Mathematical Problems of Computer Science 45, 59--66, 2016.

Secure Multiparty Computations for Collaboration between Competing Service Providers

Davit H. Danoyan

Yerevan State University
e-mail: danoyan@gmail.com

Abstract

We introduce a platform for secure computations and describe the platform workflow. We detail the Golreich-Micalli-Widgerson protocol employed for secure multiparty computation in our platform with optimization techniques applied to it. Being based on white-box cryptography, the underlying oblivious transfer protocol avoids the use of expensive public key operations and provides good overhead compared to similar systems. Also we point out some useful application scenarios and provide a real world application design based on our platform.

Keywords: Secure multiparty computation, Oblivious transfer, White-box cryptography

1. Introduction

Secure computations were introduced by Yao in [19] searching a solution to the Millionaires problem, where two millionaires want to find out who is the richest of them without revealing to each other or any other party of their actual property. Years later several approaches were developed for the solution of this problem and its generalized version with n millionaires as Yao's garbled circuit based protocol [12], GMW [18] and others. Although those protocols were valid solutions for the problem, the computational resources available at that time were insufficient for using them in practice. The first practical implementation of a secure computation protocol was presented in [1] nearly a decade ago. Later, several enhancements and new implementations have been developed [2].

Many applications were proposed for online auctions [5], e-commerce [7], data mining [6], secure computations outsourcing [4], etc. In this paper we present a new application scenario, where several service providers possessing specific private information wish to make secure computations on their collective data.

In section 2 we will present some background on the GMW multiparty computation protocol and our optimization techniques. Section 3 will be the brief description of our platform used for the secure computations in the specific application. The applications' motivation and description can be found in section 4 and section 5 is the conclusion of this paper.

2. GMW Protocol

The GMW protocol is intended for computation of any function that is possible to represent as a single pass Boolean circuit, where the permitted operations (circuit gates) are XOR and AND. As the protocol avoids the need of trusted third parties, communication is required between all pairs of participants.

Unlike the Yao's garbled circuits protocol, where one party is the circuit generator and the other party is the circuit evaluator, here all the parties possess the circuit locally and each of them plays an evaluator with input shares of himself and the other participants.

2.1 Secret Sharing

Each party generates n shares for each input wire w $(s_{w1}, \dots, s_{wn})$ randomly so, that $\bigoplus_{i=1}^{n} s_{wi} = s_w$. Share generation doesn't really require much effort to satisfy the above equation. Party P_j generates n-1 shares randomly for all other parties $P_i, i \neq j$. To comply with the equation, it just has to adjust its own share correspondingly - $s_{wj} = (\bigoplus_{\mathrm{i \neq j}} s_{wi}) \oplus s_w$.

After the share generation, each share s_{wi} is sent to the corresponding participant P_i. Now, having shares for all input wires, each participant proceeds to circuit evaluation. After the evaluation, each participant P_i would have its own value s_{wi}, so for each output wire w of the circuit we again will have an n-tuple $(s_{w1}, \dots, s_{wn})$. To combine those values into a single value for each wire the protocol suggests a specific way of gate evaluation, according to which the n-tuples for each output wire will play as shares of parties, and the final value will be computed with $s_w = \bigoplus_{i=1}^{n} s_{wi}$ formula. As the circuit consists of XOR and AND gates only, below we provide the methods for both gates' evaluation that should be used for all input, intermediate and output gates of the circuit and show that they operate as expected.

2.2 Gate Evaluation

Evaluation of XOR gates.

Suppose we have an XOR gate with input wires u and v and an output wire w. All parties already have their shares for the input wires $(s_{u1}, \dots, s_{un})$ and $(s_{v1}, \dots, s_{vn})$. Each party P_i computes the value of the output wire in the following way: $s_{wi} = s_{ui} \oplus s_{vi}$. The n-tuple $(s_{w1}, \dots, s_{wn})$ would be a valid sharing for w's value, because

$$s_w = \bigoplus_{i=1}^{n} s_{wi} = \bigoplus_{i=1}^{n} (s_{ui} \oplus s_{vi}) = (\bigoplus_{i=1}^{n} s_{ui}) \oplus (\bigoplus_{i=1}^{n} s_{vi}) = s_u \oplus s_v.$$

