ISSN 2278-3091
Volume 9, No.3, May - June 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcs241932020.pdf
https://doi.org/10.30534/ijatcse/2020/241932020

An Enhanced Privacy protection scheme for Profile-based personalized search

Anas EL-ANSARI[1], Abderrahim BENI-HSSANE[2], Mostafa SAADI[3]
[1,2]LAROSERI laboratory, Computer Science Department, Sciences Faculty, Chouaib Doukkali University, El Jadida 24000, Morocco, anas.elansari@gmail.com
[3]Univ Sultan Moulay Slimane, LaSTI laboratory, ENSA Khouribga, B.P 77 Khouribga 25000, Morocco

ABSTRACT

With the information overload on the Internet, intelligent personalized systems come to play a critical role by providing tailor-made services to the user based on his interests. An example that recently becomes popular is the personalized search systems, offering users personalized answers. One of the major issues these systems face is the lack of user's trust due to the lack of privacy protection. Giving the user a personalized browsing experience usually comes at the cost of his privacy. Thus, most people are afraid of using such applications. To address this issue in this work, we propose a trade-off scheme between the personalization quality and the privacy risk, to keep the latter under control. We have studied the assets and drawbacks of the existing profile-based personalized search systems in general, from a privacy protection perspective. Furthermore, we present a new model to protect the user's privacy on different levels, using homomorphic encryption to enhance data protection.

Key words: Privacy, Personalized search, User Profile, Homomorphic Encryption.

1. INTRODUCTION

It has become difficult for people to find information on the web that satisfies their needs since information resources continue to grow and have far exceeded human processing capabilities [1]. The sheer information abundance often prevents people from finding desired information, and aggravates making correct and informed choices. For those reasons, users need intelligent personalized applications that can simplify information access and content discovery, based on each user's preferences, and delivers services in a most valuable and convenient way. An example of personalized systems that have recently become quite popular is the personalized search system. These systems offer users personalized answers based on their interests.

Two main challenges face personalization systems in general. The first one is building an accurate user profile that represents the user's real changing interests [2]. The second challenge is the privacy protection problem [3]; giving the user a personalized browsing experience comes at the cost of his privacy. Thus, most people are afraid of using such applications. Former research on this field focused on building profiles implicitly, by observing the user's activity. Because peoples' interests change over time, they focused on the implicit methods for constructing a user profile that can adapt and reflect the changing user interests. Since those systems depend mainly on collecting personal data, the privacy protection is one of our major concerns.

In this work, we propose a new scheme that improves the user privacy protection on different levels. The purposes of our research, in general, are first to increase the personalization quality by using accurate profiles capable of reflecting the user's changing interests [4]. Second, to ensure the user privacy by protecting his sensitive data on the client-side, through the Internet channel, and more importantly on the server-side where most privacy risks come (data misuse, leakage, etc.). The following section discusses related works. Next, we present a comparative study of the existing personalization system structures focusing on the user's privacy. Followed by a fourth section where we propose our new privacy protection model. The paper ends with the conclusion and perspectives.

2. RELATED WORK

This research relates to the field of personalization and recommender systems, the process of delivering information, items or services to the user considering his specific interests in the most adequate way and at the right time. These systems collect different user data types continuously, which raises many privacy-protection concerns. We discuss former works in this section focusing on three main points: web personalization systems applications, personalization methods and the privacy protection solutions.

- Web personalization system applications:

Some personalized Web systems were developed to help users browse news articles ([5], [6]), find scientific and research papers ([7]), purchase favorite products (Amazon [8], eBay [9]) improve search results ([10], [11]), recommend jobs ([12], [13]) or a combination of the above tasks ([14], [15]).

- Personalization methods:

Most personalized systems build user profiles by collecting and analyzing browsing history (visited Web pages) [16].

Other data sources have also been used, such as bookmarks in Basar [15], queries and search results in [17], [14] and [11]; even photos in [18]. The use multiple user data sources with Big Data technologies boosts the user modeling performance [19].To construct the profiles, personalized systems use a variety of learning techniques including the vector space model ([5], [10]), the probabilistic model [20], genetic algorithms [21], clustering [22], or even deep learning techniques [23]. Since the generated profile might comprise irrelevant topics, some of the above systems use filtering or rating algorithms to improve the profile's accuracy.

- Privacy protection solutions:

Most studies focus on improving the personalized services quality disregarding the user's privacy issues. Papers like [24], [25], [26] tried to address the privacy protection problems in personalized systems by protecting user identification using techniques such as the pseudo-identity, the group identity, no identity, and no personal information. Authors in [24] proved that the solution for the 1st level is too fragile. The last two levels' solutions are unpractical due to the high cost in communication and cryptography. Therefore, most efforts focus on the second level. Both [25] and [26] provide anonymity for users online by generating a group profile of k-users.Other works consider protecting the user's sensitive data, especially his profile. In this type of privacy-preserving systems, two main technique lines are used. Cryptographic methods and differentially private solutions.

