CONTENT-BASED RECOMMENDATIONS WITH APPROXIMATE INTEGER DIVISION

T. Veugen[*] *and Z. Erkin*

Cyber Security Group, Department of Intelligent Systems
Delft University of Technology
2628 CD Delft, The Netherlands

ABSTRACT

Recommender systems have become a vital part of e-commerce and online media applications, since they increased the profit by generating personalized recommendations to the customers. As one of the techniques to generate recommendations, content-based algorithms offer items or products that are most similar to those previously purchased or consumed. These algorithms rely on user-generated content to compute accurate recommendations. Collecting and storing such data, which is considered to be privacy-sensitive, creates serious privacy risks for the customers. A number of threats to mention are: 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 a cryptographic approach 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. Our proposal explores basic 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 division protocol which enables us to hide commercially sensitive similarity values, which was not the case in previous works.

***Index Terms*—** Recommender systems, privacy, secure multi-party computation, homomorphic encryption, secure division.

1. INTRODUCTION

Due to the increased use of the Internet, online services have exhibited phenomenal growth in the last decade. As a crucial component of e-commerce, customized services increase profits of the retailers by creating personalized profiles and using the information in such profiles for prediction algorithms, such as collaborative filtering techniques [1]. More precisely, a recommendation is generated for a particular customer by observing the characteristics of the previously purchased products [13]. However, as such systems heavily depend on personal data, which can be misused, transferred or sold to third parties, there are serious privacy concerns. Consequently, there have been studies on privacy-preserving recommender systems to address the challenge of providing a system, where customized services can be performed without harming the privacy of the customers.

There are a wide range of techniques for privacy protection, including data perturbation [2] and cryptography [9]. Polat and Du in [12] suggested hiding the personal data statistically. Shokri *et al.* presented a recommender system that is built on distributed aggregation of user profiles. McSherry and Mironov proposed a method using differential privacy, which has a trade-off between accuracy and privacy [10]. Atallah *et al*. presented a privacy-preserving collaborative forecasting and benchmarking mehtod to increase the reliability of local forecasts and data correlations using cryptographic techniques [3]. Canny presented 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 for recommender systems based on collaborative filtering, based on cryptographic techniques like homomorphic encryption and secure multi-party computation [7, 8].

In this paper, we present a privacy-preserving version of a content-based recommender system based on cryptography. Unlike previous cryptography-based works like [6], in our system we achieved to hide all privacy and commercially sensitive data. More precisely, we use a secure division protocol by Veugen [14] to additionally hide the commercially sensitive similarity measures matrix. The content-based recommender system by Erkin et al. [6] uses packing to reduce the number of encryptions, and to enable parallel computations on encrypted data. To avoid leakage of the similarity matrix, we modified Veugen's division protocol to be able to deal with packed data, leading to a solution that is both secure and efficient.

In our setting, we assume that Bob has the item similarity matrix $s_{(i,j)}$, $1 \leq i, j \leq L$, which are integers denoting the

[*]TNO Technical Sciences, P.O. Box 96800, 2509 JE The Hague, The Netherlands

similarity measure between item i and item j. Alice holds a preference vector consisting of M, $M \ll L$, ratings p_m for content item m, $1 \leq m \leq M$. Like in [6], we assume the items have been ordered such that Alice's ratings correspond with the first M items, which means that Bob knows the indices of the items (but not their ratings). Let $\mathcal{I}$ be the set of similar items, which contains N indices of items that are similar (have a similarity value above a certain threshold) to (almost) all M rated items of Alice. The protocol should output recommendations r_i to Bob for each $i \in \mathcal{I}$, where

$$r_i = \left(\sum_{m=1}^{M} p_m \cdot s_{(i,m)} \right) \div \left(\sum_{m=1}^{M} s_{(i,m)} \right) \qquad (1)$$

is the integer division of two summations. Bob is not allowed to learn Alice's preferences p_m, and Alice is not allowed to learn Bob's similarity measures $s_{(i,j)}$. Our contribution compared to the state-of-the-art is that the divisors $v_i = \sum_{m=1}^{M} s_{(i,m)}$ remain unknown to Alice, which is important as they contain information on the similarity matrix.

