From Private Set Intersection to Secure Database Joins using Cross-Layer Design

Long Gu
Systems Group, TU Darmstadt, Germany

Zsolt István
Systems Group, TU Darmstadt, Germany

1 MOTIVATION AND PROBLEM STATEMENT

The adoption of the General Data Protection Regulation (GDPR) and other similar frameworks protect end-users against indiscriminate data collection and processing, but they make data sharing across different data owners challenging. Shared data, however, can have great utility – as shown by applications in machine learning, finance, and big data analysis. One approach to achieve sharing while protecting the privacy of end-users is to use Secure Multi-Party Computation (SMPC), a technique that enables several parties to jointly compute a function without disclosing private information.

Recent related works show that SMPC can be succesfully applied to the domain of databases but they have also highlighted that SMPC operations are typically orders of magnitude more compute-intensive than their plain-text counterparts: for instance, the Secure Join in SMCQL [1], although it has been already improved by secure query optimization, still requires almost two minutes to handle 15GB of data, which is unpractical in many real-world scenarios. Finding a balance between privacy guarantees, runtime scalability, and absolute performance is the main challenge when developing a database with a privacy-protected data sharing features. Our work is motivated by this challenge and proposes one way to address it – we consider this an important goal, given that in the future, the significance of privacy-preserving databases will only grow.

2 PROPOSED SOLUTION

Our goal is to investigate how to build an efficient privacy-preserving database engine using a Cross-Layer design approach. As opposed to many related work from the database community, that considers the SMPC operations as "black boxes", and many related works from the cryptography community that considers the database operators as a "fixed design", in our Cross-Layer approach we will optimize across layers, assuring that any database primitives are matched to the most efficient cryptographic primitive without trading off on security/privacy guarantees. As the poster explains, our first step is to evaluate a number of candidate SMPC protocols. We can establish the design of balanced privacy-preserving database primitives by dissecting and examining the properties of relavant SMPC protocols. In addition, in order to further improve the performance of database operators, it is also one of our goals to remove hardware-level inefficiencies of the chosen cryptographic primitives. We will achieve this by making execution hardware-aware or, if possible, by using hardware accelerators – this way the Cross-Layer design will encompass the hardware layer as well.

3 ONGOING AND FUTURE WORK

As the first step, we investigate how joins, one of the most important database operations, can be implemented with privacy-preserving guarantees across two parties in the most efficient way. Joins have many similarities with Private Set Intersection (PSI) in SMPC, and could be implemented using them. PSI protocols, such as the recently proposed KKRT16 [5], are relatively high performance (our micro-benchmarks show that the intersection of millions of data points can be computed in 4.9 seconds). A core operation in KKRT16 is to obliviously evaluate a pseudo-random function (OPRF) between two parties. The fundamental cryptographic primitive of OPRF is Oblivious Transfer (OT), which is a technique that enables the sender to deliver one of multiple messages to the receiver, based on the receiver's input, without knowing which has been delivered. We implemented and tested a "simplest OT" [3]. In order to improve the efficiency of OT, we also tested the benefits of Beaver Derandomization [2]. But to solve the problem of private set intersection of large datasets, an effective online OT protocol alone is insufficient. So, we also tested the IKNP03 [4] protocol, which borrows ideas from KKRT16.

In our first steps, we deconstructed the key steps of a PSI protocol and we constructed a secure join. Our early results demonstrate that the use of OT-based secure join is feasible: the performance of secure join is not greatly constrained by OT as a cryptographic primitive. In the online phase, using the OT Derandmization scheme only costs 0.005 seconds for 1 million OTs. Moreover, because of IKNP03 [4], the number of OTs to be computed can be decoupled from the data set size. The largest overhead results from the fact that the pseudo-random function is computed jointly by the two parties in order to compare if the components are equal without disclosing the actual elements.

In addition to simple OT, we also implemented Private Equality Test (PET) based on OT-based PSI [6], which allows us to compute the result of secure join on certain attributes in a very efficient way. This is needed because we must take into account comparisons between specific values and data columns in addition to secure join across datasets. And the result demonstrates that such a direct conversion is possible for a single-condition query, but because the length of the input data affects how many OT executions are needed, the traffic will increase in line with the length of the data.

In the future we will explore other SMPC protocols, including Fully Homomorphic Encryption (FHE) and Secret-Sharing scheme (SSS), with the goal of contrasting their performance to OT and studying which is easier to integrate with a proper query processing engine and which one is more efficient on modern hardware.

REFERENCES

[1] Johes Bater, Gregory Elliott, Craig Eggen, Satyender Goel, Abel Kho, and Jennie Rogers. 2016. SMCQL: Secure Querying for Federated Databases. (2016).

[2] Donald Beaver. 1995. Precomputing Oblivious Transfer.

[3] Tung Chou and Claudio Orlandi. 2015. The Simplest Protocol for Oblivious Transfer.

[4] Yuval Ishai, Joe Kilian, Kobbi Nissim, and Erez Petrank. 2003. Extending Oblivious Transfers Efficiently. In *Annual International Cryptology Conference*.

[5] Vladimir Kolesnikov et al. 2016. Efficient Batched Oblivious PRF with Applications to Private Set Intersection.

[6] Benny Pinkas, Thomas Schneider, and Michael Zohner. 2014. Faster Private Set Intersection Based on OT Extension *(SEC'14)*. USENIX Association, USA.