INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

Homomorphic Encryption Based Privacy Protection for Personalised Web Search

Krishan Kumar[1], Mukesh Kumar Gupta[2], Nishant Saxena[2]*, Vivek Jaglan[3]

1 Department of Computer Science and Engineering, Suresh Gyan Vihar University, Jaipur, India
2 Department of Electrical Engineering, Suresh Gyan Vihar University, Jaipur, India
3 Department of Computer Science and Engineering,, DPG Institute of Technology and Management, Gurgaon, India

OPEN ACCESS

Received: 06.01.2022

Accepted: 28.01.2022

Published: 25.02.2022

Citation: Kumar K, Kumar Gupta M, Saxena N, Jaglan V (2022) Homomorphic Encryption Based Privacy Protection for Personalised Web Search. Indian Journal of Science and Technology 15(8): 318-325. https://doi.org/10.17485/IJST/v15i8.44

* **Corresponding author**.

nishant.saxena@mygyanvihar.com

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Objective: Privacy of user information in web search applications is traded off with quality of search results generated for a query by web search engine. In this article, we have developed a novel model utilizing Homomorphic Encryption for privacy protection of the user without compromising the quality of search results and response time. **Methods:** Response time is calculated using the GreedyDP and GreedyIL techniques, which is critical since encryption is usually followed by complicated calculations. Using the Homomorphic Encryption (HE) technique the user request is encrypted, rendering it unreadable or interpretable by eavesdroppers. Although the server will be unable to decode the requests, it will be possible to process those using algorithms and computations. The suggested model was utilized to evaluate the reaction time performance of four distinct current HE techniques. **Findings**: From the proposed model, it is inferred that, the performance of Gentry HE is superior to others since it takes 6% less time than its nearest competitor Paillier. Implementations show that the developed model, query encryption, does not create response delays and so supports the framework. **Novelty:** This research proposes a new PWS model with HE to increase data security and privacy in online search applications. The suggested study uses the GreedyDP and GreedyIL new methodologies.

Keywords: Personalised Web Search; user profile; Homomorphic Encryption; GreedyDP; GreedyIL

1 Introduction

Personalization of the online search engine is necessary in order to extract information on the web in accordance with the preferences of the users while offering access to the data and expertise available on the World Wide Web[(1)]. There has been an ever-increasing amount of data available on the WWW, making it challenging for the search engine to provide users with the information they are looking for (PWS).

In earlier researches, user profile has been constructed based on user interaction with a search engine. The developers created and implemented a Google wrapper around the Google search engine, which is used to log the queries, search results, and clicks on a user basis. Then the information is used to construct user profiles, users register, and log in to store cookies on their local machines with user IDs(2). The technique is developed for one particular search engine only. It seems to be a limitation for the existence of personalised search apps that people don't want to share their personal information with search engines. Privacy in PWS is a tough study subject because of this. User customised privacy-preserving search (UPS)(3) can keep hold of privacy protection need; but this is faulted with the prospects of eavesdropping when a generalised profile is forwarded to the server. Privacy-enhancing PWS which has been involved in the creation of the hierarchical user profile, primarily based on the personal interests of the users because most of the users do not want to share private information with the search engine. This technique does not support the concept of run time profiling which is considered a big drawback(4,5). A generalized offline user profile is created only once, and it is used for every query without making any changes in it. WordNet is publicly accessible data covering the entire topic domain of human knowledge(6,7). The major drawbacks of this technique are, they didn't take benefits of high computation speed and search algorithms at server side and use more bandwidth by sending user profile with each query.

At the same time, the web server is susceptible to assaults such as URL manipulations, which compromise the privacy of the user. To avoid this kind of assault, it would be sensible to utilise Homomorphism Encryption (HE), which combines data encryption with privacy protection, to create a new method(5,8,9). That's why attempts are made to incorporate the HE request for privacy protection into tailored online searches.

2 Proposed Model for Privacy Protection

The system proposed by us has addressed all the important issues discussed in the previous sections. Contrary to the run time profiling which is not supported by current profile-based personalized web search systems and user profiles being generated offline only once provides poor search quality, we have implemented the runtime profiling technique with the help of the GreedyDP algorithm and greedy IL algorithm. This online profiler is implemented on the client machine itself. Figure 1 explains how the proposed system works. The user profile and encrypted query are submitted to the server for desired results. This will increase computation cost and complexity at the server end, but it is worth protecting the privacy of the user.

