Privacy-Preserving Content-Based Recommender System[*]

Z. Erkin, M. Beye, T. Veugen[†] and, R. L. Lagendijk
Information Security and Privacy Lab
Delft University of Technology
2628 CD, Delft, The Netherlands
{z.erkin, m.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.

Categories and Subject Descriptors

E.3 [**Data Encryption**]: Public key cryptosystems

General Terms

Security, algorithm, design, application, performance

Keywords

Content-based recommender systems, privacy, homomorphic encryption.

1. INTRODUCTION

Statistics show that e-commerce has exhibited rapid growth in the last decade [15]. 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 [23]. Content-based recommendation is by far the most used recommendation system in practice and used by well-known providers such as Amazon.com, where each of user's purchased and rated items are matched to similar items rather than matching the user to similar customers [17].

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 [22]. 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.

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

[*] This work is supported by STW Kindred Spirits Project.

[†] This author is also with TNO, P.O. Box 5050, 2600 GB Delft, The Netherlands.

perturbation [2] and cryptography [16], have been investigated primarily in literature. Polat and Du in [20, 21] suggest hiding the personal data statistically, which has been proven to be an insecure approach [26]. 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 [24]. McSherry and Mironov proposed a method using differential privacy, which has a similar trade-off between accuracy and privacy [18]. Cissée and Albayrak present an agent system where trusted software and secure environment are required [7]. 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 [5, 6]. 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 [9, 11, 10]. 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. In [12], Erkin *et al.* propose a cryptographic protocol that does not require active participation of the users, however the involvement of a semi-trusted third party in the protocol is necessary.

1.2 Our Contribution

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 [13] 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. However, using cryptographic techniques, particularly homomorphic encryption, introduce a considerable overhead in terms of computation and communication cost compared with the recommender systems with plain text data. This makes the privacy-preserving version of the algorithms impractical to use in real life. To improve the state-of-the-art so that the private recommendations can be generated more efficiently compared to existing work, we propose a cryptographic protocol for generating private recommendations using content-based filtering by taking the following aspects into account.

- 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 as in Amazon.com or the browser history as in Youtube to be known to the retailer, it is not necessary to hide the information that an item is rated, or visited, but the rating for that item should be protected. This assumption, where it holds, can help us to improve the efficiency significantly as there will be less values to be encrypted. In the case where visited items and the ratings are correlated, the ratings for all items, including the non-rated items, should be encrypted.
- A number of previous works such as [12] considers using a semi-trusted third party to be involved in the cryptographic protocol so that recommendations can be generated even when the users are off-line. However, it is not easy to find such entities in business. Therefore, we consider a realistic e-commerce application scenarios, which is based on a server-client business model that does not involve any trusted third party.
- We reduce the high cost of working in the encrypted domain significantly by using look-up tables and data packing, and avoiding expensive operations on the encrypted data such as secure comparison [12].

While our work is certainly not the first to tackle this topic, we believe that the proposed techniques in this paper are appealing due to their simplicity, few assumptions and no need for any trusted third parties. The resulting cryptographic algorithm performs considerably well due to its *simplicity* and realistic assumptions. The complexity analysis and experimental results show that our proposal improves the state-of-the-art in Signal Processing in the Encrypted Domain one step further.

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 three privacy-preserving versions of the filtering algorithm in Section 3. We also present the complexity analyses along with a security discussion in this section. We show the result of our experiments and compare our results with the previous work in Section 4. We conclude our paper in Section 5.

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 [8].

2.2 Homomorphic Encryption

The Paillier cryptosystem presented in [19] 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 [19].

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.

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 $i \in \mathcal{I}$.

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

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	$\llbracket$m$\rrbracket$	Encryption of m

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 $\llbracket \vec{w} \rrbracket = (\llbracket w_1 \rrbracket, \llbracket w_2 \rrbracket, \ldots, \llbracket w_N \rrbracket)$ and $\vec{v} = (v_1, v_2, \ldots, v_N)$.

6. Alice decrypts $\llbracket \vec{w} \rrbracket$ 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 $\llbracket w_i \rrbracket$ 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 domain as given in Equation 5. To simplify the computations, Bob can create a look-up table for Alice. For this purpose, Bob computes $\llbracket p_i \rrbracket^j$ 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 $\llbracket p \rrbracket^j = \llbracket p \rrbracket \cdot \llbracket p \rrbracket^{j-1}$.

3.2.2 Data Packing (PPA-LUT/DP)

After Bob generates $\llbracket w_i \rrbracket$ for $i \in \{1, \ldots, N\}$, he sends them to Alice. However, we can deploy data packing as in [4, 25, 10] to reduce the communication costs. Data packing is quite useful considering that generated recommendations in a typical recommender systems are only a few bits and the message space for public key cryptosystems is much larger, e.g. $n = 1024$ bits for Paillier with a modest key size. Therefore, packing multiple values in a single encryption is beneficial when operations in the subsequent steps are planned carefully.

