Demonstration

Distributed Secure Search in the Personal Cloud

Thu T.B. Le[1,3], Nicolas Anciaux[1,2], Sébastien Gilloton[1], Saliha Lallali[1,2], Philippe Pucheral[1,2], Iulian Sandu Popa[1,2], Chao Chen[1,2]

[1] INRIA, France
FirstName.LastName@inria.fr

[2] U. Versailles St-Q. en Y., France
FirstName.LastName@uvsq.fr

[3] HCMC U. of Technology, Vietnam
thule@hcmut.edu.vn

1. INTRODUCTION

The Personal Cloud paradigm emerges as a decentralized and privacy preserving solution to manage personal documents under users' control. It can be seen as an alternative to the current Web model, which centralizes the complete digital life of millions of individuals in data silos, and increases frustration generated by the weak control of the individuals on the way their personal data are shared, used and disseminated. The home cloud is the most emblematic form of Personal Cloud. It can be thought of as a dedicated box connected to the user's internet gateway, equipped with storage, computing and communication facilities [11], running a personal server and acquiring data from multiple sources [3]. This personal server is in charge of organizing the personal dataspace in a document database style to ease its management and to protect it against loss, theft and abusive use. Many startups (e.g., OwnCloud, CozyCloud, etc.) and research projects (e.g., PlugDB at Inria or OpenPDS at MIT) investigate this direction.

To make the vision of the Personal Cloud reality, two important challenges related to data management have to be considered. First, leaving the data management control into the user's hands pushes the security issues to the user's computing platform as well. Hence, besides the management of collections of personal documents of any type, the personal cloud takes the security and privacy in charge. This requires protecting personal documents and their metadata by means of encryption and evaluation of access control rules. This challenge is paramount considering that the Personal Cloud paradigm puts a significant part of the digital life of the individual in the user's hands.

The second challenge is rooted in the high level of decentralization of the Personal Cloud, i.e., each user owns her personal cloud (see Figure 1). Indeed, this user-centric architecture must not hinder the development of global data services of great interest for the individuals, the companies and the community, which is also required to meet a viable business model. Hence, as in a centralized setting, certain applications require crossing data from multiple individual Personal Clouds. This is the case of any application developed for communities of users sharing a similar interest. For example, within a community of patients suffering from the same pathologies, each participating user may provide her own set of information such that distributed searches may help to identify within the community the most relevant documents related to current treatments or symptoms, or more generally, help users to share and benefit from each other experiences. Clearly, this must be performed without exposing the privacy of the participating users.

This demonstration tackles precisely these two challenges. Our approach relies on a secure hardware based co-server, called *secure token* hereafter, which provides a search engine interface to users and applications to manage the documents with high security and privacy guarantees. This search engine, described in detail in [5], manages the encryption/decryption of documents, answers local searches and enforces access control rules on the fly with good performance. In this demonstration, we extend this previous work to provide a secure distributed search engine, such that applications can query the documents stored in a large number of Personal Clouds. Any global search has to be accomplished while preserving the participants' privacy, i.e., no further information beside the computation result can be learned by any participant or a third party. This property is key to encourage users to participate in global computations.

The implementation of a secure distributed search engine is however challenging. Considering a top-k search where the score of each document is evaluated by a *tf-idf* metric and answering global searches over a (large) community of users require (1) to compute some global values (i.e., the total number of shared documents and the inverse frequency of the query terms in all these documents), (2) to evaluate the score of each document accessible for the query according to these global values, and then (3) to identify the k documents with the highest scores among the set of participants.

Although users may accept to contribute to a given community of interest by granting a right to search over a subset of their own personal documents, the risk that any compromised participant gains access to the complete collection of documents must be avoided. This can be achieved by minimizing the amount of information exposed during query evaluation. While the final result of each query (i.e., the k most relevant documents) can be published to the community, neither the intermediate computations nor the complete set of local documents eligible for a given query should be revealed.