Evaluation of AND gates

Suppose we have an AND gate with input wires u and v and an output wire w. The shares of input values of u and v are $(s_{u1}, \dots, s_{un})$ and $(s_{v1}, \dots, s_{vn})$. The output value s_w should be computed so, that

$$\begin{aligned} s_w = s_u \wedge s_v = (\bigoplus_{i=1}^{n} s_{ui}) \wedge (\bigoplus_{i=1}^{n} s_{vi}) = \\ = (\bigoplus_{i=1}^{n} (s_{ui} \wedge s_{vi}))\ \oplus (\bigoplus_{i<j} ((s_{ui} \wedge s_{vj}) \oplus (s_{uj} \wedge s_{vi}))). \end{aligned}$$

The first sum $\bigoplus_{i=1}^{n} (s_{ui} \wedge s_{vi})$ can be computed locally for every participant, but for evaluation of the second sum one should communicate with other participants. To avoid the collection of individual shares of other parties, oblivious transfer protocols are used for AND gate evaluation in the following way.

As P_i has to interact with several P_j-s to compute each element of

$$\bigoplus_{i<j}\left(\left(s_{ui} \wedge s_{vj}\right) \oplus \left(s_{uj} \wedge s_{vi}\right)\right),$$

P_j generates a random value $r_j^{\{i,j\}}$ and composes four values with $r_j^{\{i,j\}}$ and its shares for input wires of the gate being computed s_{uj} and s_{vj}.

$$r_j^{\{i,j\}}, \qquad r_j^{\{i,j\}} \oplus s_{uj}, \qquad r_j^{\{i,j\}} \oplus s_{vj}, \qquad r_j^{\{i,j\}} \oplus s_{uj} \oplus s_{vj}.$$

Then the parties invoke a $1-out-of-4$ oblivious transfer with P_i as receiver and P_j as sender. P_i requests one of the abovementioned values with its index composed of its shares for the input wires s_{ui}, s_{vi}. P_i computes the required sum with the received value $r_i^{\{i,j\}}$

$$r_i^{\{i,j\}} \oplus r_j^{\{i,j\}} = (s_{ui} \wedge s_{vj}) \oplus (s_{uj} \wedge s_{vi}).$$

The proof of correctness of the AND gates' computation method described above can be found in section 7.5.2.2 of [3].

So P_i, after getting all $r_j^{\{i,j\}}$-s from all parties with $j > i$, can compute its share for the output wire w.

$$s_{wi} = (s_{ui} \wedge s_{vi}) \oplus (\bigoplus_{i<j}\left(\left(s_{ui} \wedge s_{vj}\right) \oplus \left(s_{uj} \wedge s_{vi}\right)\right).$$

Given the methods of share based evaluation of both type gates, each party computes the circuits' output value shares. The computation of the full value of an output wire can be done after each participant sends the result of its computations to others.

Note, that computation of an XOR gate share values is free in terms of network communication and almost free, in terms of computation (n XOR operation). However, the computation of AND gates takes $\binom{n}{2}$ OT invocations.

2.3 Optimizations

White-Box Oblivious Transfer

Oblivious transfer is an essential cryptographic primitive heavily used in secure computations. It was introduced by Rabin in [13] as a protocol where the sender sends a message to the receiver with $\frac{1}{2}$ probability and does not reveal if the receiver got the message or not. A modified version of this protocol was introduced later by Even et al. in [14], currently known as the basic 1-out-of-2 OT protocol. This protocol assumes that the receiver has a selection bit $s\epsilon\{0,1\}$ and the sender has two bits $m_0, m_1\epsilon\{0,1\}$. After protocol execution the receiver gets m_s and nothing about m_{1-s} and the sender does not reveal s.

Unfortunately this protocol depends on public-key operations, which require exponentiation operations which are pretty expensive, considering the huge amount of OT operations required for secure computations (in Yao's protocol OTs are used for all input bits of one of the participants, in GMW OTs are used for all AND gates). In [11] a new approach to OT was introduced using white-box cryptography techniques instead of public-key operations, allowing to reduce cost of an OT in several orders of magnitude, in case of many invocations, as reported.