Bob packs the $\llbracket w_i \rrbracket$'s as given in Algorithm 1, where $\Delta = 2k + \log(M)$ is the bit-length of the compartments in $\tilde{w}$ as Δ is the maximum bit-length of w_i's. At the end of this computation, Bob will have the packed sums: $\tilde{w} = w_N | \ldots | w_2 | w_1$, where $|$ denotes the concatenation. Then, Bob sends the packed recommendations to Alice as before.

Algorithm 1 Packing weighted sums.

```
$\tilde{w} = w_N$
for $i = 1$ to $N$ do
    $\llbracket \tilde{w} \rrbracket \leftarrow= \llbracket \tilde{w} \rrbracket^{2^{\Delta}} \cdot \llbracket w_{N-i} \rrbracket = \llbracket \tilde{w} \cdot 2^{\Delta} + w_{N-i} \rrbracket$ ,
end for
```

In the above data packing procedure, we assumed that all of the N weighted sums fit in a single encryption. Assuming that $s_{(i,j)}$'s and p_i's are both $k = 4$ bits and Alice has $M = 64$ ratings, the bit-length of w is $\Delta = 2k + \log(M) = 14$ bits. It is obvious that this number depends on M which is the number of rated elements in $\vec{p}$. Moreover, Bob generates N recommendations, meaning that $N \cdot \Delta$ bits are needed in total. Thus, $N_e = \lceil N \cdot \Delta / n \rceil$ encryptions will be used to pack all of the w_i's. So in our example $N_e = 1$ whenever $N \leq 73$. In practice, Bob can fine-tune the parameters M, N and δ to generate a desired number of recommendations and obtain a sufficient performance.

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 three versions of the privacy-preserving recommender system. Note that exponentiation with a k-bit number takes roughly $1.5k$ multiplications. We also give the communication cost in the number of encryptions to be transmitted.

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 [19]. 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 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. NUMERICAL RESULTS

We implemented the three versions of the privacy-preserving content-based recommender system: PPA, PPA-LUT and PPA-LUT/DP using C++ and GMP Library on a Linux machine with 64-bit microprocessor and 16 GB of RAM, running Suse 10.3. We created a synthetic data set and conducted our experiments using different parameters to test the performance of the three versions. Note that the cryptographic protocol has a complexity linear to the number of items in the data set, therefore we just focus on the choice of parameters and analyze the performance of these three versions considering run-time and bandwidth using the synthetic data set. Another important issue to consider is that our experiments provide numerical results for the worst case scneario, where Alice receives recommendations for the whole set of items. In a real deployment, much smaller set of items is recommended, meaning that the efficiency of the cryptographic protocol will be better, as discussed later in this section.

4.1 Run-time

Figure 1 shows the run-time of each implementation based on the parameters given in Table 3. We assume that Alice has rating from 1 to 10 and similarities are scaled and rounded to integers in the same range. Therefore, k is set to 4 bits. For our test, we assume that Alice has ratings for 98% of items as given in Table 3. We consider the worst case and assume that Bob generates $N = L - M$ recommendations for Alice, thus δ is not considered.

Our experiments show that PPA-LUT outperforms the other two with a run-time of 40 minutes with 102 400 items. As in shown in complexity analysis in Section 3.3, using data packing requires Bob to perform more operations on the encrypted data but demands less bandwidth for Bob and less computation for Alice. In PPA-LUT/DP, only packing 102 400 encrypted recommendations takes approximately 46% of the whole computation. This percentage increases to 96% for 1024 recommendations, where generating recommendations are less expensive compared to packing. As a conclusion, the choice between PPA-LUT and PPA-LUT/DP highly depends on the processing power and available bandwidth in a real deployment.

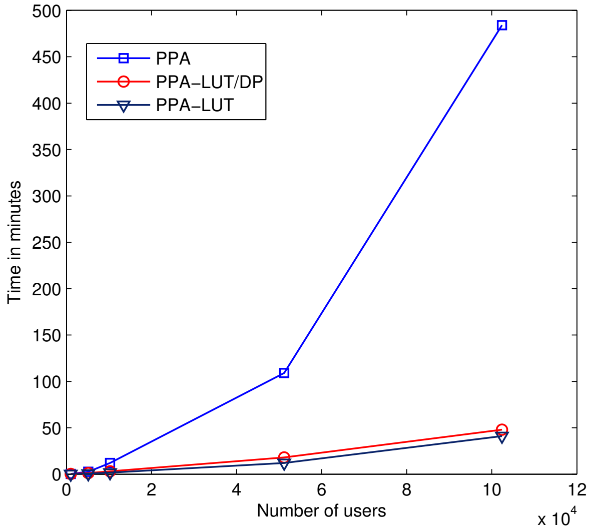

Figure 1: Run-time of privacy-preserving algorithm (PPA), with look-up table (PPA-LUT) and with look-up table and data packing (PPA-LUT/DP) in the worst-case scenario.