Computing a result without revealing input data can be done (1) by outsourcing the data on a trusted party, but we consider this option as not satisfactory in personal cloud context where no trusted entity clearly appears in the scenario, (2) by using Secure-Multi-Party (SMC) cryptographic techniques, but these techniques cannot currently meet both query generality and scalability objectives [10] or (3) by relying on privacy preserving distributed query computation techniques (see Section 3).

Our approach capitalizes on the tamper resistant hardware available on each personal cloud to form a global secure decentralized data platform. No plaintext data will be exposed outside of the secure elements except the final result to be published. The risk of data disclosure thus only depends on the possibility of the secure elements of certain participants to be compromised, i.e., the secure token has been tampered with and the decryption keys it contains may become accessible to malicious participants. Although these attacks are highly difficult and costly to conduct, they cannot be totally ignored. It is therefore mandatory to quantify the impact of such an attack and to propose computation strategies minimizing it. We then introduce privacy metrics linked to the amount of intermediate personal data made accessible to the secure infrastructure during the computation and show how it can be minimized.

The aim of this demonstration is to show that practical and efficient solutions can be devised to evaluate distributed searches within large communities of Personal Cloud users with a very limited privacy risk for the participants. More precisely, we

demonstrate that "gossip" based computations [8] (1) can provide accurate search results with good performance and scalability (i.e., large communities of users, with thousands to millions of personal clouds) and (2) can drastically minimize the risk of privacy violation even if compromised participants are involved in the computation.

2. ARCHITECTURE AND SCENARIOS

Let us consider the Secure Personal Cloud Platform of Alice as pictured in Figure 1. The main component is a home cloud data system gathering personal data from multiple sources (employer, banks, hospitals, commercial web sites, etc.) and devices (smart meters, quantified-self devices, smartphones, cameras, etc.). The Personal Cloud can be implemented by any type of computing platform with storage facilities such as a set-top box or a plug computer. This data system is complemented by a secure co-server which can be hosted by any type of tamper-resistant devices flourishing today, e.g., Mobile Security Card (produced by Giesecke & Devrient), Personal Portable Security Device (produced by Gemalto and Lexar), Multimedia SIM card [4] or Secure Portable Token [5]. Whatever its commercial name and form factor, a tamper-resistant device embeds a secure microcontroller (e.g., a smart card chip) connected to a large NAND Flash memory (e.g., an SD card) and can communicate with a host through a USB, Bluetooth or Ethernet port. Open hardware secure tokens are also provided (e.g., by the Versailles Science Lab, http://tinyurl.com/UVSQ-Lab), and can be built by any electronic manufacturer.

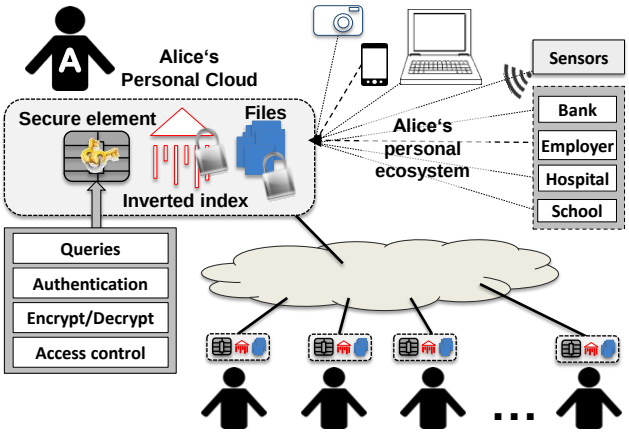

Figure 1: Secure Personal Cloud architecture

Alice participates here to a community of patients. Each secure token acts as a participant in a secure distributed search allowing users and applications to express full-text search queries over document collections stored within the community. The token being the root of security, it is in charge of data access control, encryption/decryption and metadata maintenance, that is the insertion, update and deletion of documents in the full-text search index. Each secure token locally stores an index of the user's documents build on the documents content and including a set of access control terms associated with these documents. It also stores cryptographic keys used to protect the documents, stored encrypted locally or remotely on external cloud storage. Each user grants access to the documents by specifying access control rules (conjunction and disjunction of access control terms). The secure tokens collaboratively compute the result of top-*k* information retrieval queries issued by users or applications by only considering the documents matching the access control rules on each personal cloud.