WBOT extension

Another optimization technique for reducing the OT invocation count is OT extension introduced by Beaver in [17] and later enhanced in [16]. We have constructed similar optimization for white-box OT, allowing to reduce the required OT invocation count to a predetermined fixed number. The details of the extension can be found in [8].

2.4 GMW Protocol Extension

Here we introduce an extended version of the GMW protocol, that enables involvement of so called passive parties that don't participate in the computation process, while having input and expecting output. Suppose $n + m$ parties wish to compute a common function based on private inputs, only m of which are passive. In this scenario we suggest the active (participating in computation) parties to behave as they do in the original protocol and share their inputs among n active parties with the method described in section 2.2. Each passive participant also generates n shares for each of its input, but doesn't keep a share for himself and distributes all shares among the active parties. The latter compute the function based on the secret shares as in the original protocol and distribute output shares among all parties. The security requirement for this protocol are also different – here we require at least one non-corrupted active party.

3. Platform Description

The platform processes a file containing a program described in an imperative language for to be computed by the participants. Beside the function description the file contains specification on the intended inputs and expected outputs. The program defines computations in a manner they would be carried out by a virtual trusted party possessing all input parameters. Our platform consist of three main modules – a compiler, and two modules responsible for two-party and multiparty secure computations respectively. Below we briefly describe the main modules.

Compiler

We implemented a compiler generating an equivalent Boolean circuit on basis $\{AND, XOR, 1\}$ implementing the described function. The compiler generates an initial Boolean circuit and passes it through various stages of optimizations trying to minimize the number of gates in generated circuit. Another major goal of the compiler is the minimization of AND gates and circuit depth. For this purpose we use low depth circuit constructions developed mainly for VLSI applications.

To deal with large circuits efficiently, the compiler generates a file containing the usage count for each gate. When a gate is processed, by a participant, the initial usage count is read from this file, and each time the output of the gate is accessed, the associated counter is decremented. Data associated with the gate is removed from memory once the counter hits zero. In this way we minimize the number of gates simultaneously kept in memory during evaluation of the circuit.

Two-Party Module

Although it is possible to perform secure computations with two participants using GMW protocol, in our platform we have a specialized module for this task. For this purpose we have implemented Yao's garbled circuits protocol [12]. The module works on the output of the compiler module. Detailed description can be found in [9].

Multiparty Module

The multiparty module is intended for secure computations with more than 2 participants. The module implements the Goldreich-Micalli-Widgerson protocol[18] with several enhancements described in [15] and in the previous section. It also operates on a Boolean circuit outputted by the compiler module.

4. Applications

Here we will describe the process of combining multiple delivery services into a unified platform that involves several members. In fact, the application described is suitable for using in different contexts like combining any delivery or transportation services where the members are service providers, who do not want to disclose their current locations or the locations they are available to cover within the fixed time to their competitors, but eager to cooperate with them to have their share in unified ordering system, thus increasing their order counts and service efficiency.

Concerning the motivation of client to use the unified system, let's compare the actions needed for getting the fastest service. In case of having an access to the unified application, the client just has to give the application her location and confirm the order. All the computation and decision processes are completed without the clients' involvement. In absence of such a system one should contact several service providers (possibly with different interfaces), give them its location, compare the offers of different providers, find the best of them and finally put an order. In the latter case the client also has to trust the information it got from service providers.

Suppose the service providers are taxi service companies with one or more cabs and the client has no priorities for choosing particular service other than fast pick-up. Taxi services do not want to make their locations visible to rival companies to avoid them getting advantage by better positioning of their cabs (for example, this can be done by a company with significantly more cabs). All cab locations are also hidden from the client, because a competitor can possibly act as a client to find out the others' cab locations and get advantage.

Having the application scenario described we show several methods as candidates for the secure computation itself. This methods are basically computing the minimum value of distances between a client as one point, and provider instances as candidates for the second point. Different distance measurement algorithms are presented below.