In the cryptographic category, current studies generally use homomorphic encryption ([27], [28], [29]) or garbled circuits ([30], [31]) as a mechanism to protect the user data. Reference [27] generate recommendations in a privacy-preserving way with the use of data packing and homomorphic encryption. In [31], the authors propose a multi-party computation protocol that achieves strong private guarantee by encrypting the private QoS data of users using Yao's garbled circuits, and homomorphic encryption. Also, authors in [29] used partially homomorphic encryption to design two protocols for privacy-preserving trust-oriented POI recommendation based on the off-line encryption and parallel computing.

Differential privacy has become the widely accepted model of privacy during the past years, this solution aims to achieve a trade-off between the privacy level (how much the differentially private noises are needed to guarantee the privacy) and the utility (how good the outputs can be available for using).Reference [32] used differential privacy in a non-social recommender system. Though, when applied to a social recommendation, the result was an unacceptable loss of utility. Authors in [33] designed a privacy built-in client that perturb data on the user device. However, the utility of perturbed data may decrease due to the inhered volatility of the whole process. Reference [34] proposed another differential privacy scheme for neighborhood-based CF that can select neighbor privately; however, fails to maintain a good trade-off between the personalization quality and privacy protection. Shou et al [3] showed that better results could be achieved with privacy guarantee if the personalization is only performed based on less sensitive user data. The main idea is to expose only the insensitive part of the profile to the search engine by taking into account the user privacy requirement. Another lightweight technique called randomized perturbation was proposed in [35], [36]. Authors claim they can obtain accurate recommendations while adding randomness from a specific distribution to the user data to counter information exposure. However, the range of randomness is chosen by experience and have no provable privacy protection guarantee.

In our work, to counter privacy risks while preserving the personalization quality in QA and search systems, we propose a new model for personalized search systems based on the client-server collaborative structure that:

- Combines the cryptographic solution (Using Homomorphic encryption) and differential privacy.
- Protects the user sensitive data exposing only the insensitive part of the user profile and using the homomorphic encryption to protect the exposed part.
- Collect, store and protect user data on the client-side.
- Needs no assumption of trust in the server.
- Avoid data perturbation and accuracy loss.

3. PRIVACY IN PERSONALIZED SEARCH SYSTEMS

User privacy is the user's ability to insulate himself or some of his information and thereby express himself selectively [37]. Privacy is becoming a big concern in many fields such as Social Networks, Cloud Computing, Personalized systems, Etc. Authors in [3] classify privacy solutions into two main categories. One covers those protecting the user's identity. While the other includes those considering the sensitive data, particularly the exposed user profile. To preserve privacy in personalized search and QA systems, we have to consider two facts. 1st, improving the personalization quality means collecting more user data. 2nd, must hide the sensitive data in the user profile to place the privacy risk under control. Thereby we focus on the privacy risks of the second class. In this section, we classify personalized search systems into three distinct structures [38]. Based on the user profile's storage (on the client-side or the server-side) and use for personalization.).

3.1 Server-Side Personalization

The 1ststructure type is Server-side personalization (Figure.1), where the user profile is stored on the search engine side. This structure requires the user to create an account to identify himself. The search engine creates and updates the user profile either from the user's explicit input (e.g., asking the user to specify his personal interests) or by implicitly collecting the user's search history (e.g., query and click-through history). The latter approach requires no additional effort from the user and contains a richer description of his personal interests. *Google Personalized*is an example adopting this architecture. Most systems with such

structure ask users to provide consent before collecting and using his data. If the user gives his permission, the search system will hold his personal data on the server. Thus, from the user's perspective, this architecture does not have the minimum level of privacy protection.

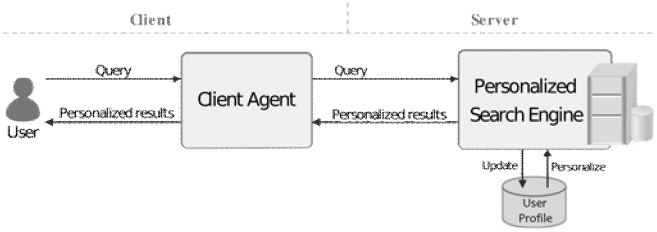

Figure 1: Server-side personalization structure

The lack of protection for such data raises privacy issues. For instance, the AOL query logs scandal [24] when the AOL Research released a file on its website containing twenty million search keywords for over 650,000 users, intended for research purposes [39]. In the report, AOL did not identify the users. Although, a number of queries included personal identifiable user information associated with identifiable user accounts. The New York Times located an individual from the released search records by cross-referencing them with phone-book listings. AOL admitted it was a mistake and removed the file, yet others redistributed the file on mirror sites. This example not only raises panic among users but also dampen the data publishers' enthusiasm in offering improved personalized services. The main advantages of this architecture resumed as follows:

- The search engine can use all of its resources (search patterns, document index) in the personalization algorithm.
- In addition, the client software generally requires no changes.
- The use of the server's high performance allows it to gain considerable time.
- Nevertheless, the personalized search systems based on this structure present some significant drawbacks:
- From the user perspective, it does not have the minimum level of privacy protection.
- Most users are afraid of using such systems, which can compromise their private data.

