Privacy-Preserving Content-Based Recommendations through Homomorphic Encryption

Z. Erkin[1], M. Beye[1], T. Veugen[1,2], and R. L. Lagendijk[1]
[1] Information Security and Privacy Lab, TU Delft, The Netherlands
[2] TNO, Delft, The Netherlands
{z.erkin, m.r.t.beye, p.j.m.veugen, r.l.lagendijk}@tudelft.nl

Abstract

By offering personalized content to users, recommender systems have become a vital tool in e-commerce and online media applications. Content-based algorithms recommend items or products to users, that are most similar to those previously purchased or consumed. Unfortunately, collecting and storing ratings, on which content-based methods rely, also poses a serious privacy risk for the customers: ratings may be very personal or revealing, and thus highly privacy sensitive. Service providers could process the collected rating data for other purposes, sell them to third parties or fail to provide adequate physical security. In this paper, we propose technological mechanisms to protect the privacy of individuals in a recommender system. Our proposal is founded on homomorphic encryption, which is used to obscure the private rating information of the customers from the service provider. While the user's privacy is respected by the service provider, by generating recommendations using *encrypted* customer ratings, the service provider's commercially valuable item-item similarities are protected against curious entities, in turn. Our proposal explores simple and efficient cryptographic techniques to generate private recommendations using a server-client model, which neither relies on (trusted) third parties, nor requires interaction with peer users. The main strength of our contribution lies in providing a highly efficient solution without resorting to unrealistic assumptions.

1 Introduction

Statistics show that e-commerce has exhibited rapid growth in the last decade [12]. According to analysts, the Internet presents a great place for customers to make a good deal as it is quite easy to compare prices from several retailers. To further increase their revenue, retailers have been successful in *personalizing* purchases by focusing on individuals rather than crowds. In particular, customer profiles are created and shopping patterns of customers are collected to be used in smart algorithms that generate a set of products, which are likely to be purchased by a target customer.

Among many smart algorithms, collaborative and content-based filtering techniques [1] have been proven effective in generating accurate recommendations for the customers. While collaborative filtering is based on similarity computations using ratings of multiple customers, content based filtering techniques are based on information on the items. In other words, a recommendation is generated for a particular customer by observing the characteristics of the previously purchased products. Both filtering techniques have their own application areas, but the accuracy of the predictions in collaborative filtering relies mostly on the amount of ratings data collected from the customers and in content-based filtering on proper product descriptions.

To improve the prediction accuracy, the retailers collect as much customer data as possible. While the benefits of personalized recommendations for the customers and the business are obvious, the collected data also create serious privacy risks for the individuals [18]. The service provider can easily identify and track individuals, especially when the collected data are combined with other publicly available resources, process the data for other purposes, transfer or sell them to third parties, or fail to provide adequate physical security. Consequences of either case will severely damage the privacy of the customers.

Our goal in this paper is to present a privacy-preserving version of a content-based recommender system within a realistic business model, which is practical for real-world use. In our scenario, a target customer provides his/her ratings to the service provider, which possesses an item-item similarity matrix. A recommendation for a target product is then generated as a weighted average of the products that the customer rated in the past. While the ratings of the customer are privacy-sensitive, the item-item similarity matrix of the service provider is commercially valuable, and thus, both should be kept private for their respective owners. Our proposal is to use homomorphic encryption [11] to realize linear operations on the encrypted data. Using homomorphic encryption provides privacy for the customer as his/her private data become inaccessible to the service provider, which does not have the decryption key. The service provider can still generate recommendations, but does this blindly, by performing homomorphic operations on the encrypted data. Because working in the encrypted domain introduces an overhead due to data expansion and expensive operations on large numbers, we address the challenge of creating an efficient solution by using look-up tables and data packing. While our work is certainly not the first to tackle this topic, we believe that the proposed techniques are appealing due to their simplicity, few assumptions and no need for any trusted third parties.