The main purpose of this application is to find the nearest item in the unified database of locations to a submitted location. With the computation securing method already appointed, we still need to have accurate location detecting and network connection hardware for the service providers and individual cabs. In this work we consider the mentioned hardware implemented in a black-box manner – we do not consider their technical and efficiency details. We also assume that the computing parties have identical maps as well and their locations follow common coordinate system, to avoid misinterpretation of function arguments.

Considering the distance measurement we have several methods as a candidate.

Euclid Distance

The most general version of the distance measurement function is computation of the Euclid distance between the customer and all the provider instances (e.g. cabs). With the coordinates given in two dimensional format, the customer's location denoted as (x_c, y_c) and for i-th provider instance (x_i, y_i) considering m provider instances, the following function should be computed to determine the "winning" bid:

$$\min_{\mathrm{i<m}}((x_c - x_i)^2 + (y_c - y_i)^2).$$

Although, this distance measurement method will be very precise in case of services operating in seas and oceans or drone deliveries, its accuracy can be very poor in case of taxi or delivery services in most cities street maps or for similar problems in any grid-like structures.

Manhattan Distance

This distance measurement algorithm is also called a taxicab distance which intuitively fills the accuracy gap mentioned for Euclid distance measurement. The distance in this method is the sum of absolute differences of the Cartesian coordinates. In this case the desired function is

$$\min_{\mathrm{i<m}}(|x_c - x_i| + |y_c - y_i|).$$

This algorithm is very simple and has a small circuit representation and, therefore, is efficient in terms of performance, but unfortunately it cannot be considered as a general purpose method.

Graph Approach

Another method for fixed traffic map but with rather irregular structure can be introduced with graph construction for specific maps, where grid-oriented Manhattan approach is not effective. This technique assumes the parties have once preprocessed the map they are going to cooperate upon and generated a graph-map for it. The generated graph should be oriented and weighted. The graph vertices are crossroads on the map and an edge (A, B) is present in the graph between vertices A and B, if there's a direct route without crossroads (not involving any other vertex) connecting those on the real map with traffic allowed in $A \to B$ direction. The weights are non-negative numbers assigned to the graph edges, equal to the distance between the corresponding crossroads.

Upon our function computation we can also add special marked vertices for the client and provider instances or simply mark their nearest nodes as special, depending on the map specifics. With this graph construction our desired function to compute the nearest provider instance will be with use of Dijkstra's algorithm [10] for finding the shortest path in graphs. The algorithm will terminate upon finding any node marked as special.

5. Conclusion

In this paper we described the multiparty computations workflow on our platform intended for secure computations. The GMW protocol was presented, along with its extension, some optimizations and a brief description of the underlying white-box oblivious transfer protocol. Delivery and transportation services were considered in respect of applying secure computations in the field and description of application and several distance measurement were considered.

As a future work we plan to enhance the security of the platform, which is currently secure in the semi-honest model. Also there are possible enhancements in application specific manner, such as reducing the computation on client side.

References

[1] D. Malkhi, N. Nisan, B. Pinkas and Y. Sella, "Fairplay - Secure Two-Party Computation System," Proceedings of the 13th USENIX Security Symposium, vol. 4, 2004.

[2] Y. Huang, D. Evans, J. Katz and L. Malka, "Faster Secure Two-Party Computation Using Garbled Circuits", USENIX Security Symposium, vol. 201, no. 1. 2011.

[3] O. Goldreich, Foundations of cryptography: volume 2, basic applications. Cambridge university press, 2009.

[4] H. Carter, B. Mood, P. Traynor and K. Butler, "Secure outsourced garbled circuit evaluation for mobile devices", *Proceedings of the USENIX Security Symposium*, pp. 1-44, 2013.

[5] P. Bogetoft, I. Damgård, T. Jakobsen, K. Nielsen, J. Pagter and T. Toft, "A practical implementtation of secure auctions based on multiparty integer computation", *Financial Cryptography and Data Security, vol. 4107 of LNCS*, pp 142-147, Springer, 2006.

[6] Y. Lindell and B. Pinkas "Secure multiparty computation for privacy-preserving data mining", *Journal of Privacy and Confidentiality 1*, no. 1., pp. 5, 2009.