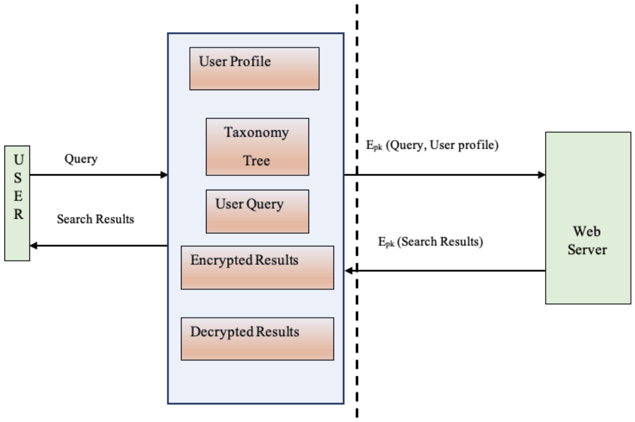

Fig 1. Proposed architecture

Users submit the query to the proxy at the client machine; the online runtime user profile is generated at the client-side based on the query and other information on the client machine. After encrypting the query and online profile with privacy requirements, it is submitted to the web server for personalized search. Search results are prepared by the server from cipher text by fully Homomorphic Encryption Technique. After this, the search results are sent to the client through proxy. Proxy decrypts the results, re-ranks them with the help of a complete user profile, and provides them to the user. Still, the adversary at the honest but curious server can identify the user with an IP address and pair of Public-Private keys.

3 Homomorphic Encryption and Privacy Requirements

PWS deals with user data and has created very serious privacy concerns. For example, personalized web search is dependent on sensitive private information which can be stolen and abused if communication is executed on remote servers. To handle this problem information processing and cryptography can be used to protect the privacy of the user. Service providers will not be able to access the content of encrypted data because it will be processed in an encrypted format. Homomorphic cryptosystems and multi-party computations (MPC) can be used to process encrypted data. A framework that comprises Homomorphic encryption and data fragmentation to improve the secure distribution of information in a cloud computing environment has also been developed(10). The challenges of security, integrity, authentication, and confidentiality have been addressed successfully with the help of Homomorphic Encryption.

An idea and mechanism for preserving the user location using Homomorphic Encryption have been presented, where the user can encrypt data with the public key and transfer it to the cloud(11). The cloud server can perform calculations and computations on the cipher text, but it cannot decrypt the results of the computations as it does not have the private key. The server responds the cipher text back to the user who can decrypt the results to get the desired results. The idea of protecting the location of the user is restored as the client can hide the location-based query from the server. So Homomorphic Encryption gives a new dimension to cloud storage and security. We can secure the plane text from being exposed using Homomorphic Encryption(12). With the advent of a Fully Homomorphic Cryptosystem, the data has become semantically secured.

A secure, efficient, and privacy-preserving technique to compute Linkage Disequilibrium measures over the genome data(13). The Approach improves the computational complexity and storage. The authors worked on genome data to check performance results using Pearson's correlation coefficient, Goodness of fit test, LD coefficient.

It is evident that a Fully Homomorphic Encryption can improve performance and security based on weak assumptions. A levelled fully homomorphic encryption has also been proposed for evaluating arbitrary polynomial-size circuit of a priori bounded depth without Gentry's bootstrapping procedure(14). The authors worked on Fully Homomorphic Encryption based on learning with error or Ring learning with error problems that have 2^{λ} security against the known attacks. Table 1 shows the comparative analysis of various Homomorphic Techniques.