The rest of the paper is organized as follows. In Section 2 we describe our privacy-preserving content-based recommender system, and the cryptographic protocols. In Section 3 we show the correctness of the protocol, prove that it is secure, and indicate its performance. The final section summarizes the conclusions.

2. PRIVACY-PRESERVING RECOMMENDER SYSTEM

We assume that Bob, the service provider, who has the item similarity matrix, has generated a key pair of an additively homomorphic encryption scheme such as Paillier [11]. Alice, the user with its preference vector, is assumed to hold the public encryption key. An encrypted value is denoted by $[.]$. Encryptions in Paillier are always reduced modulo n^2, where n is a large composite number, but to increase clarity this reduction is not explicitly mentioned in our formulas.

2.1. Main approach

Alice would like to have the estimated ratings r_i, which are computed as in Equation 1, without leaking the preference vector p_m to Bob, and without leaking the similarity matrix $s_{(i,m)}$ to Alice. The set $\mathcal{I}$ of similar items is defined as $\{i_1, \ldots, i_N\}$. To this end, Alice and Bob perform the following steps.

1. Bob sends M packed encryptions to Alice $[s_{(i_1,m)} \mid s_{(i_2,m)} \mid \ldots \mid s_{(i_N,m)}]$, for $m = 1, \ldots M$.
2. Alice multiplies them with p_m (through exponentiations), and gets $[\omega_m] = [p_m \cdot s_{(i_1,m)} \mid p_m \cdot s_{(i_2,m)} \mid \ldots \mid p_m \cdot s_{(i_N,m)}]$, for $m = 1, \ldots M$.
3. Alice adds the M packed encryptions and gets $[\tilde{w}] = [w_{i_1} \mid w_{i_2} \mid \ldots \mid w_{i_N}]$, where the recommendation numerators w_i, $i \in \mathcal{I}$, equal $\sum_{m=1}^{M} p_m \cdot s_{(i,m)}$. Due to the additively homomorphic property, $[\tilde{w}] = \prod_{m=1}^{M} [\omega_m]$.
4. Bob computes the divisors $v_i = \sum_{m=1}^{M} s_{(i,m)}$, for each $i \in \mathcal{I}$.
5. Alice and Bob perform an "approximate division" protocol of packed values with a private divisor, as described below.
6. After the protocol, Alice has the estimated $[r_i]$ for all $i \in \mathcal{I}$, so we can run a decryption protocol, which is explained in Subsection 2.3.

A packing of integers is simply a bitwise concatenation of the integers to form one large integer. As an example, Equation 2 shows how the integers ρ_i, each consisting of W bits, can be packed into one integer $\tilde{\rho}$. Because the similarity values are packed, the multiplication with the preference value can be performed in step 2 by only one exponentiation. Here we use the additively homomorphic property of $[.]$ to get $[s_{(i,m)}]^{p_m} = [p_m \cdot s_{(i,m)}]$.

2.2. Approximate division

Suppose Alice has the encrypted packed numerators $[\tilde{w}] = [w_{i_1} \mid w_{i_2} \mid \ldots \mid w_{i_N}]$. Bob has the decryption key, and has the denominators (divisors) $v_{i_1} \ldots v_{i_N}$. Let S denote the maximal number of bits of a similarity value $s_{(i,m)}$, and let P be the maximal number of bits of a preference value p_m, then the maximal number of bits W of a numerator w_i is $P + S + \lceil \log_2 M \rceil$. We assume both Alice and Bob know the integer W, which is a common assumption in secure multiparty computations: the inputs are private, but their lengths (maximal number of bits) are known.

To get the encrypted ratings $[r_i]$, where $r_i = w_i \div v_i$, Alice and Bob run the following protocol, which is based on ideas from Veugen [14]. For simplicity, we assumed that $\mathcal{I} = \{1, \ldots, N\}$.

