Privacy-Preserving Remote Knowledge System

Markus Dahlmanns, Chris Dax, Roman Matzutt, Jan Pennekamp, Jens Hiller, Klaus Wehrle
Communication and Distributed Systems, RWTH Aachen University, Germany
Email: {dahlmanns, dax, matzutt, pennekamp, hiller, wehrle}@comsys.rwth-aachen.de

***Abstract*—More and more traditional services, such as malware detectors or collaboration services in industrial scenarios, move to the cloud. However, this behavior poses a risk for the privacy of clients since these services are able to generate profiles containing very sensitive information, e.g., vulnerability information or collaboration partners. Hence, a rising need for protocols that enable clients to obtain knowledge without revealing their requests exists. To address this issue, we propose a protocol that enables clients (i) to query large cloud-based knowledge systems in a privacy-preserving manner using Private Set Intersection and (ii) to subsequently obtain individual knowledge items without leaking the client's requests via few Oblivious Transfers. With our preliminary design, we allow clients to save a significant amount of time in comparison to performing Oblivious Transfers only.**

I. Introduction

The increased adoption of cloud-based services has led to an increased centralization of knowledge at specialized service providers for, e.g., search engines [1], malware detectors [2], or medical records [3]. A similar trend can be expected in the industrial sector due to the high demands for exchanging knowledge among industrial collaborators, e.g., cross-domain collaboration promises to increase productivity and save costs [4]. However, this reliance on external service providers introduces severe privacy risks: Attackers could gain knowledge of malware infestations of single devices, search engine providers generate extensive user profiles [1], and medical providers gain new tools to track individuals or discriminate against them [3]. Furthermore, industrial corporations need to maintain confidentiality of their business secrets even for increased levels of collaboration [4].

Thus, users of cloud-based knowledge systems need mechanisms that protect their privacy to seize the services' full potential without simultaneously exposing themselves to new risks. Addressing this demand, we present initial results of a protocol for privacy-preserving user queries to cloud-based knowledge systems. In our scenario, clients create a request set containing request items representing their interests. Correspondingly, the knowledge system maintains the database containing knowledge items mapping to offered information.

Our protocol enables clients to efficiently request individual knowledge items from large cloud-based knowledge systems using an efficient combination of Private Set Intersection (PSI) and Oblivious Transfers (OTs). In principle, OTs already allow for exchanging small pieces of data without leaking its contents by sending completely encrypted databases and obliviously disclosing only the keys for the requested items. However, their significant runtime overhead and resource consumption [5] renders them unfeasible to apply to large request sets. To this end, we propose to first filter client requests using PSI before selectively applying OTs on a subset to realize feasible, privacy-preserving user requests.

978-1-7281-2700-2/19/$31.00 2019 © IEEE

II. Related Work

Previous approaches to privacy-preserving database lookups typically involve an additional trusted third party [6], [7]. Hence, users need to trust both remote parties to not collude. We seek to change this current necessity and thereby increase the users' confidence in an adequate protection of their privacy.

In this vein, initial approaches such as BLOOM [3] already tackled the privacy issues associated with genomic data by obliviously offloading the processing. However, since our goal is to separate the requester from the knowledge item owner BLOOM is not applicable in our scenario.

III. Efficiently Requesting Knowledge in Private

To fill the gap not covered by related work, we propose an efficient privacy-preserving method for querying knowledge systems that enables clients to request knowledge items using a request set (i) without revealing anything about the requests nor (ii) learning additional knowledge items from the provider. Figure 1 illustrates a sequence chart of our envisioned protocol. To allow the client to privately retrieve information we rely on a combination of one-sided *Private Set Intersection (PSI)* and non-random *Oblivious Transfers (OTs)*. Our main idea is to reduce the number of expensive OTs required for obtaining requested knowledge items by first filtering the available knowledge using PSI. Thus, we enable the client to compute a set intersection of its request set and the knowledge database before it requests actual knowledge items using OTs. This approach promises to be especially efficient if the expected number of *relevant* knowledge items to be retrieved is small compared to the request set size.

The key requirement to seize this potential is that the client and the knowledge system use a suitable indexing scheme to support both PSI and OTs. In our design, the index of a knowledge item is based on a hash value derived from use-case specific attributes that are known to both, the client and the knowledge system, e.g., search behavior or malware. To this end, we use a one-time *preparation phase* in which the knowledge system computes the indices for all knowledge items using a predefined cryptographic hash function. Now, clients can establish a secure connection to the knowledge system and start the *initialization phase*, i.e., compute the

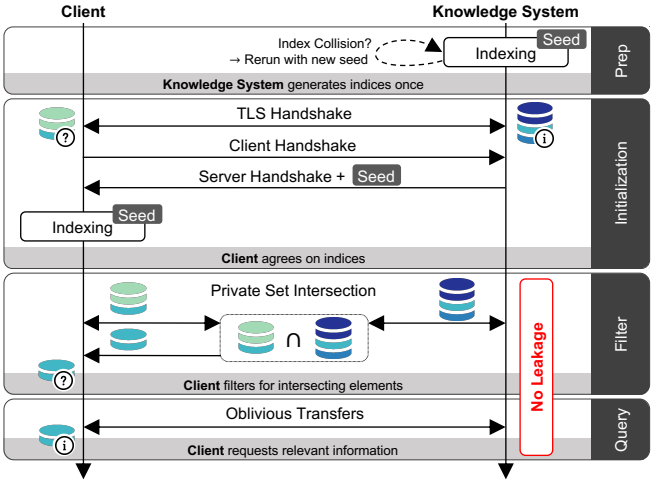

Fig. 1. The filter phase of our protocol reduces the client's request set size using Private Set Intersection. Hence, during the query phase the client needs to retrieve only relevant knowledge items via Oblivious Transfers. The preparation and initialization phases are necessary to generate required indices.

indices for their request set after an initial handshake. If the seed used for collision handling does not change, the client can reuse the calculated indices for future requests.