To address these structure problems, especially the user privacy problems, the client-side structure can be a solution.

3.2 Client-Side Personalization

The second type of structure is Client-side personalization (Figure.2), stores the user profile on the client side. The client agent sends queries to the search engine and receives results, same as an ordinary web search scenario. The client agent also performs a query expansion to generate a new personalized query before sending it to the search engine. Furthermore, as in [2], the client agent ranks the search results to match user preferences.

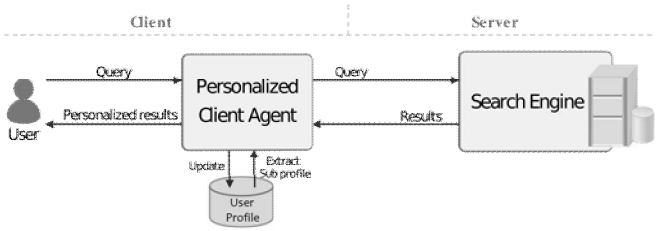

Figure 2: Client-side personalization structure

The advantages of this structure are as follows:

- Offer a richer user profile: combining the user's search history with his contextual activities (visited web pages) and personal data (emails, bookmarks) and producing a richer user profile.
- Reduce the privacy concerns since the user profile is on the client side.
- Distribute the overhead in computation and storage for personalization among the clients.

However, client-side personalization has some drawbacks:

- The client usually receives many results from the search engine, which increases the re-ranking process time and reduce its efficiency.
- Besides, the personalization algorithm cannot use the server knowledge (PageRank score of a result document).

Recent studies, to address the above drawbacks and improve the personalization quality without compromising user privacy, use a client-server collaborative structure.

3.3 Client-Server Collaborative Personalization

The client-server collaborative structure (Figure.3) is a balance between the past structures. The profile is stored on the client side, and the server is also used in the personalization process. At query time, the client agent extracts a sub-profile from the user profile to send it to the search engine along with the query. The search engine personalizes the results using the received context.

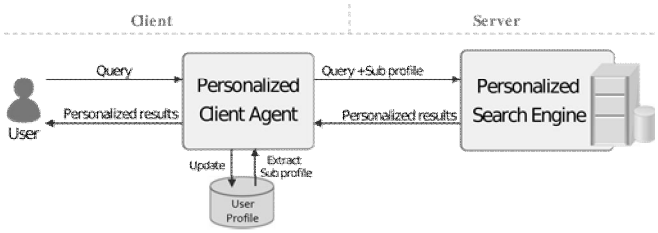

Figure 3: Client-Server collaborative structure

Personalization research in this category is minimum, probably due to the relatively complex architecture. In [3], the contextual information sent to the server is a generalized profile that specifies the user's search preferences without exposing the sensitive data in the user profile. The client agent extracts a sub-profile relevant only to a particular query. This sub-profile is a condensed version of the original user profile (generally a few terms or a weight vector from a user's search history). Thus, such structure can reduce the personal data obtained by the search engine.

Compared to previous structures, this one has more assets:

- Offers better privacy protection than a server-side structure, because the amount of user data collectible on the server-side is lower than in the case of a server-side personalization.
- It allows the use of a search engine's internal resources in the personalization algorithm.
- It presents a more personalized set of results.

Nevertheless, this structure has also some drawbacks:

- The condensed contextual information is not as efficient as the original user profile.
- Presents a privacy risk, even though the original user profile is not exposed, the generalized ones can still be collected on the server-side or with an eavesdropping attack (Figure.4).

Shou's model aims at protecting the user's privacy against a typical attack called eavesdropping. During a browsing session, a user sends many queries to the server, with each one a short version of his profile. Figure.4 shows how an attacker can obtain a significant proportion of the original profile, by collecting the sub-profiles and using the online ontology to figure the rest.

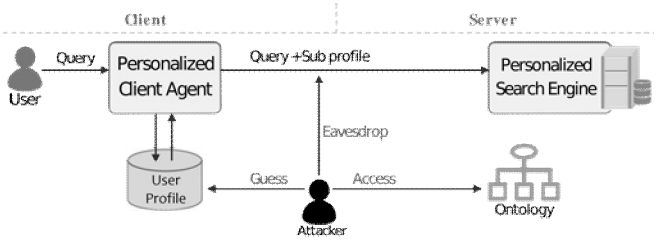

Figure 4: Attack model of personalized search (eavesdropping).

Considering the user profile P, each time a user enters a query q the system sends a part of P. If the attacker captures each generalized profile Gi, it is possible after n query to get a significant portion of the user profile P using the online taxonomy. Moreover, even if the generalized profile Gi contains no private data, the attacker can still obtain the full user profile by comparing the G_n to the ontology.

$$\sum_{i=1}^{n} G_i = G_n \rightarrow P \qquad (1)$$

Where n is a number of queries (depends on the user activity and time). To illustrate how an attacker can breach the user privacy, Figure.5 shows an example of a user profile (a) with two generalized profiles (Ga and Gb). The grey concepts in this figure reflect the user's private data. And the generalized profiles contain no sensitive data because the system stops at the parent nodes.