1.1 Related Work

The need for privacy protection for e-commerce, particularly those using collaborative filtering techniques, triggered research efforts in the past years. Among many different approaches, two main directions, which are based on data perturbation [2] and cryptography [13], have been investigated primarily in literature. Polat and Du in [16, 17] suggest hiding the personal data statistically, which has been proven to be an insecure approach [20]. Shokri et al. present a recommender system that is built on distributed aggregation of user profiles, which suffers from the trade-off between privacy and accuracy [19]. McSherry and Mironov proposed a method using differential privacy, which has a similar trade-off between accuracy and privacy [14]. Cissée and Albayrak present an agent system where trusted software and secure environment are required [6]. Atallah et al. proposed a privacy-preserving collaborative forecasting and benchmarking to increase the reliability of local forecasts and data correlations using cryptographic techniques [3]. Canny also presents cryptographic protocols to generate recommendations based on matrix projection and factor analyses, both of which suffer from a heavy computational and communication overhead [4, 5]. Erkin et al. propose more efficient protocols based on cryptographic techniques like homomorpic encryption and secure multi-party computation for recommender systems based on collaborative filtering [8, 10, 9]. However, in their proposals, the users are actively involved in the computations, which makes the overall construction more vulnerable to timeouts and latencies in the users' connections. Moreover, the computations that a single user has to perform involve encryptions and decryptions in the order of thousands, which makes the system impractical to run for the users.

1.2 Our Contribution

Existing literature on protecting private data using homomorphic encryption focuses on user-based collaborative filtering. As discussed before, content-based filtering is widely used in e-commerce and therefore, there is a need for generating private recommendation in a privacy-preserving manner. To the best of our knowledge, our proposal is the first one to tackle this problem within the context of content based algorithms. However, the cryptographic techniques deployed in the aforementioned related work introduce a considerable overhead in terms of computation and communication cost, which makes the privacy-preserving version of the algorithms impractical to use. To improve the state-of-the-art so that the private recommendations can be generated efficiently, we propose a cryptographic protocol for generating private recommendations using content-based filtering. We achieve this by considering several aspects. Firstly, we define our privacy requirements carefully. Consider that in previous works, private data, that is customer ratings and the final recommendations, and in some cases the intermediate values of the algorithms, are kept secret from the retailer by means of encryption. This is a valid requirement for preserving privacy in several recommender systems. However, if recommendations are generated for an e-commerce application, the next natural step for the customer is to purchase an item. If we assume the purchase history to be known to the retailer, attempting to hide which items have been rated by the customer by encrypting all of the ratings does not make sense. Secondly, we consider a realistic e-commerce application which is based on a server-client business model that does not involve any third party. Thirdly, we reduce the high cost of working in the encrypted domain significantly by avoiding expensive operations on the encrypted data and using look-up tables. The resulting cryptographic algorithm is the most efficient existing algorithm due to its simplicity and realistic assumptions as shown in the complexity analysis.

1.3 Organization

The paper is organized as follows. We describe our security assumptions, explain homomorphic encryption and summarize our notation in Section 2. After a brief introduction to content based filtering, we present two privacy-preserving versions of the filtering algorithm in Section 3. We also present in this section the complexity analyzes of the two versions. We conclude our paper in Section 4.

2 Preliminaries

In this section, we describe our security assumptions, briefly introduce homomorphic encryption and present the notation used in this paper.

2.1 Security Assumptions

We build our protocol on the semi-honest, also known as honest-but-curious, model. This assumption is realistic in the sense that retailers have a business reputation, which they do wish to protect by performing the required service properly, in this case generating recommendations. We assume that customers are interested in getting proper recommendations by providing valid ratings for the products that are presented in the system. Moreover, the actions of customers are limited by the software provided by the service provider, e.g. a browser plug-in or an applet. Obviously, we neglect attacks by third parties, assuming that the communication channels between the service provider and the customers are secured end-to-end using technologies like IPSec or SSL/TLS [7].