The documents can be any form of files (pictures, text files, pdf files, mails, data streams produced by sensors, etc.) associated to a set of terms. The terms are extracted from the file content and from metadata describing it (e.g., name, type, date, creator, words, visterms, tags set by the user herself). A query issued by a user or an application can be any form of term search expression, with a ranking function (e.g., *tf-idf* [15]) identifying the top-k most relevant documents (see Section 3). Only the documents granted to the user/application must appear in the query result. Hence, such a search engine can be used locally to query any documents entering a user dataspace, in the same spirit as a Google desktop or Spotlight augmented with access control capabilities.

A distributed search can be initiated by any member of the community to retrieve the most relevant documents for the query within the community. Each member within the community can contribute to the query by granting access to her own documents.

The question is how to pool these dataspaces without the assistance of a central server. The objective indeed is to avoid centralizing sensitive information that may be at risk in case the server is compromised. We consider the existence of a network infrastructure enabling direct information exchange between any pair of personal clouds taking part in a distributed search query. We also assume that each participant owns a public/private key pair. Within a community, all participants exchange their public key at the time of registration. This could be achieved using traditional PKI or GPG techniques.

3. TECHNICAL CHALLENGES

Search engine requirements. To identify the top-k most relevant documents in a given collection for a certain query, a ranking function is used to score each document. For this demonstration, we use the classical *tf-idf* function:

$$tf.idf(d) = \sum_{t \in Q} f_{d,t} \cdot log\left(\frac{N}{F_t}\right)$$

where $f_{d,t}$ is the occurrence number of term t in the document d, N is the total number of indexed documents and F_t is the number of documents that contain t. For a local query (which searches into single personal cloud) N and F_t are local values (i.e. the local number of indexed documents and the local number of documents that contain the term t). To evaluate a global search, the scores computed in the different personal clouds must be comparable. This requires computing beforehand the global values of N and F_t to be used in the previous formula. Then, the local top-k scores have to be exchanged and compared to find the global result. To this end each participant has to transmit their data to others.

Security constraints. The secure tokens are the unique source of trust in the architecture. They are endowed with a tamper resistant element (secure microcontroller) which prevents physical attacks and also its owner from having access to the secret data it contains and manipulates. Hence, even the holder of the secure token cannot spy intermediate data manipulated by his token during a computation (similar with a banking card holder that has no access to the cryptographic secret stored in his card and cannot spy its data processing). However, despite its high level of security, we cannot exclude the possibility of having a small percentage of hacked tokens (e.g., as a result of a sophisticated attack from the token owner). Such an attack would lead the personal cloud owner to have access to the cryptographic material stored inside her secure token. From this, she can potentially decrypt any encrypted information sent during the computation to her personal cloud. One objective of this demonstration is to evaluate the risk taken by the participants of a community to have their personal data unexpectedly exposed in this case. This risk analysis is essential since the hardware is left into users hands. Breaking a set of secure tokens should not put the personal documents of the whole community at risk. To evaluate that risk, we measure (1) for a given set of (potentially compromised) secure tokens, the amount of intermediate results and documents exposed during the evaluation of a distributed search query, and (2) for a given set of (not compromised) participants, the amount

of her own information transmitted to remote secure tokens and the number of secure tokens to which this information is disseminated. The first metric gives an estimate of the benefit of an attack for the attackers. The security being evaluated as a ratio between the cost of the attack and its benefit, the lower the value is, the better the security is. The second metric estimates the impact of the information leak for an honest participant. Our objective is to keep both metrics as low as possible, and never favor a solution which puts the complete dataset at risk in case of successful attacks.