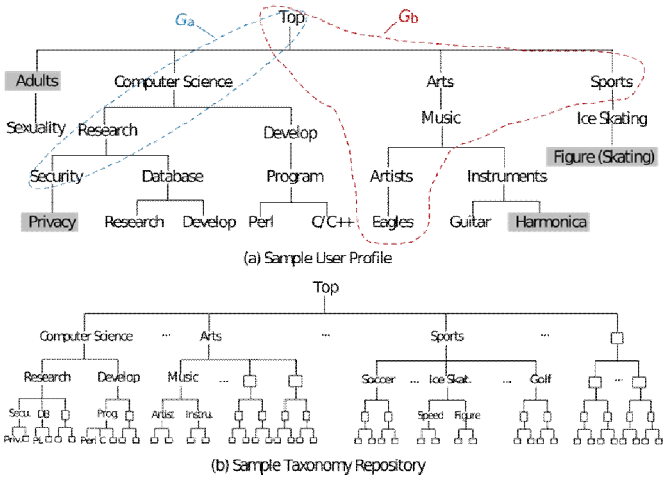

Figure 5: Taxonomy based user profile.

However, in Ga for example, the attacker can retrieve the sub-tree of Security relying on the taxonomy (b) in the same Figure.5, where Security is the parent of two nodes including a private one (Privacy). Therefore, if the probability of touching both branches is equal, the attacker has 50 percent confidence on Privacy leading to a high privacy risk. To avoid this model's drawbacks, we propose a scheme (Figure.7) based on the Client-Server collaborative personalization structure, which aims at protecting the original profile and limiting the possibility of guessing the real user profile from the generalized one.

4. PROPOSED PROFILE PROTECTION MODEL

In order to reduce the privacy risks in our system, we need to protect the user data on different levels. Starting with the user profile and reinforcing the system's structure even on the server-side.

4.1 User Profile

Building user profiles consist of learning from user browsing behaviors, hence this process is activated after each browsing session. In our model, we create and store the user profile on the client side. For reasons we discuss later in this section, instead of a single user profile we use a three layers profile:

- Short-term.
- Long-term.
- The archive.

After each browsing session, the system prepares and classify a list of visited concepts to the short-term and long-term layer. Users become suddenly interested in a topic, and once this topic loses its importance, they abandon it. For example, when the football world cup starts, sports fans would become more interested in this event. Once the contest ends, they would likely turn their interest to other sports.

The **short-term** layer includes the user's recent interests. One way to discover these interests is to define a threshold, and the system classifies all new concepts with weights above it to the short-term layer.

The **long-term** interests are more stable than the short-term ones. For example, programming languages are a stable interest for a user who works as a programmer. Therefore, the short-term layer contains the changing user interests while the long-term contains the stable ones.

The **archive** contains interests that are no longer important to the user. Concepts that are gradually losing their weight values (importance) eventually move to the archive.

We used a 3 layers profile for the following assets:
- The three layers profile will help the system adapt to user's interest change.
- With the short and long-term layers, we separate stable interests from occasional ones.
- Preserves user's privacy by minimizing the risk of guessing the real profile from the generalized one sent to the server.

Shou's model aims at minimizing the amount of sensitive data in the generalized profile sent with the query to the server. Though, even with this approach, there is still a privacy risk of exposing a significant part of the original profile.

For each query q, the client agent generates a contextual profile G from the user's original profile P corresponding to q. Then, it transfers the pair (q+G) to the server. Although the generalized profile G contains no sensitive information, it can be used to discover the user profile. By collecting all the G profiles, the attacker can obtain a significant part of the user profile P using the online taxonomy (described in formula 1).

To address this drawback, we use only the short-term layer to generate the G profile based on the query terms. The client agent extracts the sub-profile G as a set of hierarchical concepts respecting the user's sensitive data. Moreover, even if the attacker collects the generalized profiles, he cannot figure the changing short-term profile STP or access the rest (LTP and archive).

$$\sum_{i=1}^{n} G_i = G_n \rightarrow STP_t \neq STP_{t+1} \qquad (2)$$
$$STP_t \neq P\{STP, LTP, Archive\}$$

The client agent uses both the long-term and short-term layers to arrange and sortthe returned results. Once the client agent creates the initial profile, the user is allowed to specify his privacy requirements by selecting the sensitive data in his profile through a graphical user interface.

4.2 Profile Generalization

Once the user enters a query, the system sends it to the server along with a generalized profile. The server then personalizes the results according to the user's interests. Figure.6 describes the generalization process, which starts by mapping the query to the reference taxonomy to extract related topics. Then, computing a relevance value for each one. Finally, comparing them with the user profile considering the user's private data.