2.2 Homomorphic Encryption

The Paillier cryptosystem presented in [15] is *additively homomorphic*. This means that there exists an operation over the cipher texts $\mathcal{E}_{pk}(m_1)$ and $\mathcal{E}_{pk}(m_2)$ such that the result of that operation corresponds to a new cipher text whose decryption yields the sum of the plain text messages m_1 and m_2:

$$\mathcal{D}_{sk}\left(\mathcal{E}_{pk}(m_1)\cdot\mathcal{E}_{pk}(m_2)\right) = m_1 + m_2\ . \tag{1}$$

As a consequence of additive homomorphism, exponentiation of any cipher text yields the encrypted product of the original plain text and the exponent:

$$\mathcal{E}_{pk}(m)^e = \mathcal{E}_{pk}(m\cdot e)\ . \tag{2}$$

Given message $m \in \mathbb{Z}_n$, Paillier encryption is defined as:

$$\mathcal{E}_{pk}(m,r) = g^m \cdot r^n \bmod n^2\ , \tag{3}$$

where n is a product of two large primes p and q, g is a generator of order n and r is a random number in $\mathbb{Z}_n^*$. The tuple (g,n) is the public key. For decryption and further details, we refer readers to [15].

The Paillier cryptosystem is probabilistic. This is particularly important for encryption of plain texts within a small range. We denote the cipher text of a message m by $[\![m]\!]$ and omit the key for the sake of simplicity.

2.3 Notation

We summarize our notation in Table 1.

Table 1: Symbols and their descriptions.

L	Number of items	$\mathcal{S}$	item-item similarity matrix
$\mathcal{I}$	Set of similar items	N	Number of items in $\mathcal{I}$
M	Number of Alice's ratings	$s_{(i,j)}$	similarity between items i and j
$\vec{p}$	Alice's rating vector	p_i	i^{th} element of $\vec{p}$
r_i	Recommendation for item i	δ	Threshold
$\vec{w}$	Vector of weighted sums	w_i	Weighted sum for item i
$\vec{v}$	Vector of sums of similarities	v_i	Sum of similarities for item i
k	bit length of ratings and scaled similarities	n	Paillier message space
$\tilde{w}$	Packed weighted sums	Δ	Bit length of weighted sums
N_e	Number of encryptions to pack all w_i's	$[\![\mathrm{m}]\!]$	Encryption of m

3 Privacy-Preserving Recommender System

In this section, we first summarize the content based recommender system algorithm on plain text data and then describe the privacy-preserving version in detail.

3.1 Content-based Recommender System Algorithm

We assume that Alice, as a user in the recommender system, has a preference vector $\vec{p}$ of dimension L, which contains $M < L$ positive ratings on content items. The remaining, non-rated items have preference value zero. Bob, the service provider, holds an item-item similarity matrix $\mathcal{S}$ of size $L \times L$, whose elements are the similarity measures between item i and item j, denoted by $s_{(i,j)}$. To generate recommendations for Alice, we follow the following procedure.

- Alice sends $\vec{p} = (p_1, p_2, \ldots, p_L)$ to Bob.
- Bob finds the set of similar items $\mathcal{I}$ to the rated items in $\vec{p}$ using similarity matrix $\mathcal{S}$. Bob creates this set by selecting the items that have a similarity to every rated item in $\vec{p}$ above a threshold δ.
- For every item $i \in \mathcal{I}$, which has N items in total, Bob generates recommendation as follows:

  $$r_i = \frac{\sum_{m=1}^{M} p_m \cdot s_{(i,m)}}{\sum_{m=1}^{M} s_{(i,m)}} \ , \tag{4}$$

  assuming that Alice has her ratings p_i for $i \in 1, \ldots, M$.
- Bob sends the ratings vector $\vec{r}$ for $r_i \in \mathcal{I}$ and the set $\mathcal{I}$ to Alice.

In the above algorithm, there are mainly three types of data that require protection with regard to privacy: Alice's preference vector, Bob's item-item similarity matrix and the generated recommendations. Notice that the dimension of $\vec{p}$ is L, which is a large number for a typical recommender system. It is natural for Alice not to rate all of the items but a small fraction. In fact, this preference vector is mostly sparse, approximately 99% in the mostly used research data sets such as MovieLens. Moreover, due to the way online applications work, the service provider has (partial) information on the *seen* items, e.g. by observing the visited pages or past purchases. Besides, Bob suggests recommendations on a *known* set of items. Because of these observations, Alice's privacy depends on hiding her *taste*, that is her liking or disliking a particular item, and the content of the recommendations rather than keeping the rated items secret.