State of the art solutions. Distributed query processing and the top-k queries are well investigated topics. Although, to our knowledge, decentralized secure computation of top-k queries on a population of secure elements has not been investigated yet, several previous proposals are related to our work. A first approach to solve the problem is to rely on a super peer (i.e., a central manager or a designated peer used as a coordinator) as proposed in [6, 12, 13, 14]. However, these solutions do not comply with our security requirement since the complete dataset becomes at risk if the super peer is compromised. Other existing approaches [1, 7] propose to organize the peers as a tree to process the queries. However, tree architectures are very sensitive to peers failures. In addition, the peers participating in the tree potentially gather a lot of branches and can thus be the transit point of a large amount of data, leading to large privacy breaches if compromised. Other solutions found in the literature use gossip protocols, which are highly suitable for fully decentralized architectures. In [6], a gossip protocol is used to broadcast top-*k* queries only to the peers who have similar interests as the querier, which is very interesting in our context if transposed to "trusted peers" and is part of our future works. However, the solution proposed in [6] assumes that the querier can see all the intermediate results coming from the participants, which would not be acceptable in our context. In Chiaroscuro [2], participating devices collaborate using gossip style computations to achieve privacy thanks to encryption and differential privacy. But this solution is dedicated to perform clustering operations on time series. A recent proposal [13] addresses the problem of computing SQL aggregate queries over an asymmetric architecture composed of potentially large populations of secure tokens and a central server. However, the focus is to prevent data inferences while delegating operations to the central server. Also, the secure tokens share the same secret key, which incurs the risk of exposing the complete dataset if one secure token is hacked.

4. DESIGN OF THE SOLUTION

Our solution to perform the distributed search relies on three main phases described as follows. In Phases 1 and 3, gossip computations algorithms [8] are used to respectively compute a sum and a top-k using of gossip computation algorithms.

Phase 1: computation of global N and F_t. The query is broadcasted to the participants. The secure token of each participant computes locally its own contribution to the global N and F_t, considering only the documents compliant with its active access control rules. The local contributions are then aggregated in order to compute the global values for N and F_t according to the push-sum algorithm proposed in [8]. The precision on the approximate values obtained on each participant for N and F_t can be controlled by the number of gossip exchange rounds which remains reasonable even for a large number of participants:

$$nb\_round = O(\log(n) - \log(\varepsilon) - \log(\delta))$$

where *nb_round* is the number of rounds necessary in a network of n participants to obtained with a probability at least $1-\delta$ a result with an error under ε. At the end of this step, each participant has an approximate value of the global N and F_t values.

Phase 2: local computations of the top-*k*. Based on these global values, each secure token computes locally the global scores for the documents compliant with its active access control rules and produces a local top-*k*.

Phase 3: computation of the global top-*k*. A new phase of gossip communication starts during which secure tokens exchange their top-*k*. At each round, each secure token receives k tuples (cloud_id, doc_id, score) from another peer and selects the k highest scores within the union of the received tuples and the local tuples. Then, it chooses randomly the next peer to which it sends its current local result. In this way, the tokens refine their local results at each round of the protocol. After a given number of rounds [8], all the tokens share a set of local results that are close to the exact global result. Due to the random characteristic of gossip protocols, the final result is only an approximation of the exact result (i.e., the one which would have been obtained on a centralized union of all the authorized documents in all participating personal clouds). The present demonstration will show that the results are good and that the improvement of the result accuracy is fast in number of rounds.

To enforce the security of the protocol and meet the privacy requirement, an asymmetric encryption/decryption system is used. Each token owns a private/public key pair. We assume in the demonstration that every participant has at disposal the complete list of public keys of all the participants. During the gossip communication phases tokens randomly choose a public key in the list, encrypt their message to transmit it to the corresponding secure token which decrypts the message with its private key. A token can thus be sure that its message can only be read by the chosen recipient, which limits the data exposure risk.

5. DEMONSTRATION

In this section, we present our prototype platform and describe the demonstration scenario covering the security and the performance of the proposed solution for distributed search in the secure Personal Cloud architecture.