Table 1. Comparative analysis of various Homomorphic Techniques

Type	Type of Homomorphic encryption Efficiency	Efficiency and Security
Unpadded RSA	Multiplicative	Deterministic and can be cracked using chosen plain text attack
ElGamal	Multiplicative	Probabilistic chosen-cipher text has the chosen-cipher text attack
Goldwasser-Micali	XOR/ Additive -encrypts a single bit at a time.	Not an efficient cryptosystem, as cipher texts may be several hundred times larger than the initial plaintext.
Paillier	Additive	Probabilistic, efficient, and simple. It involves only one multiplication for each homomorphic addition and one exponentiation for each homomorphic multiplication
Gentry	Fully	Efficient as it computes an arbitrary number of operations on encrypted data. However practically not possible.

It is obvious that web search engine is not supposed to remember about the user's private sensitive information, due to encryption of inputs, intermediate and output results are encrypted. At the same time, users also do not have any information about the algorithms at web search engine end users for finding search results. This informal information is not sufficient rather all cryptographic protocols, privacy reserving data communication should be accompanied by security proof and protocols(15).

The level of security generally depends on the assumptions about the intentions and capabilities of adversaries. It is assumed that the attacker cannot break hard mathematical problems because of its limited computation power. It is also assumed that keys are generated as well as certified by a trusted third party if needed.

The web search engine can be considered as a curious but honest adversary who is participating in the process. Web search engine follows all the protocols correctly to provide desired results to the user but being curious it collects all input, intermediate, and the output to learn about the user. Web search engine's intentions as malicious adversaries can influence the computation of search results that users obtain are possibly incorrect. Web search engines can easily influence the output by changing user's values ¦q(1)¦, ¦q(2)¦and using some fictive values and may lead to wrong output. It can encrypt the values by using the user's public key SK. A malicious outsider adversary which is not part of the computation can change the search results by replacing the user's encrypted input with encrypted bogus values. In the call the communication between user and web search engine is not secured with cryptographic techniques. Even though outsider adversaries are the truth of real-world application, our focus lies on adversaries participating in the process.

Privacy against malicious adversaries is hard to achieve and not much work is reported in the literature. The transformation needs proof against all intermediate computation using zero-knowledge proofs and commitment schemes although these can be computationally demanding. One more aspect of the attacker model is collusion, if more than one user is using the same private-public key for communication with a web search engine then they can conceal one another's sensitive information. The final aspect of privacy is the algorithm that the web search engine is using[16]. Users should not be allowed to send arbitrary input information to the web search engine because it can infer critical information about the algorithm with the help of a sensitivity attack. We can keep the algorithms at web search engines secret by not providing direct input-output relation. It appears worth noting that in privacy-protected solutions the web search engine will be able to identify the user with IP address and unique public-private key pair.

4 System Model

Users are benefited by providing private information; but their privacy is at stake as the service providers cannot be trusted to preserve the fidelity of the personal information of the user; Therefore, this information is always on the verge of leakage or stolen. In personalized web search, two parties are involved - the user who owns sensitive/private information query q(i) and the web search engine which has all the algorithms f(.) to provide desired results to the user.

The user is interested in web search results f(q(i)) but does not want to reveal the query q(i) to the Web search engine because of privacy issues. At the same time, Web search engine does not or cannot reveal its algorithm f(.) to the user for commercial or computational reasons. Web search engines have an algorithm that can process two queries q(1) & q(2) to get a binary classification result C as shown in Figure 2.

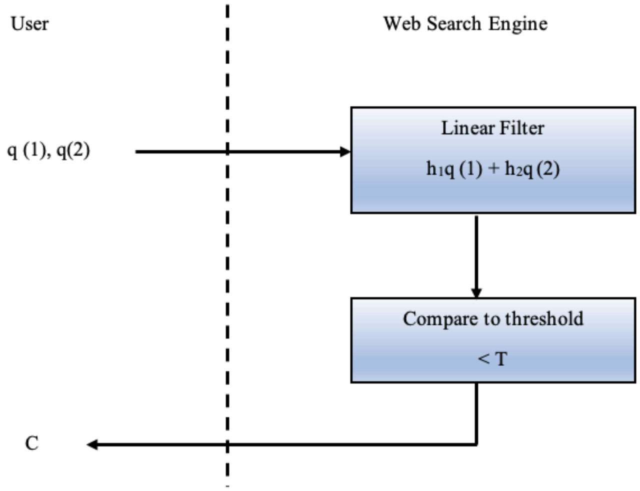

Fig 2. Block Diagram for data Communication

$$C = \begin{cases} 0, \text{ if } h1q(1) + h2q(2) < T \\ 1, \text{ otherwise} \end{cases} \tag{1}$$