1. Bob encrypts the v_i, $1 \leq i \leq N$, and sends them to Alice. Bob also sends the lenghts (number of bits) $V_i = \lceil \log_2 v_i \rceil$ of each v_i to Alice. We might also assume they have the same length, but that's not necessary.
2. Alice generates $2N$ random numbers ρ_i^d and ρ_i^m, $1 \leq i \leq N$. The ρ_i^m contain V_i bits, and the ρ_i^d contain $W - V_i$ bits. To guarantee the packed number will be sufficiently blinded, the random number ρ_1^d has to be somewhat larger, namely $W - V_1 + \kappa$ bits, where κ is the statistical security parameter.
3. Alice computes the encrypted random numbers $[\rho_i] = [v_i \cdot \rho_i^d + \rho_i^m] = [v_i]^{\rho_i^d} \cdot [\rho_i^m]$ for each i, $1 \leq i \leq N$,

and packs them together (starting at the right with item number N) to $[\tilde{\rho}]$. The superscripts d and m refer to the integer division result d and the modular remainder m (see also Subsection 3.1).

4. Alice computes $[\tilde{z}] = [\tilde{w}] \cdot [\tilde{\rho}]$, and sends it to Bob. The number $\tilde{\rho}$ can be considered as one large random number that is used to additively blind $\tilde{w}$ (see also Subsection 3.2).

5. Bob decrypts $\tilde{z}$, which consists of N departments of bit size W. Only the first (left-most) department is somewhat larger and consists of $W + \kappa$ bits. Bob unpacks $\tilde{z} = z_1 \mid z_2 \mid \ldots \mid z_N$ into N integers z_i, computes $z_i \div v_i$, and sends these N integer division results encrypted to Alice.

6. Alice computes the N encryptions $[\hat{r}_i] = [(z_i \div v_i) - \rho_i^d] = [z_i \div v_i] \cdot [\rho_i^d]^{-1}$, which contain good approximations of the ratings r_i.

7. Alice and Bob run one (packed) decryption protocol to get $\hat{r}_i$, which are good approximations of the r_i. In fact, $r_i \leq \hat{r}_i \leq r_i + 2$, which is shown in Subsection 3.1.

The plain text size of the encryptions scheme should be at least $NW + \kappa + 1$ bits, to avoid carry-overs.

2.3. Decryption

After the "Approximate division" protocol, Alice has obtained N encrypted estimated recommendations $[\hat{r}_i]$, $i \in \mathcal{I}$, and wants to have them decrypted. The length of each recommendation, which is the maximal number of bits of a preference value, equals P. To this end, the following steps are needed, in which packing is used for efficiency reasons.

1. Alice packs the N encryptions into one $[\tilde{r}] = [\hat{r}_{i_1} \mid \hat{r}_{i_2} \mid \ldots \mid \hat{r}_{i_N}]$. Due to the additively homomorphic property, this can be performed in $(N-1)(P+1)$ multiplications: $[\tilde{r}] = (\ldots([\hat{r}_{i_1}]^{2^P} \cdot [\hat{r}_{i_2}])^{2^P} \ldots)^{2^P} \cdot [\hat{r}_{i_N}]$.

2. Alice chooses a random number ϱ of at least $PN + \kappa$ bits (alternatively, Alice chooses a random number of full plain text size), encrypts it, and adds it to $\tilde{r}$ to additively blind that value.

3. Alice sends the encrypted blinded value $[\tilde{r} + \varrho]$ to Bob, who decrypts it, and sends the result back.

4. Alice subtracts ϱ, and unpacks the result $\tilde{r}$ to obtain the decrypted $\hat{r}_i$, $i \in \mathcal{I}$.

The packed decryption protocol costs only one decryption by Bob, and works as long as the plain text size of the crypto system is at least PN bits. Because the random number ϱ has κ more bits than $\hat{r}$, Bob will not learn $\hat{r}$.

3. ANALYSIS

We show that our protocol is secure and correctly computes the required output. The most important part is the approximate division subprotocol as described in Subsection 2.2, because the rest is similar to the protocol of [6]. Finally, we analyse the performance.

3.1. Correctness

The packed integer $\hat{w}$ is actually a concatenation of N numbers w_i. By adding a large random integer $\hat{\rho}$ to $\hat{w}$, a small random integer ρ_i is added to each compartment. Each random number ρ_i uniquely corresponds with a random pair (ρ_i^d, ρ_i^m) such that $\rho_i = \rho_i^d \cdot v_i + \rho_i^m$ and $0 \leq \rho_i^m < v_i$. The first random number ρ_i^d is actually the result of the integer division $\rho_i \div v_i$, and the second random number ρ_i^m is the remainder of that division. In the protocol, the random numbers ρ_i^m contain V_i bits, and $v_i < 2^{V_i}$, so the random number ρ_i^m is not uniformly drawn from $[0, v_i)$, but from $[0, 2^{V_i})$. This might cause a slight inaccuracy in $\hat{r}_i$ when ρ_i^d is subtracted in step 6, because sometimes $\rho_i = (\rho_i^d + 1) \cdot v_i + (\rho_i^m \bmod v_i)$, namely when $\rho_i^m \geq v_i$. To actually affect r_i, the numerator value w_i also has to satisfy $w_i \bmod v_i = v_i - 1$.