5.1. Platform

Hardware Platform. The demonstration platform is an instance of the architecture depicted in Figure 1. A laptop is used as the communication infrastructure, and 20 secure tokens (see Figure 2) are used as participants. Each token is running the distributed search algorithm based on gossip computations as presented in the previous section, and evaluates the local access control rules. The local searches are performed using a previous prototype [11]. The secure tokens are equipped with a 32 bit RISC MCU clocked at 120 MHz with 128 KB of static RAM and 1MB of NOR Flash (to store the code of the distributed search engine). The MCU is connected to a smartcard chip hosting the cryptographic material and to a µSD card which stores the inverted index use by the embedded search engine. The PC which connects to all tokens via a USB port plays the role of the network infrastructure, controls the communications between tokens, and shows exchanged data and results it receives from tokens.

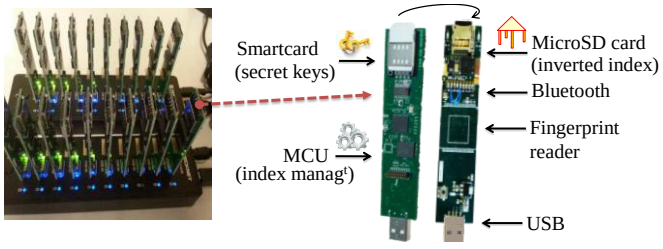

Figure 2: Secure Tokens used in the demonstration.

Graphical User Interface.. The GUI is used to control the system and evaluate our privacy metrics. Any participant can issue a global search query and retrieve the relevant files. The demonstration interface compares the efficiency, performance and privacy offered by our distributed search algorithm with a secure

solution where all the secure tokens would share the same encryption key (in the spirit of [13]) and with tree based approaches inspired by [1]. The accuracy of the search is compared with a central search computed over the union of the document databases stored in the participating personal clouds (using traditional precision and recall metrics). To evaluate the privacy level of our computation technique, the GUI also shows the probability of data disclosure to the attackers and the number of external documents that an attacker could read. To this end, the interface enables choosing certain participants to be considered as being corrupted.

Dataset. We use the Pseudo-desktop collection [9] as the baseline dataset of the demonstration. It includes more than 27000 documents (representing emails, photos, pdf, docs, and ppts). We randomly spread the documents among the tokens. We also use a synthetic dataset to show the scalability of the approach with more documents in each personal cloud.

5.2. Scenario

The first goal of this demonstration is to present the performance of the distributed search. Each participant has set her own access control rules. One of them issues a query (set of terms) and chooses the expected error rate. The number of iterations in the gossip phases is then fixed according to this error rate. The result (an approximate top-*k* obtained by each participant) is compared to the exact one, which is computed by the PC on the union of the authorized documents. Query times, average score error, precision and recall are presented and compared. The demonstration shows that good performance can be achieved even for low error rates. The average score error is given by the difference between the scores obtained in the approximate results for each token and the exact score obtained with a centralized search. The recall (respectively, precision) is given by the ratio between the number of documents present in both the approximate result and the exact result, and the number of results (respectively, the number of documents in the exact result with a score greater than the minimum in the approximate result).

The second goal of this demonstration is to focus on the privacy properties of gossip computations. The attendees choose some personal clouds as being corrupted before running the query. First, the interface will show the ratio between the number of distinct couples (cloud_id, doc_id, score) computed by each token which are exposed to the attackers, and the total number of couples (cloud_id, doc_id, score) in the local top-*k* computed during the query. Second, for each participant considered as honest, the interface plots the ratio between the number of couples which do not appear in the result and have been transmitted (directly or transitively) to one of the attackers during the execution, and the number couples which have been computed locally (typically *k*).

These two metrics obtained with our search algorithm will be compared to those obtained using alternatives representative of state of the art solutions: (1) a tree based evaluation, where the data flow between the participants is modeled as a tree structure (inspired by [1]) and (2) a secure query execution where all secure tokens would share a same secret key to evaluate the query (inspired by [13]).