[7] S. Choi, K. W. Hwang, J. Katz, T. Malkin and D. Rubenstein, "Secure multi-party computation of boolean circuits with applications to privacy in on-line marketplaces", *Topics in Cryptology–CT-RSA 2012*, Springer Berlin Heidelberg, pp. 416-432, 2012.

[8] D. Danoyan, "Extending white-box cryptography based oblivious transfer protocol", *Proceedings of the Yerevan State University, Physical and Mathematical Sciences no. 1*, pp. 40-44, 2016

[9] D. Danoyan and T. Sokhakyan, "A generic framework for secure computations", *Proceedings of Russian-Armenian (Slavonic) University 2015 (Physical, mathematical and natural sciences)*, vol. 2, pp. 14-21, 2015.

[10] E. W. Dijkstra, "A note on two problems in connexion with graphs", *Numerische mathematik 1*, no. 1, pp. 269-271, 1959.

[11] A. Jivanyan and G. Khachatryan, "Efficient oblivious transfer protocols based on white-box cryptography", *AUA Internal reports*, 2013.

[12] A. Yao, "How to Generate and exchange secrets", *In 27th FOCS*, pp. 162-167, 1986.

[13] M. Rabin, "How to exchange secrets by oblivious transfer", *Tech. Memo TR-81, Aiken Computation Laboratory*, Harvard University, 1981.

[14] S. Even, O. Goldreich and A. Lempel, "A randomized protocol for signing contracts", *Communications of the ACM*, vol. 28, pp. 1985.

[15] T. Schneider and M. Zohner, "GMW vs. Yao? Efficient secure two-party computation with low depth circuits", *Financial Cryptography and Data Security*, pp. 275-292. Springer Berlin Heidelberg, 2013.

[16] Y. Ishai, J. Kilian, K. Nissim and E. Petrank, "Extending oblivious transfers efficiently", *CRYPTO 2003, Springer-Verlag (LNCS 2729)*, pp. 145-161, 2003.

[17] D. Beaver, "Correlated pseudorandomness and the complexity of private computations", *STOC 1996*, pp. 479-488, 1996.

[18] O. Goldreich, S. Micali and A. Wigderson, "How to play any mental game, or a completeness theorem for protocols with honest majority", *19th Annual ACM Symposium on Theory of Computing (STOC)*, pp. 218-229, 1987.

[19] A. Yao, "Protocols for secure computations", *Foundations of Computer Science, 1982. SFCS'08. 23rd Annual Symposium on 1982*, IEEE, pp. 160-164, 1982.

Submitted 04.11.2015, accepted 25.01.2016

Գաղտնի բազմամասնակից հաշվարկներ մրցակցող ծառայություն մատուցողների համագործակցության համար

Դ. Դանոյան

Ամփոփում

Այս հոդվածում ներկայացված է բազմամասնակից անվտանգ հաշվարկների համակարգ և նկարագրված է նրա աշխատանքի սկզբունքները: Համակարգի իրականացման համար օգտագործվել է Golreich-Micalli-Widgerson (GMW) հաղորդակարգը որոշ լավարկումներով: Քանի որ օգտագործվող անտեղյակ փոխանցման հաղորդակարգը հիմնված է գաղտնագրման սպիտակ արկղի եղանակի վրա, որտեղ չեն օգտագործվում բաց բանալիով գաղտնագրման թանկարժեք գործողությունները, մեր համակարգը, նմանատիպ այլ համակարգերի համեմատ ավելի արագագործ է: Հոդվածում նաև նկարագրվում է տվյալ համակարգի վրա հիմնված կիրառության օրինակ:

Безопасные многосторонние вычисления для сотрудничества конкурирующих провайдеров услуг

Д. Даноян

Аннотация

В этой статье представлена платформа для безопасных многосторонних вычислений и описан процесс его работы. Для реализации вычислительной платформы был использован оптимизированный протокол Golreich-Micalli-Widgerson (GMW). Поскольку лежащий на основе протокол забывчивой передачи разработан на новых криптографических принципах, не использующих дорогостоящие операции с открытым ключом, наша система обеспечивает хорошую производительность по сравнению с подобными системами. Вместе с этим, приводится разработка приложения основанного на предложенной платформе.