A. Query-Topic Mapping

The purposes of query-topic mapping are first, to compute a rooted sub-tree of the user profile P that contains all topics relevant to q. Second, is to calculate the preference values of q and all concepts in P. The client agent performs this procedure in the following steps:
- Find the set of topics R(q) in the ontology R relevant to q.
- Overlap R(q) with P to obtain the seed profile G0, which is a rooted sub-tree of P.

The client agent then creates a 1st version of the generalized profile G0 and refines it based on the user's privacy preferences. Then, it calculates a query-topic relevance value for each topic to help improve the search results.

B. Query-Topic Relevance

The query-topic relevance indicates how important a topic is to the giving query. Based on this metric the server can personalize the search results.
To calculate this metric, authors in [3] retrieve documents relevant to a user query q from the reference ontology using the conventional approach. Then, they classify all documents by their related topics and calculate the relevance for each concept as the number of its related documents (formula 3).

$$Rel(q,C_i) = Number\ of\ documents\ of\ C_i\ \ relevant\ to\ q \qquad (3)$$

However, using this formula in the following case:
- Topic1 (Total # of doc = 10, # of doc relevant to q=5)
- Topic2 (Total # of doc = 5, # of doc relevant to q=4)

Would result in considering the first topic more relevant to the query q than the second, which is logically inaccurate. This query is relevant to 80% of the documents in the second topic and only to 50% in the first one. Therefore, to improve the query-topic relevance formula, we propose to consider the total number of documents for each topic as described in the following formula:

$$Rel(q, Ci)=\frac{Number\ of documents of C_i relevant to q}{Total\ number\ of documents of C_i} \qquad (4)$$

C. Generalization Process

For some queries, called distinct queries this whole process of personalization contributes little or even reduces the search quality, while exposing the profile to the server would risk the user's privacy. A distinct query is a clear one that needs no personalization and used by most users to look for the same result. For example, by the question (who is the director of the *Titanic movie*) users are looking for the same answer (*James Cameron*). Same when typing (*Google*), most users are looking for the search engine.

Although personalization has a major impact on some queries' results, it can be useless in certain situations. Therefore, if the client agent identifies a distinct query during the generalization phase, it cancels the entire process and sends the query to the server without a generalized profile. The following algorithm scheme describes the whole process:

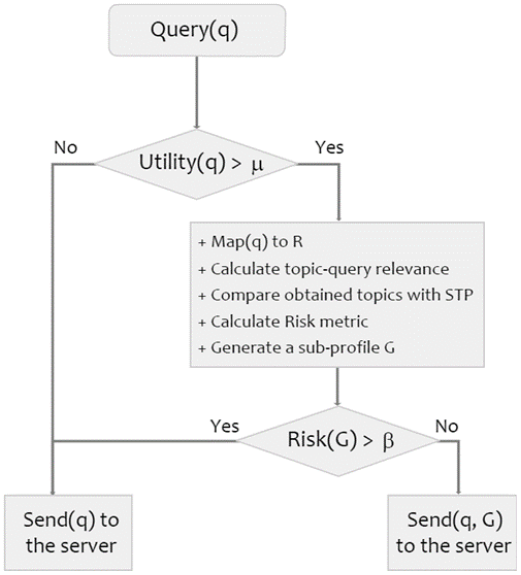

Figure 6: Generalization process algorithm.

Where:

- q: the user's query
- Utility(q): calculates the utility of generalizing a profile for the giving query q (distinct query or not)
- Risk(G): calculates the risk of exposing the topics in the generalized profile G
- μ and β: Thresholds
- R: the reference ontology/taxonomy.
- G: the generalized profile.

In the generalization process (Figure.6), two metrics are critical. First, the utility metric to decide whether to generate a profile with the query or not. Second, is the risk metric, which, based on the user's privacy requirements, helps decide which topics to expose in the generalized profile. The metrics and algorithms used in the profile generalization process were described in details in [3].

4.3 Proposed Model Structure

Our model (in Figure.7), is based on a client-server collaborative architecture (Figure.3) and aims at eliminating the drawbacks of the previous models:

- It uses a rich profile of tree layers for enhanced personalization.
- It eliminates the privacy risk of the generalized user profiles that can still be collected on the server side or in eavesdropping attack.
- The client receives a set of personalized and re-ranked results.
- On the server-side, the personalization algorithm can use all the server resources.
- Our model also preserves the user's privacy on different levels using homomorphic encryption.

Major search engines use HTTPS to secure the internet channel between the client and the server. However, many privacy scandals like AOL or the recent Facebook scandal showed that the risk could come from the server-side, not only from the channel. Therefore, the threat model (Figure.4) discussed in section 3.3 is not the only risk on the user privacy.

In our model, we aim at protecting the user profile from the server side risks (Data leakage or misuse), also reinforcing the channel security and data protection by using homomorphic encryption.