5.3. Demonstration results

In terms of performance, the execution query time with our proposal is larger compared to a centralized database. This is obvious since our system requires a number of gossip iterations. However, the execution time is reasonable even on large databases and with large numbers of participants (obtained by simulation in the demonstration). In terms of precision, the average score error is less than 10% and the precision lies between 0.6 and 0.9 with a relatively low number of iterations. These values show that our proposal can be used in practice. In terms of privacy and security, our proposal shows much better results than existing approaches. Typically, honest users involved in the computation disclose few couples (cloud_id, doc_id, score) to the attackers. Considering 10% of attackers, the probability to expose a couple to an attacker is around 20% in our experiments. Our technique could be improved by choosing in the first gossip steps only users considered as trusted by the participant. This would decrease this number drastically (this is part of our future works). And for an attacker, the benefit of an attack is very small since only a very tiny proportion of the intermediate results can be obtained.

6. CONCLUSION

The emerging Personal Could paradigm holds the promise of a Privacy-by-Design storage and computing platform where personal data remains under the individual's control while being shared by valuable applications. In this demonstration, we present a distributed secure search engine with the objective to provide a high level of security founded on the introduction of low cost secure tokens in the architecture. This architecture minimizes the loss of privacy risks even if some participants are compromised, i.e., could bypass the tamper resistance of the token. While many personal cloud platforms are flourishing, riding the wave of repeated scandals blemishing the typical centralized management of personal data, none of them provides such a tangible source of trust to the individuals. We hope that the platform demonstrated here, which enable both local application and distributed ones, emphasizes the interest of studying new database techniques based on secure hardware for the database community. .

7. ACKNOWLEDGMENTS

This study is founded by the ANR KISS project grant n°ANR-11-INSE-0005 and by the Inria International Project Lab CityLab.

8. REFERENCES

[1] Akbarinia, R., Pacitti, E., and Valduriez, P. 2006. Reducing network traffic in unstructured p2p systems using top-k queries. Distributed and Parallel Databases, 19.

[2] Allard, T., Hébrail, G., Masseglia, F., Pacitti, E. Chiaroscuro: Transparency and Privacy for Massive Personal Time-Series Clustering. In ACM SIGMOD 2015.

[3] Anciaux, N., Bonnet, P., Bouganim, L., Nguyen, B., Sandu Popa, I., and Pucheral, P. Trusted cells: A sea change for personal data services. In CIDR, 2013.

[4] Anciaux, N., Bouganim, L., Guo, Y., Pucheral, P., Vandewalle, J.-J., & Yin, S. Pluggable personal data servers. In ACM SIGMOD, demo. paper, 2010.

[5] Anciaux, N., Lallali, S., Sandu Popa, I. and Pucheral, P. A Scalable Search Engine for Mass Storage Smart Objects. PVLDB, 8(9), 2015.

[6] Bai, X., Guerraoui, R., Kermarrec, A.-M. and Leroy, V. 2011. Collaborative personalized top-k processing. ACM TODS, 36.

[7] Cao, P. and Wang, Z. Efficient top-k query calculation in distributed networks. In PODC, 2004.

[8] Kempe, D., Dobra, A. and Gehrke, J. Gossip-based computation of aggregate information. In FOCS, 2003.

[9] Kim, J. Y. and Croft, W. B. Retrieval Experiments using Pseudo-Desktop Collections. In CIKM, 2009.

[10] Kissner, L., Song, D. X. Privacy-Preserving Set Operations. In CRYPTO, 2005.

[11] Lallali, S., Anciaux, N., Sandu Popa, I., Pucheral, P. A secure search engine for the personal cloud. In ACM SIGMOD, demo. paper, 2015.

[12] Michel, S., Triantafillou, P. and Weikum, G. Klee: a framework for distributed top-k query algorithms. In VLDB, 2005.

[13] To, Q.-C., Nguyen, B., and Pucheral, P. Privacy-Preserving Query Execution using a Decentralized Architecture and Tamper Resistant Hardware. In EDBT, 2014.

[14] Wang, H., Tan, C. C., Li, Q. 2010. Snoogle: A search engine for pervasive environments. In Trans. on Par. & D. Sys.,21(8).

[15] Zobel, J. and Moffat, A. 2006. Inverted files for text search engines. In ACM Computing Surveys, 38(2).