While Alice's taste and final recommendations are privacy-sensitive, the item-item matrix $\mathcal{S}$ is commercially valuable for Bob. $\mathcal{S}$ cannot be made public or sent to Alice to generate recommendations since this will destroy Bob's business. In the following sections, we describe a cryptographic mechanism to protect the content of Alice's preferences, the item-item similarity matrix and the generated recommendations.

3.2 Privacy-Preserving Algorithm (PPA)

We assume that Alice has a Paillier key pair and is capable of performing encryption and decryption. The privacy-preserving version of the content-based recommender system described before works as follows.

1. Alice encrypts the non-zero elements of $\vec{p}$, which are in total M elements, where $M \ll L$, using her public key and sends them to Bob: $[\![\vec{p}]\!] = ([\![p_1]\!], [\![p_2]\!], \ldots, [\![p_M]\!])$. We assume that Alice rates the first M items for the sake of simplicity.

2. Bob creates $\mathcal{I}$, the set of similar items by selecting items in $\mathcal{S}$ that have $s_{(i,j)} > \delta$ for each p_i. We assume that $\mathcal{I}$ has N items, where $N = L - M$ in the worst case.

3. Bob computes a weighted sum for item $i \in \mathcal{I}$.

$$[\![w_i]\!] = \prod_{m=1}^{M} [\![p_m]\!]^{s_{(i,m)}} = \left[\!\!\left[\sum_{m=1}^{M} p_m \cdot s_{(i,m)}\right]\!\!\right] , \tag{5}$$

where we scale and round $s_{(i,m)}$ to an integer to enable calculating in the encrypted domain.

4. Bob also computes the sum of similarities for item i, which are also scaled to k-bit integers, similar to the ratings:

$$v_i = \sum_{m=1}^{M} s_{(i,m)} . \tag{6}$$

5. Bob sends Alice $[\![\vec{w}]\!] = ([\![w_1]\!], [\![w_2]\!], \ldots, [\![w_N]\!])$ and $\vec{v} = (v_1, v_2, \ldots, v_N)$.

6. Alice decrypts $[\![\vec{w}]\!]$ and computes recommendations:

$$r_i = \frac{w_i}{v_i} \text{ for } i \in \{1, \ldots, N\} . \tag{7}$$

While the above algorithm is straightforward to apply in the encrypted domain, there are a number of challenges in realization, considering performance. Recall that Alice encrypts her preferences using the Paillier cryptosystem, which introduces a considerable data expansion: a 4-bit rating turns into a 2048-bit cipher text by using a key of size 1024-bits for a modest security level. Therefore, computation of encrypted $[\![w_i]\!]$ becomes computationally expensive since it involves exponentiations of large numbers, which creates a serious performance concern for applying this algorithm in real life. The size of $\mathcal{I}$ also creates additional computational and communication overhead since the number of similar items in a real system can be in the order of thousands. Generating recommendations for a large set of items can thus become overwhelming for Bob. Moreover, transmission of these N recommendations, each encrypted separately, requires high bandwidth. Therefore, we investigate techniques to reduce the computational and communication costs of the above algorithm taking these observations into account.

3.2.1 Look-up Table (PPA-LUT)

Assuming that Alice has M rated items, generating N recommendations under encryption is computationally expensive for Bob. To improve the performance in terms of computation, we can use a look-up table. Consider Eq. (4) and the following recommendations:

$$\begin{aligned} r_1 &= p_1 \cdot s_{(1,1)} + p_2 \cdot s_{(1,2)} + \ldots + p_M \cdot s_{(1,M)} \\ r_2 &= p_1 \cdot s_{(2,1)} + p_2 \cdot s_{(2,2)} + \ldots + p_M \cdot s_{(2,M)} \\ &\ldots \\ r_N &= p_1 \cdot s_{(N,1)} + p_2 \cdot s_{(N,2)} + \ldots + p_M \cdot s_{(N,M)} , \end{aligned} \tag{8}$$

where we omit the denominator. Alice's preferences are multiplied with different $s_{(i,j)}$, which are positive k-bit integers. Recall that multiplications turn into intensive exponentiations in the encrypted

Table 2: Complexity of the privacy-preserving recommender system.