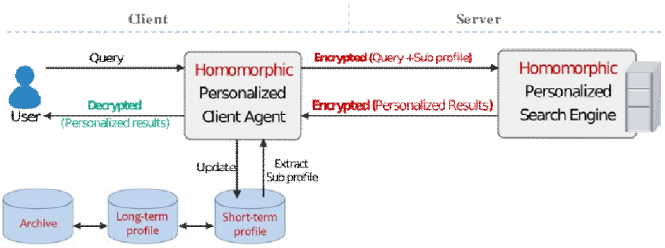

Figure 7: Proposed Privacy-preserving model.

Homomorphic encryption's philosophy is to delegate computing to the server without giving access to the data. The next section discusses the subject and the reasons for choosing this type of encryption system. We expect the homomorphic encryption to play a significant role in privacy protection. Especially, after its promising results on cloud systems [40].To illustrate the use of a homomorphic encryption system in our model, Figure.8 presents a use case scenario.

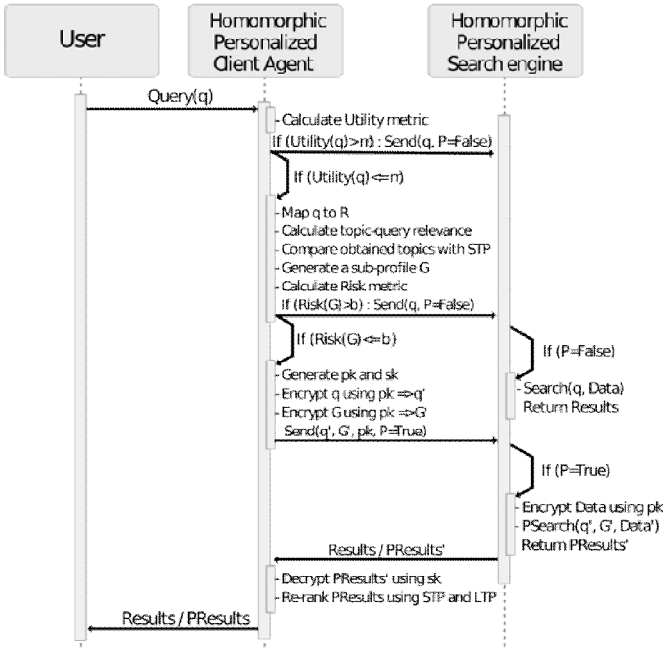

Figure 8: Homomorphic Encryption use case scenario.

In short, when a user enters a search query, the client agent generalizes a sub-profile (if needed) then encrypts and sends the pair (query + generalized profile) to the server along with the pk and a personalization value P. The latter helps the server decide either to run an encrypted personalized search or a normal one. Finally, the client agent decrypts the result with the private key to perform a re-ranking process using the whole profile (long-term and short-term).

Many challenges face the use of homomorphic encryption in such a system. Starting with efficiency, having algorithms that are effective, secure and supports all types of operations.

Another challenge is the robustness of the encryption and the size of the public key. In the RSA crypto-system for example, the size of the public key must be higher than 1024 bits to avoid its factorization. A bigger key size affects the data processing time, especially on the encryption/decryption.

Fully homomorphic encryption supports the personalized search over encrypted data using encrypted queries and profiles. However, it has the disadvantage of producing a very large cipher-text, larger than the corresponding plaintext. This increases the search time due to the encrypted index size.

To solve this problem, we can use compression techniques [41], or other encryption techniques (Quantum Homomorphic Encryption [42]).

5. CONCLUSION

The work presented in this paper focuses on protecting the user's privacy by protecting the sensitive data in his profile. The model we propose counters various privacy treats on different levels. The use of the homomorphic encryption as an extra protection layer reinforces the user's privacy protection, which is still one of the major issues in the field of web personalization systems.

A personalized system must ensure privacy protection to earn the user's trust. Otherwise, only a minority of users will use it, to whom the personalized experience is more important than their privacy. In the future, we plan to implement our model in a personalized question answering system we have built in a previous work [43]. Also, enhancing the user profile created in [4] by gathering all the possible user data (browsing activities, social networks profiles, etc.).