When the random number ρ_i is added to the private value w_i, the result should fit into the i^{th} compartment, which consists of W bits. We allow for a small one-bit carry-over to the next compartment. Because $w_i + \rho_i = w_i + \rho_i^d \cdot v_i + \rho_i^m \leq (2^W - 1) + (2^{V_i} - 1) \cdot (2^{W - V_i} - 1) + (2^{V_i} - 1) < 2^{W+1} - 1$, the carry-over can never exceed one bit. This carry-over might increase the random value ρ_{i-1}^m by at most one.

To show that $\hat{r}_i = (z_i \div v_i) - \rho_i^d$ is a good approximation of r_i we refer to the analysis of Veugen [14]. The main idea is that $(w_i + \rho_i) \div v_i = (w_i \div v_i) + (\rho_i \div v_i) + c$, where c is the binary comparison result of $(w_i + \rho_i) \bmod v_i$ and $\rho_i \bmod v_i$. Furthermore, z_i is often equal to $w_i + \rho_i$, except when there has been a carry-over from the previous compartment in which case $z_i = w_i + \rho_i + 1$. Finally, $\rho_i \div v_i$ is usually equal to ρ_i^d, except when $\rho_i^m \geq v_i$ in which case $\rho_i^d = (\rho_i \div v_i) + 1$. So in the worst case, $\rho_i^m \geq v_i$, $z_i = w_i + \rho_i + 1$, $w_i \bmod v_i = v_i - 1$, and $c = 1$, and then $\hat{r}_i = (z_i \div v_i) - \rho_i^d = (w_i + \rho_i + 1) \div v_i - \rho_i^d = (w_i \div v_i) + (\rho_i + 1) \div v_i + 1 - \rho_i^d = r_i + 2$.

We conclude that the outputs $\hat{r}_i$ of the "Approximate division" protocol satisfy $r_i \leq \hat{r}_i \leq r_i + 2$. Therefore, by using $\hat{r}_i - 1$ as an estimate, the absolute error is bounded by one.

3.2. Security

To guarantee that $\tilde{w}$ is statistically hidden in $\tilde{z}$, we have to show that $\tilde{\rho}$ is actually a random number of at least $NW + \kappa$ bits. By definition,

$$\tilde{\rho} = \sum_{i=1}^{N} \rho_i \cdot 2^{W(i-1)}, \tag{2}$$

Part	Multiplications	Encr.	Decr.	Bandwidth	Rounds
Main approach	$M-1+\frac{3}{2}MP$	M	0	M	$\frac{1}{2}$
Approximate division	$\frac{3}{2}N(W-V)+2+$ $(N-1)(W-V+1)+N$	$3N+1$	1	$2N+1$	$\frac{1}{2}$
Decryption	$(N+1)(P+1)+1$	1	1	1	1

Table 1. Computational and communication complexity

where we assumed for simplicity, as in Subsection 2.2, that $\mathcal{I} = \{1, \ldots, N\}$. Furthermore, $\rho_i = \rho_i^d \cdot v_i + \rho_i^m$, and the numbers ρ_i^d and ρ_i^m are uniform random numbers of $W - V_i$ and V_i bits respectively. Since $v_i < 2^{V_i}$, the first W bits of ρ_i can be considered as random bits. Consequently, $\tilde{\rho}$ can be considered as the addition of a uniform random number of NW bits, and a smaller packed number with compartments $\rho_i \div 2^W$. Since the first random number ρ_1^d contains κ additional bits, we can conclude that $\tilde{w}$ is indeed statistically secure within $\tilde{z}$.