	PPA[1]		**PPA-LUT**[2]	
	Alice	**Bob**	**Alice**	**Bob**
Encryption	M	-	M	-
Decryption	N	-	N	-
Multiplication	-	$N \cdot (M-1)$	-	$M(N+2^k)-N$
Exponentiation	-	$N \cdot M$	-	-
Communication	M	N	M	N

[1] **PPA**: Privacy-Preserving Algorithm

[2] **PPA-LUT**: Privacy-Preserving Algorithm with Look-Up Table

domain as given in Equation 5. To simplify the computations, Bob can create a look-up table for Alice. For this purpose, Bob computes $[\![p_i]\!]^j$ for $j \in \{1, \ldots, 2^k\}$ for every p_i and replaces the appropriate values for the computation of recommendations. It is clear that to generate N recommendations, Bob should only compute $M \cdot 2^k$ exponentiations over mod n^2 rather than $M \cdot N$ exponentiations. Moreover, this exponentiations can be implemented as a chain of multiplications since $[\![p]\!]^j = [\![p]\!] \cdot [\![p]\!]^{j-1}$.

3.3 Complexity

The complexity of the proposed mechanism for generating private recommendations for Alice depends on the operations on the encrypted data. We assume that the cost of operations in the plain domain is negligible. In Table 2, we present the number of operations for encryption, decryption, multiplication and exponentiation for Alice and Bob for the two versions of the privacy-preserving recommender system. We also give the communication cost in the number of encryptions to be transmitted.

Note that exponentiation with a k-bit number takes roughly $1.5k$ multiplications.

From Table 2 we see that using a look-up table reduces the complexity significantly. On the other hand, while data packing reduces the communications cost, it also introduces an extra computational burden to Bob.

The protocol we presented in this paper has only one round of interaction, which means Alice sends her preferences and gets values from Bob to compute the recommendations herself.

3.4 Security Discussion

Our cryptographic protocol is based on the semi-honest security model that assumes Alice and Bob follow the protocol steps. Assuming that the communication between Alice and Bob is secured, meaning that any third party is prevented from intervening, we focus on analyzing whether our privacy requirements are satisfied. Note that our security assumptions only rely on the security of the cryptosytem, namely Paillier, and do not rely on any other security assumptions.

Recall that our goal is to hide Alice's preferences and final recommendations from Bob and Bob's item-item similarity matrix from Alice. Alice, who has the decryption key, encrypts her preference vector using the Paillier cryptosystem, which is semantically secure [15]. This means that Bob cannot observe the content of the encryptions even though Alice has ratings in a small range. Bob computes the weighted sums under encryption using the secure Paillier cryptosystem. This guarantees the secrecy of the generated recommendations towards Bob.

We have only one aspect to consider regarding the security of our protocol: can Alice deduce meaningful information on Bob's item-item similarity matrix by having $\vec{p}$, w and v in clear text? It is clear from Eq. (4) that for $M \cdot N$ unknowns ($s_{(i,j)}$'s), Alice has only M p_i's, N and v's and N w's. Therefore, it is not possible for Alice to solve this linear system with $2N$ equations and $M \cdot N$ unknowns without further information as long as $M > 2$.

4 Conclusion

Customization of purchases in e-commerce provides advantage to the retailers to increase their revenue. As a simple and effective method, content-based recommender systems have been widely used in business. Like other techniques, content-based recommender systems rely on customer's preferences, which can be highly privacy sensitive and open to misuse by even the retailer itself. We believe that it is possible to protect customers' private data without disrupting the service by using homomorphic encryption. In our proposal, we encrypt the customer's private data and provide a privacy-preserving version of the recommender system with which the retailer can generate recommendations as usual. We minimize the overhead introduced by working in the encrypted domain by employing look-up tables, which replaces expensive operations on the encrypted data. Our proposal is suitable for business as it is built on the server-client model and does not require any third parties, which is a difficult requirement to fulfil in the real-world. The complexity analysis show that privacy-preserving content-based recommender system is highly efficient.