Here, the queries q(1), q(2), and classification $C \in \{0,1\}$ are private to the user; so these are kept secret from the Web Search Engine[17]. The linear weights h1, h2, and T (Threshold) are private to Web search engines. We can say Web Search engine should be unaware of not only the queries q(i) but also the search results C. The assumptions made in the contemplated work are: -

Assumption 1: The Web Search Engine calculates f(q(i)) precisely without disrupting the value of C. Web search engine's attacks to obtain input queries, intermediate results h1q(1) +h2q(2), or C, will make it honest-but-curious or it can be considered a semi-honest party

Assumption 2: In this case, the User should not make many search query requests to the search engine, because it may help Web Search Engine in calculating the values of h1,h2, and T by implementing trial and error technique which is considered as sensitivity attack and these attacks are untreatable in cryptography.

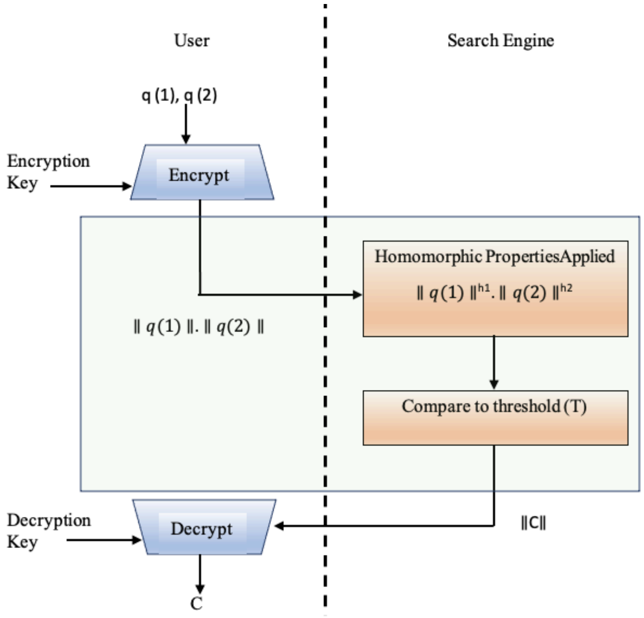

Fig 3. Privacy Protected version of data Communication

5 Query Encryption

Query encryption is divided in two steps. First step is Encrypted samples and second step is homomorphic cryptosystems.

5.1 Encrypted Samples

Now let us consider a situation where user submits web search engine, the query q(i) in encrypted form. User encrypts q(i), using a public key cryptosystem with the additively homomorphic property(8–11). The key properties of additive homomorphic encryption are:

$$D_{sk}\left(E_{pk}\left(m_1\right).E_{pk}\left(m_2\right)\right) = m_1 + m_2 \tag{2}$$

$$D_{sk}\left(E_{pk}\left(m\right)w\right) = w.m$$

User sends only its public key and ciphertext to the web search engine; subsequent steps are explained in the Figure 3. For better understanding, ciphertext notation for encrypted query using public key pk.

q(i) encrypted with public key

$$E_{pk}\left(q\left(i\right)\right) = \left(q(i)\right| \tag{3}$$

With Additively Homomorphic Public Key Encryption we can rewrite equation (1) as given below.

$E_{pk}(h_1\left(q\left(1\right)\right) + h_2(q\left(2\right)) = (h_1\left(q\left(1\right)\right) + h_2(q\left(2\right)|$
$= (h_1\left(q\left(1\right)\right).h_2(q\left(2\right)|$

$$= (q\left(1\right)||h_1|(q\left(2\right)|h_2 \; mod \; n \tag{4}$$

It appears from equation (4) that the web search engine does not need the secret decryption key SK from the user to compute the encrypted result of linear combination h1q(1) + h2q(2) from the ciphertext values ¦q(1) ¦and ¦q(2) ¦. The web search engine does not need user involvement in computing search results ¦h1q(1) ¦+¦h2q(2) ¦after submitting the query. The search results computed by the search engine are still encrypted and can be decrypted by the user with SK.