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

	PPA[1]		PPA-LUT[2]		PPA-LUT/DP[3]	
	Alice	**Bob**	**Alice**	**Bob**	**Alice**	**Bob**
Encryption	M	-	M	-	M	-
Decryption	N	-	N	-	N_e	-
Multiplication	-	$N \cdot (M-1)$	-	$M(N+2^k)-N$	-	$N(M+2^k)+N(\Delta+1)$
Exponentiation	-	$N \cdot M$	-	-	-	-
Communication	M	N	M	N	M	N_e

[1] **PPA**: Privacy-Preserving Algorithm
[2] **PPA-LUT**: Privacy-Preserving Algorithm with Look-Up Table
[3] **PPA-LUT/DP**: Privacy-Preserving Algorithm with Look-Up Table and Data Packing

Table 3: Parameters.

L	M	Δ in bits, for **PPA-LUT/DP**	N_e for **PPA-LUT/DP**
1024	32	13	13
5120	128	15	75
10240	512	17	171
51200	1024	18	915
102400	2048	19	1933

Compared to previous works, our results are highly promising to deploy in real life. Note that in our experiments, we generate recommendations for the whole data set, which is **not** the case in practice. Generating recommendations for a small set of items chosen by Bob has a run-time in the order of *milliseconds* since Bob only needs to create a very small look-up table of size $M \cdot 2^k$ and generate a number of recommendations using inexpensive multiplications over mod n^2. For example consider the case where Alice has 2048 ratings out of 102 400 available items in the system. It takes Bob only 610 milliseconds to create the look-up table and 16 milliseconds to generate a single recommendation. Thus, for a common recommender system that generates 10-100 recommendations, the run-time of the privacy-preserving algorithm is 0.16 to 1.6 *seconds*. These numbers can be improved even further in a real system as our proposal consists of operations that are highly parallelizable.

To the best our knowledge, our proposal is the first privacy-preserving content-based recommender systems. Therefore, we only provide experimental results from related works, which are based on collaborative filtering techniques that involve computing similarities among users and items. It takes 135 seconds in [11] and 75 seconds in [10] to generate recommendations for a recommender system with 10 000 users and 1000 items. In [5], it takes longer time, roughly 15 hours as stated by the authors, to generate recommendations. However, [6, 5] are based on the malicious security model and thus the authors use computationally more expensive cryptographic techniques like commitment schemes and zero-knowledge proofs [14].

4.2 Bandwidth

The three versions of the recommender systems we present in this paper are one round protocols: Alice sends her encrypted preferences and receives encrypted weighted sums. No further communication between Alice and Bob is required. Therefore, Alice sends M encryptions in either version and receives N in PPA and PPA-LUT and N_e encryptions in PPA-LUT/DP as in Table 2. For the experimental setting with 102 400 items, Alice should send $M \cdot 2n$ bits, which is equal to 512 KB. Bob sends back either 25 MB or 484 KB of data, depending on the choice of PPA-LUT and PPA-LUT/DP. Recall that these numbers are based on the worst-case scenario. In a real setting, Bob sends much less recommendations to Alice.

The bandwidth requirement of our proposals is significantly smaller than [10, 11, 5]. [11] proposes a method that requires Alice and Bob to transmit 8 KB and 1.8 GB of data, respectively. Similarly, Alice and Bob transmit 88 MB and 168 MB of data, respectively, in [10].

5. 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, and data packing. 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 numerical results show that privacy-preserving content-based recommender system is highly efficient even in the worst case scenario with more than 100 000 items, in which the retailer generates a single recommendation in 16 milliseconds. Within a realistic setting, where reasonable amount of recommendations are generated per customer, our proposal presents a very efficient method to generate private recommendations.

6. 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] T. Bianchi, A. Piva, and M. Barni. Composite signal representation for fast and storage-efficient processing of encrypted signals. *IEEE Transactions on Signal Processing*, 2009.

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

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

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] Z. Erkin, T. Veugen, T. Toft, and R. L. Lagendijk. Generating private recommendations efficiently using homomorphic encryption and data packing. *IEEE Transactions on Information Forensics and Security*, 2012. to appear.

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

[14] O. Goldreich. *Foundations of Cryptography II*. Cambridge University Press, 2004.

[15] 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.

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

[17] G. Linden, B. Smith, and J. York. Amazon.com recommendations: item-to-item collaborative filtering. *Internet Computing, IEEE*, 7(1):76 – 80, jan/feb 2003.

[18] 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.

[19] 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.

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

[21] 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.

[22] 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.

[23] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, WWW '01, pages 285–295, New York, NY, USA, 2001. ACM.

[24] 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.

[25] J. R. Troncoso-Pastoriza, S. Katzenbeisser, M. U. Celik, and A. N. Lemma. A secure multidimensional point inclusion protocol. In *ACM Workshop on Multimedia and Security*, pages 109–120, 2007.

[26] 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.