REFERENCES

1. Challam, V., Gauch, S., Chandramouli, A., "**Contextual search using ontology-based user profiles"**. In: *Large Scale Semantic Access to Content (Text, Image, Video, and Sound)*. RIAO 2007. Le centre de hautes etudes internationales d'informatique documentaire, Paris, France, pp. 612–617. URL http://dl.acm.org/citation.cfm?id=1931390.1931447
2. Hawalah, A., Fasli, M., 2015. "**Dynamic user profiles for web personalization"**. Expert Systems with Applications 42 (5), 2547 – 2569. URL http://dx.doi.org/10.1016/j.eswa.2014.10.032
3. Shou, L., Bai, H., Chen, K., Chen, G., 2014. **Supporting privacy protection in personalized web search**. IEEE Transactions on Knowledge and Data Engineering 26 (2), 453–467.
4. Anas El-Ansari, Abderrahim Beni-Hssane, and Mostafa Saadi. 2020. **An improved modeling method for profile-based personalized search**. In *Proceedings of the 3rd International Conference on Networking, Information Systems & Security (NISS2020)*. Association for Computing Machinery, New York, NY, USA, Article 55, 1–6. DOI:https://doi.org/10.1145/3386723.3387874
5. Wu, C., Wu, F., An, M., Huang, J., Huang, Y., Xie, X., 2019. "**NPA: neural news recommendation with personalized attention"**. CoRR abs/1907.05559. URL: http://arxiv.org/abs/1907.05559
6. Bountouridis, D., Harambam, J., Makhortykh, M., Marrero, M., Tintarev, N., & Hauff, C. (2019, January). **Siren: A simulation framework for understanding the effects of recommender systems in online news environments**. In *Proceedings of the conference on fairness, accountability, and transparency* (pp. 150-159).
7. Mohseni, M., Maher, M. L., Grace, K., Najjar, N., Abbas, F., Eltayeby, O., 2019. "**Pique: Recommending a personalized sequence of research papers to engage student curiosity"**. In: Artificial Intelligence in Education. Springer International Publishing, Cham, pp. 201–205.
8. Smith, B., Linden, G., may 2017. **"Two decades of recommender systems at amazon.com"**. IEEE Internet Computing 21 (03), 12–18.
9. Greenstein-Messica, A., Rokach, L., 2018. **"Personal price aware multi-seller recommender system: Evidence from ebay"**. Knowledge-Based Systems 150, 14-26.
10. Yu, P., Ahmad, W. U., Wang, H., 2018. **"Hide-n-seek: An intent-aware privacy protection plugin for personalized web search"**. In: *The 41st International ACM SIGIR Conference on Research; Development in Information Retrieval*. SIGIR '18. ACM, NY, USA, pp. 1333–1336.
11. Tanudjaja, F., Mui, L., 2002. **Persona: A contextualized and personalized web search**. In: *Proceedings of the 35th Annual Hawaii International Conference on System Sciences* (HICSS'02) Volume 3. HICSS '02. IEEE Computer Society, Washington, DC, USA, pp. 67- URL http://dl.acm.org/citation.cfm?id=820741.820985
12. Adeagbo M. A., Akhigbe B. I. 2, Afolabi B. S, "**Towards A Job Recommender Model: An Architectural-Based Approach**" Volume 8, No.6, November 2019 ISSN 2278-3091, International Journal of Advanced Trends in Computer Science and Engineering. https://doi.org/10.30534/ijatcse/2019/94862019
13. MarveeCheska B. Natividad, Bobby D. Gerardo, Ruji P. Medina, **A Career Track Recommender System for Senior High School Students using Fuzzy Logic** Volume 8, No.5, September - October 2019 ISSN 2278-3091, International Journal of Advanced Trends in Computer Science and Engineering. https://doi.org/10.30534/ijatcse/2019/97852019
14. Pazzani, M., Muramatsu, J., Billsus, D., 1996. **Syskill & webert: Identifying interesting web sites**. In: *Proceedings of the Thirteenth National Conference on Artificial Intelligence* - Vol 1. AAAI Press, pp. 54–61.
15. Thomas, C. G., Fischer, G., 1997. **"Using agents to personalize the web"**. In: *Proceedings of the 2Nd International Conference on Intelligent User Interfaces*. IUI '97. ACM, New York, NY, USA, pp. 53–60.
16. Dennis, W. L., Erwin, A., Galinium, M., 2016. **"Data mining approach for user profile generation on**

advertisement serving". In: *8th International Conference on Information Technology and Electrical Engineering* (ICITEE). pp. 1–6.