The web search engine can compare the encrypted result ¦h1q(1) + h2q(2) ¦with the plaintext threshold T with the help of the user. Without the help of the user, the web search engine cannot calculate the result by itself. Otherwise, the web search engine will easily decrypt most of the ciphertext by binary search. Arithmetic Comparison Protocol (ACP) and MPC Using Garbled Circuits are used to calculate results. Web search engine holds the encrypted results ¦C¦after completion of the interactive protocol. Web search engine submits ¦C¦to the user, which can use the result after decryption with its secret key SK.

The complexity of communication increases when a public-key cryptosystem is used as compared to plaintext. This is because of data expansion, computation overhead, and interactive protocols for comparing.

5.2 Homomorphic Cryptosystems

Operations like linear filters, signal transformation, and correlation evaluations are the inner products of two discrete-time signals; we can also consider them as arrays of values x(i) and y(i). The inner product can be defined for the M sample –

$$I =< x(.), y(.) \geq \left(\begin{matrix} x_1 & x_2 & \cdots & x_m \end{matrix} \right] \begin{bmatrix} y_1 \\ y_2 \\ y_m \end{bmatrix} = \sum_{i=1}^{m} x(i)\, y(i)$$

Where,

X(i) is individually encrypted

Y(i) is in plain text.

Using notation (3) and by implementing the additive-homomorphic property of the Paillier public-key cryptosystem equation (5) can be directly operated on encrypted signal samples ¦x(i)¦

$\sum_{pk}(I) = \sum_{pk}(\sum_{i=1}^{m} x(i)\, y(i))$

$= \prod_{i=0}^{m} \sum_{pk} x(i)\, y(i)$

$$= \prod_{i=0}^{m} (x(i)|\, y(i) \tag{6}$$

This is a generalized expression for the M sample. Equation (6) states that linear operations can be implemented on encrypted signals without interactive protocols between parties.

Equation (5) can be rewritten as given below if x(i) & y(i) are both encrypted.

$$E_{pk}(x(i)\,.y(i)) = E_{pk}(x(i)).E_{pk}(y(i))$$

Here, (.) produces ¦x(i).y(i) ¦which is additively homomorphic cryptosystem and not the usual modular multiplication. As per equation (7) it possesses multiplicative homomorphic property. A cryptosystem that has additive, as well as multiplicative homomorphic properties, is known as a Fully Homomorphic Cryptosystem. It is a rather inefficient means to implement fully homomorphic encryption (FHE) and a secure two-party multiplication protocol can be used for testing purposes.

If message x(i) & y(i) contain M samples then, we can calculate Squared Error Distance D which can be defined as-

$D = (x(.) - y(.)|_2$

$= \sum_{i=1}^{m} (x(i) - y(i))^2$

$$= \sum_{i=1}^{m} x(i)^2 + \sum_{i=1}^{m} y(i)^2 - 2\sum_{i=1}^{m} (x(i)\,.y(i))^2 \tag{8}$$

Considering the case that x(i) is available as cipher text ¦x(i)¦

$E_{pk}(D) = E_{pk}\left(\sum_{i=1}^{m} (x(i) - y(i))^2\right)$

$= \left(\sum_{i=1}^{m} x(i)^2 + \sum_{i=1}^{m} y(i)^2 - 2\sum_{i=1}^{m} (x(i)\,.y(i))^2\right|$

$$= \prod_{i=1}^{m} (x(i)|^2 . \prod_{i=1}^{m} (x(i)|\,.y(i)\,.\, \prod_{i=1}^{m} y(i) \tag{9}$$

Y(i)2 requires the encryption which can be computed using the public key.

Table 2. Hardware and Software Requirements

Hardware	CPU	Intel Core i3 CPU @1.80 GHz
	RAM	6 GB
	External Storage	1 TB
Software	Internet	Gigabit Ethernet
	Operating System	Windows 10 - 64 bit
	Java	Java Version 11
	Middleware	Eclipse IDE

6 Results and discussion

The developed model is tested by implementing existing algorithms for performance comparison. GreedyDP and GreedyIL algorithms are implemented in Java programming for creating user profiles. Tomcat Server and PostgreSQL database is used at the backend. Hardware and software configurations used are mentioned in Table 2.

