate in Computer Engineering Department at Anadolu University. His research interest is collaborative filtering in general; and specifically shilling attacks on collaborative filtering.

Mr. Bilge received his BSc degree in Electrical and Electronics Engineering Department and MSc degree in Computer Engineering Department from Anadolu University, Eskisehir, Turkey. He is currently a PhD candidate in Computer Engineering Department at Anadolu University, where he carries out research in privacy-preserving data mining.

Mr. Kaleli is with the Department of Computer Engineering at Anadolu University, Turkey. He got his Master's degree and PhD from the same department in 2008 and 2012, respectively. His research interest is privacy-preserving data mining in general; and specifically studies distributed data-based collaborative filtering with privacy.

Mr. Polat is an Associate Professor in Computer Engineering Department at Anadolu University, Turkey. He got his Master's degree and PhD from Computer Science Department at Syracuse University in 2001 and 2006, respectively. His research interests are collaborative filtering with privacy, private predictions on selected models, and privacy-preserving data mining.