References

[1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Eng.*, 17(6):734–749, 2005.

[2] R. Agrawal and R. Srikant. Privacy-preserving data mining. *SIGMOD Rec.*, 29:439–450, May 2000.

[3] S. Agrawal, V. Krishnan, and J. Haritsa. On addressing efficiency concerns in privacy-preserving mining. *Proc. of 9th Intl. Conf. on Database Systems for Advanced Applications (DASFAA)*, pages 113–124, 2004.

[4] J. F. Canny. Collaborative filtering with privacy. In *IEEE Symposium on Security and Privacy*, pages 45–57, 2002.

[5] J. F. Canny. Collaborative filtering with privacy via factor analysis. In *SIGIR*, pages 238–245, New York, NY, USA, 2002. ACM Press.

[6] R. Cissée and S. Albayrak. An agent-based approach for privacy-preserving recommender systems. In *AAMAS '07: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, pages 1–8, New York, NY, USA, 2007. ACM.

[7] N. Doraswamy and D. Harkins. *IPSec: The New Security Standard for the Internet, Intranets, and Virtual Private Networks.* Prentice Hall PTR, Upper Saddle River, NJ, USA, 1999.

[8] Z. Erkin, M. Beye, T. Veugen, and R. L. Lagendijk. Privacy enhanced recommender system. In *Thirty-first Symposium on Information Theory in the Benelux*, pages 35–42, Rotterdam, 2010.

[9] Z. Erkin, M. Beye, T. Veugen, and R. L. Lagendijk. Efficiently computing private recommendations. In *International Conference on Acoustic, Speech and Signal Processing-ICASSP*, pages 5864–5867, Prag, Czech Republic, May/2011 2011.

[10] Z. Erkin, T. Veugen, and R. L. Lagendijk. Generating private recommendations in a social trust network. In *The International Conference on Computational Aspects of Social Networks (CASoN 2011)*, Salamanca, Spain, 2011. IEEE.

[11] C. Fontaine and F. Galand. A survey of homomorphic encryption for nonspecialists. *EURASIP Journal on Information Security*, 2007, 2007.

[12] L. Indvik. Forrester: E-commerce to reach nearly $300 billion in U.S. by 2015. `http://mashable.com/2011/02/28/forrester-e-commerce/`, February 28 2011. Online.

[13] Y. Lindell and B. Pinkas. Privacy preserving data mining. In *Journal of Cryptology*, pages 36–54. Springer-Verlag, 2000.

[14] F. McSherry and I. Mironov. Differentially private recommender systems: building privacy into the net. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 627–636, New York, NY, USA, 2009. ACM.

[15] P. Paillier. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In J. Stern, editor, *Advances in Cryptology — EUROCRYPT '99*, volume 1592 of *LNCS*, pages 223–238. Springer, May 2-6, 1999.

[16] H. Polat and W. Du. Privacy-preserving collaborative filtering using randomized perturbation techniques. In *ICDM*, pages 625–628, 2003.

[17] H. Polat and W. Du. SVD-based collaborative filtering with privacy. In *SAC '05: Proceedings of the 2005 ACM symposium on Applied computing*, pages 791–795, New York, NY, USA, 2005. ACM Press.

[18] N. Ramakrishnan, B. J. Keller, B. J. Mirza, A. Y. Grama, and G. Karypis. Privacy risks in recommender systems. *IEEE Internet Computing*, 5(6):54–62, 2001.

[19] R. Shokri, P. Pedarsani, G. Theodorakopoulos, and J.-P. Hubaux. Preserving privacy in collaborative filtering through distributed aggregation of offline profiles. In *RecSys '09: Proceedings of the third ACM conference on Recommender systems*, pages 157–164, New York, NY, USA, 2009. ACM.

[20] S. Zhang, J. Ford, and F. Makedon. Deriving private information from randomly perturbed ratings. In *Proceedings of the Sixth SIAM International Conference on Data Mining*, pages 59–69, 2006.