As evident from Figure 4, the response time increases gradually with increase in seed profile size in case of GreedyDP algorithm while for GreedyIL, initially when seed size is small the increase is gradual but at later stages when seed size increases to 5, 6 the changes are exponential in nature. GreedyDP is taking lesser time than GreedyIL for responding to queries, it is observed that it takes 50 % time only for higher seed size of 5,6 and Therefore, GreedyDP is providing better results as compared to GreedyIL in terms of response time.

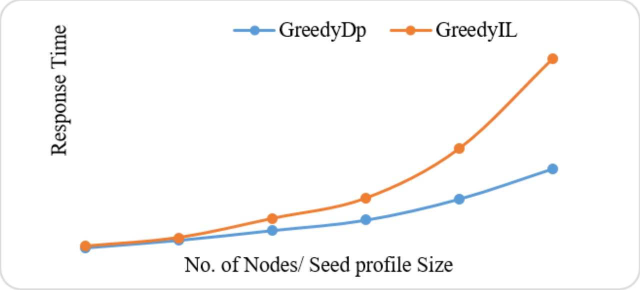

Fig 4. Number of nodes vs. response time

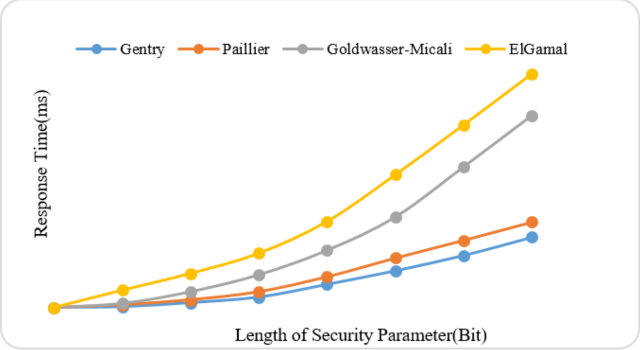

Fig 5. Time cost for the cipher text comparison

The programming part was completed in JAVA and recorded the average time of cipher text comparison phase in the ElGamal, Goldwasser-medically, Paillier, and Gentry techniques. In Figure 5, it can be observed that response time increases gradually with increase in length of security parameter and the proposed Gentry HE is showing minimum response time. Therefore, Gentry HE is performing better as compared to other techniques in terms of response time. The response time of Gentry is 6 % less than Paillier and it takes almost one third time of other algorithms.

7 Conclusions

This study has presented a new model for improving user privacy in personalized web search, by which the users can hide sensitive information related with customized privacy it is not required to send information to the server with the help of runtime profiling at the client machine.

User queries and customized profiles are sent to the server in an encrypted format using Homomorphic Encryption and therefore, cannot be accessed by eavesdroppers.

The server is also sending results in encrypted format which can be decrypted by the user only at the client machine.

The implementation of existing Homomorphic Encryption techniques on the developed model has revealed that response time increases with increase in length of security parameter but, the performance of Gentry HE is better than others as it takes 6% lesser time than its nearest competitor Paillier.

It is further concluded that, GreedyDP algorithm performs better than GreedyIL in terms of response time for creating user profiles and that GreedyDP takes 50 %-time only as compared to GreedyIL for higher seed size of 5,6. The developed model i.e encryption of queries does not cause any delay in responses as is evident from implementations and therefore supports the framework.