17. Rich, E., 1998. **Readings in intelligent user interfaces**. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, Ch. User Modeling via Stereotypes, pp. 329–342.
18. Berger, H., Denk, M., Dittenbach, M., Pesenhofer, A., Merkl, D., 2007. **Photo-based user profiling for tourism recommender systems**. In: Psaila, G., Wagner, R. (Eds.), E-Commerce and Web Technologies. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 46–55.
19. Farseev, A., Nie, L., Akbari, M., & Chua, T. S. (2015, June). **Harvesting multiple sources for user profile learning: a big data study**. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval* (pp. 235-242). https://doi.org/10.1145/2671188.2749381
20. Chaney, A. J., Blei, D. M., Eliassi-Rad, T., 2015. **A probabilistic model for using social networks in personalized item recommendation**. In: *Proceedings of the 9th ACM Conference on Recommender Systems. RecSys '15*. ACM, New York, NY, USA, pp. 43–50. https://doi.org/10.1145/2792838.2800193
21. Lv, G., Hu, C., Chen, S., **Research on recommender system based on ontology and genetic algorithm**. Neurocomputing 187, 92 – 97, recent Developments on Deep Big Vision. 2016.
22. Liao, C.-L., Lee, S.-J., 2016. **A clustering-based approach to improving the efficiency of collaborative filtering recommendation**. Electronic Commerce Research and Applications 18, 1-9.
23. Singhal, A., Sinha, P., Pant, R., 2017. **Use of deep learning in modern recommendation system: A summary of recent works**. CoRR. URL http://arxiv.org/abs/1712.07525
24. Hafner, K., 2006. **Tempting data, privacy concerns; researchers yearn to use AOL logs, but they hesitate**. The New York Times.
25. Xu, Y., Wang, K., Yang, G., Fu, A. W., 2009. **Online anonymity for personalized web services**. In: *Proceedings of the 18th ACM Conference on Information and Knowledge Management. CIKM '09*. ACM, New York, NY, USA, pp. 1497–1500.
26. Zhu, Y., Xiong, L., Verdery, C., 2010. **Anonymizing user profiles for personalized web search**. In: *Proceedings of the 19th International Conference on World Wide Web. WWW '10*. ACM, NY, USA, pp. 1225–1226. https://doi.org/10.1145/1772690.1772886
27. Erkin, Z., Veugen, T., Toft, T., Lagendijk, R. L., Jun. 2012. **Generating private recommendations efficiently using homomorphic encryption and data packing**. Trans. Info. For. Sec. 7 (3), 1053–1066. URL: https://doi.org/10.1109/TIFS.2012. 2190726
28. Liu, A., Li, Z.-X., Liu, G.-F., Zheng, K., Zhang, M., Li, Q., Zhang, X., Sep 2017. **Privacy-preserving task assignment in spatial crowdsourcing**. Journal of Computer Science and Technology 32 (5), 905–918.
29. Liu, A., Wang, W., Li, Z., Liu, G., Li, Q., Zhou, X., Zhang, X., 10 2017. **A privacy-preserving framework for trust-oriented point-of-interest recommendation**. IEEE Access PP, 1–1.
30. Liu, A., Zheng, K., Li, L., Liu, G., Zhao, L., Zhou, X., 2015. **Efficient secure similarity computation on encrypted trajectory data**. *IEEE 31st International Conference on Data Engineering*, 66–77.2015.
31. Li, L., Liu, A., Li, Q., Liu, G., Li, Z., Jul. 2016. **Privacy-preserving collaborative web services qos prediction via yao's garbled circuits and homomorphic encryption**. J. Web Eng. 15 (3-4), 203–225. URL http://dl.acm.org/citation.cfm?id=3177210.3177212
32. McMcSherry, F., & Mironov, I. (2009, June). **Differentially private recommender systems: Building privacy into the netflix prize contenders**. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 627-636). https://doi.org/10.1145/1557019.1557090
33. Shen, Y., Jin, H., 2016. **Epicrec: Towards practical differentially private framework for personalized recommendation**. In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. CCS '16*. ACM, New York, USA, pp. 180–191. URL http://doi.acm.org/10.1145/2976749.2978316
34. Zhu, T., Li, G., Ren, Y., Zhou, W., Xiong, P., **Differential privacy for neighborhood-based collaborative filtering**. In: *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining* (ASONAM 2013). pp. 752–759.
35. Polat, H., Du, W., 2003. **Privacy-preserving collaborative filtering using randomized perturbation techniques**. In: *Proceedings of the Third IEEE International Conference on Data Mining*. ICDM '03. IEEE Computer Society, Washington, DC, USA, pp. 625–.
36. Zhu, J., He, P., Zheng, Z., Lyu, M. R., 2015. **A privacy-preserving qos prediction framework for web service recommendation**. In: *Proceedings of the 2015 IEEE International Conference on Web Services*. ICWS '15. IEEE Computer Society, Washington, DC, USA, pp. 241–248. URL https://doi.org/10.1109/ICWS.2015.41
37. Wikipedia, 2019. **Privacy wikipedia**. URL: https://en.wikipedia.org/wiki/Privacy
38. Shen, X., Tan, B., Zhai, C., Jun. **Privacy protection in personalized search**. SIGIR Forum 41 (1), 4–17.2007. URL http://doi.acm.org/10.1145/1273221.1273222
39. Wikipedia, 2019. **Aol search data leak**. URL: https://en.wikipedia.org/wiki/AOL_search_data_leak
40. El Makkaoui, K., Beni-Hssane, A., Ezzati, A., & El-Ansari, A. (2017). **Fast cloud-RSA scheme for promoting data confidentiality in the cloud computing**. Procedia computer science, 113, 33-40. URL: https://doi.org/10.1016/j.procs.2017.08.282
41. Boucenna, F., Nouali, O., Kechid, S., Tahar Kechadi, M.,**Secure inverted index based search over encrypted cloud data with user access rights management**. Journal of Computer Science and Technology 34 (1), 133–154. 2019. URL https://doi.org/10.1007/s11390-019-1903-2

42. Zhou, Q., & Lu, S. (2017). **Quantum Search on Encrypted Data Based on Quantum Homomorphic Encryption**. arXiv preprint arXiv:1711.10066. URL https://arxiv.org/abs/1711.10066
43. El-Ansari, A., Beni-Hssane, A., & Saadi, M. (2017). **A multiple ontologies based system for answering natural language questions**. In *Europe and MENA Cooperation Advances in Information and Communication Technologies* (pp. 177-186). Springer, Cham. URL: https://doi.org/10.1007/978-3-319-46568-5_18