3.3. Complexity

For our complexity analysis, we count the number of operations on encrypted numbers. In the first step of the main approach, Bob has to perform M encryptions. During the second step, Alice has to do M exponentiations with exponents consisting of P bits. For the third step, Alice has to compute $M - 1$ multiplications.

In the approximate division protocol, Bob has to encrypt N integers. To compute the encrypted random numbers $[\rho_i]$, Alice has to perform N exponentiations with exponents of length $W - V_i$. The random numbers ρ_i^m can be packed first in the plain domain, and consequently simultaneously added through one encryption and one multiplication. The packing takes a further $(N-1)(W-V+1)$ multiplications. The addition requires Alice to perform only one multiplication. Consequently, Bob performs one decryption, and N encryptions. Finally, Alice needs N encryptions and N multiplications to subtract the ρ_i^d again.

During the decryption protocol, Alice needs $(N-1)(P+1)$ multiplications to pack the N recommendations. She needs one encryption and one multiplication to blind it. At the end, Bob performs one decryption.

We assume that an exponentiation with an exponent of size P requires $\frac{3}{2}P$ multiplications. Table 1 shows the total computational complexity, where Bob's share consists of $M + 2N$ encryptions, and 2 decryptions.

In terms of bandwidth, our protocol requires an additional exchange of N encrypted values during the approximate division protocol. We summarize the overall complexity in Table 1, where the values for the bandwidth represent the number of encrypted values transmitted. With respect to the number of rounds, our protocol requires two rounds instead of one, due to the approximate division protocol. The sending of encryptions by Bob in the first step of both the main approach and approximate division protocol can be combined into one round, as summarized in Table 1. Compared to the solution of [6], it is clear that we introduce extra computation and bandwidth requirements for the approximate division protocol, as expected, which roughly doubles the complexity.

For a clear view on performance, we also provide an estimate. Typical values for our system parameters are $P = S = 4$ and $M = N = 64$ [6], in which case $V = P + \log_2 M = 10$ and $W = P + S + \log_2 M = 14$. With these parameters, all values could be packed in one encryption of 1024 bits. So in a regular setting, the division protocol takes 639 multiplications, whereas the other two parts take 447 and 326. The number of encryptions needed in the division protocol is 193, compared to 64 and 1 in the other parts. When Paillier is used, a value can be encrypted by only one multiplication by choosing $g = n + 1$ [11]. Given that the division protocol also needs one more decryption, we predict a doubling of the computational complexity due to the division protocol.

In terms of run-time, the system in [6] required seconds to run with the same parameters. Therefore, the overall execution time of our protocol is also in the same order. Clearly, we achieved a higher level of security at the expense of doubling the computational and communication costs.

4. CONCLUSIONS

We developed a system for recommending items in a privacy-preserving way by using a content-based item similarity matrix. Compared to previous solutions, we avoided the leakage of the divisors v_i, which contain information about the commercially sensitive item similarities. The costs of introducing a secure division protocol led to a doubling of the computational and communication complexity, and a slight loss in recommendation accuracy. However, the system neither relies on trusted third parties, nor requires interaction with peer users. In addition, our proposal offers an efficient *and* much more secure solution for this class of recommender systems.

Acknowledgements

This publication was supported by the Dutch national program COMMIT.

5. 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.*, 17:734–749, June 2005.

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

[3] M. Atallah, M. Bykova, J. Li, K. Frikken, and M. Topkara. Private collaborative forecasting and benchmarking. In *WPES '04: Proceedings of the 2004 ACM workshop on Privacy in the electronic society*, pages 103–114, New York, NY, USA, 2004. ACM.

[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] Z. Erkin, M. Beye, T. Veugen, and R. Lagendijk. Privacy-preserving content-based recommender system. In *The 14th ACM Workshop on Multimedia and Security*, pages 77–84, Coventry, UK, 2012. ACM.

[7] 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.

[8] Z. Erkin, T. Veugen, T. Toft, and R. Lagendijk. Generating private recommendations efficiently using homomorphic encryption and data packing. *IEEE Transactions on Information Forensics and Security*, 7(3):1053–1066, 2012.

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

[10] 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.

[11] 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.

[12] 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.

[13] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *WWW10 Conference*, pages 285– 295, Hong Kong, 2001.

[14] T. Veugen. Encrypted integer division and secure comparison. *International Journal of Applied Cryptography*, 3(2), 2014.