References

1) Grida M, Fayed L, Hassan M. User Profile: Theoretical Background. *International Journal of Engineering Trends and Technology*. 2020;68(8):10–17. Available from: https://dx.doi.org/10.14445/22315381/ijett-v68i8p203s.
2) Jing M, Zhicheng D, Ji-Rong W. FedPS: A Privacy Protection Enhanced Personalized Search Framework. In: Proceedings of the Web Conference 2021. Association for Computing Machinery. 2021;p. 3757–3766. Available from: https://doi.org/10.1145/3442381.
3) Yao J, Dou Z, Wen JR. FedPS: A Privacy Protection Enhanced Personalized Search Framework. *Proceedings of the Web Conference 2021*. 2021. doi:10.1145/3442381.3449936.
4) Wu Z, Shen S, Li H, Zhou H, Zou D. A comprehensive study to the protection of digital library readers' privacy under an untrusted network environment. *Library Hi Tech*. 2021. Available from: https://dx.doi.org/10.1108/lht-07-2021-0239.
5) Kumar K, Gupta MK, Jaglan V. Privacy Protection in Personalized Web Search Using Software Applications – Tools and Plug-Ins. *SSRN Electronic Journal*. 2021. Available from: https://dx.doi.org/10.2139/ssrn.3884638.
6) Wu Z, Lu C, Zhao Y, Xie J, Zou D, Su X. The Protection of User Preference Privacy in Personalized Information Retrieval: Challenges and Overviews. *Libri*. 2021;71(3):227–237. Available from: https://dx.doi.org/10.1515/libri-2019-0140.
7) Wu Z, Li R, Zhou Z, Guo J, Jiang J, Su X. A user sensitive subject protection approach for book search service. *Journal of the Association for Information Science and Technology*. 2020;71(2):183–195. Available from: https://dx.doi.org/10.1002/asi.24227.
8) Wu Z, Wang R, Li Q, Lian X, Xu G, Chen E, et al. A Location Privacy-Preserving System Based on Query Range Cover-Up or Location-Based Services. *IEEE Transactions on Vehicular Technology*. 2020;69(5):5244–5254. Available from: https://dx.doi.org/10.1109/tvt.2020.2981633.
9) Beigi G, Guo R, Nou A, Zhang Y, Liu Y. Protecting User Privacy : An Approach for Untraceable Web Browsing History and Unambiguous User Profiles. In: and others, editor. WSDM '19: Proceedings of the Twelfth ACM International Conference on Web Search and Data. 2019;p. 213–221. Available from: https://doi.org/10.1145/3289600.3291026.
10) Wu Z, Xie J, Lian X, Pan J. A privacy protection approach for XML-based archives management in a cloud environment. *The Electronic Library*. 2019;37(6):970–983. Available from: https://dx.doi.org/10.1108/el-05-2019-0127.
11) Shuqi L, Zhicheng D, Chenyan X, Xiaojie W, Ji RW. Knowledge Enhanced Personalized Search. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. Association for Computing Machinery. 2020;p. 709–718. Available from: https://doi.org/10.1145/3397271.3401089.
12) Wu Q, He K, Chen X. Personalized Federated Learning for Intelligent IoT Applications: A Cloud-Edge Based Framework. *IEEE Open Journal of the Computer Society*. 2020;1:35–44. Available from: https://dx.doi.org/10.1109/ojcs.2020.2993259.
13) Yujia Z, Zhicheng D, Ji-Rong W. Encoding History with Context-aware Representation Learning for Personalized Search. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. Association for Computing Machinery. 2020;p. 1111–1120. Available from: https://doi.org/10.1145/3397271.3401175.
14) Shuqi L, Zhicheng D, Xu J, Jian Y, Ji-Rong N, W. PSGAN: A Minimax Game for Personalized Search with Limited and Noisy Click Data. In: and others, editor. SIGIR'19: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development i. 2019;p. 555–564. Available from: https://doi.org/10.1145/3331184.3331218.
15) Li T, Sahu AK, Talwalkar A, Smith V. Federated Learning: Challenges, Methods, and Future Directions. *IEEE Signal Processing Magazine*. 2020;37(3):50–60. Available from: https://dx.doi.org/10.1109/msp.2020.2975749.
16) Li E, Zeng L, Zhou Z, Chen X. Edge AI: On-Demand Accelerating Deep Neural Network Inference via Edge Computing. *IEEE Transactions on Wireless Communications*. 2020;19(1):447–457. Available from: https://dx.doi.org/10.1109/twc.2019.2946140.
17) Feng J, Rong C, Sun F, Guo D, Li Y. PMF: A privacy-preserving human mobility prediction framework via federated learning. *Proc ACM Interactive Mobile Wearable Ubiquitous Technologies*. 2020;4:1–21. Available from: https://doi.org/10.1145/3381006.