Afterward, in the *filter phase* the client starts performing a one-sided PSI with the knowledge system using these indices. Hence, only the client obtains the intersection of the index sets and leaks no information about the entries in her request set. Furthermore, the knowledge system does not provide more information than requested by the client since the client is only able to obtain the indices in the intersection.

In the final *query phase*, the client uses all indices contained in the intersection and performs one OT for each to retrieve the corresponding knowledge item from the knowledge system. The client then possesses the requested knowledge items.

IV. Preliminary Results

We implemented an initial prototype of our protocol in Python relying on C++ libraries for PSI [8] and OT [5] to show its potential. We deployed our client and server application both on the same machine equipped with two Intel Xeon E5-2630 @ 2.20 GHz and 32 GB RAM running Ubuntu 18.04.

Figure 2 visualizes the average time to completion of user requests and the client and knowledge system runtimes over 30 runs with their 99 % confidence intervals. Our knowledge system maintains a set of about 67 000 entries and the client's request set contains only matching entries, resulting in one OT each after the filter phase. Since the preparation phase needs to be performed once and the initialization phases can be omitted for subsequent requests, we focus on the filter and query phases. Furthermore, we consider request sets consisting of up to 100 items and observed that available memory appears as a bottleneck for our initial prototype for larger request sets. Our results show that the client can save up to 2 h (99 %) when no match occurs during the filter phase, i.e., the query phase after the PSI can be skipped, compared to performing the necessary 10 000 OTs when no filter is applied. For growing intersections the runtime increases nearly linearly since more OTs during

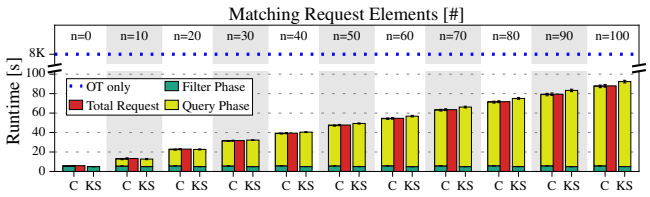

Fig. 2. Computation time on Client (C) and Knowledge System (KS) during the filter and query phases with different amount n of matching items. Furthermore, time needed by the client for the total request and runtime of performing OTs only for a request set size of 10 000 (blue, dotted line; calculated). Introducing the filter phase saves up to 99 % of time ($n = 0$).

the query phase are required for information requests. The additional overhead for the server shown in Figure 2 stems from a tear-down phase introduced by our implementation. In conclusion, our approach has a high potential of heavily unburdening the client in scenarios where low numbers of matches are expected, e.g., during regular vulnerability scans.

V. Conclusion and Future Work

Our early results promise significant advances for operating a privacy-preserving remote knowledge system. As future work we envision to further investigate the overheads for the client and the knowledge system, especially during the query phase. First simple optimizations indicate an inherent trade-off between privacy, performance, and memory consumption.

Furthermore, the amount of information transmitted through OTs in our implementation is currently limited to 16 byte. We thus plan to investigate an increase in information transferable in a single request. Moreover, instead of using OTs that return a single knowledge item from the knowledge set during the query phase performing OTs that return a variable number might be beneficial as this would reduce the total number of OTs. The evaluation of this has to be done in future as well.

Concluding, we showed the feasibility of querying large knowledge systems in a privacy-preserving manner by filtering request sets using Private Set Intersection before finalizing the request via potentially expensive Oblivious Transfers, which saves up to 99 % of time. Hence, our approach enables clients to retrieve information from cloud services privacy-preservingly and efficiently, but we seek to improve it further.

ACKNOWLEDGMENTS: The authors would like to thank the German Research Foundation (DFG) for the kind support within the Cluster of Excellence "Internet of Production" (IoP) under project ID 390621612.

References

[1] B. Schneier, *Data and Goliath: The hidden battles to collect your data and control your world*, 2015.
[2] Symantec Corporation, "Using Cloud-based Artificial Intelligence for Enterprise-Focused Advanced Threat Protection," Tech. Rep., 2018.
[3] J. H. Ziegeldorf *et al.*, "BLOOM: BLoom filter based Oblivious Outsourced Matchings," *BMC Medical Genomics*, 2017.
[4] J. Pennekamp *et al.*, "Towards an Infrastructure Enabling the Internet of Production," in *IEEE ICPS 2019*.
[5] M. Orrù, E. Orsini, and P. Scholl, "Actively Secure 1-out-of-N OT Extension with Application to Private Set Intersection," in *CT-RSA 2017*.
[6] E. De Cristofaro, Y. Lu, and G. Tsudik, "Efficient Techniques for Privacy-Preserving Sharing of Sensitive Information," in *Trust 2011*.
[7] D. Boneh *et al.*, "Private Database Queries Using Somewhat Homomorphic Encryption," in *ACNS 2013*.
[8] P. Rindal and M. Rosulek, "Malicious-Secure Private Set Intersection via Dual Execution," in *ACM CCS 2017*.