RAVI EKAMBARAM
rekambaram@albany.edu

FAST AND SECURE THREE PARTY COMUTATION: THE GARBLED CIRCUIT APPROACH

ABSTRACT

There are many approaches based on multi-party computation that permits single corrupt party because of the high efficiency outcome of the protocol. An innovative approach was proposed for Three-Party computation using the garbled circuit to deal with single, malicious party and for increasing security. The protocol used constant number of rounds, hash functions, “Pseudorandom Generators”. [1] Basically, the approach doesn’t use expensive and public key operations protocols like oblivious transfer which was used in Two-Party Computation. The efficiency of the above approach is comparable with the existing system. Various libraries were used during the implementation for the garbling scheme. Experiments were performed using benchmarking circuits. Techniques such as hash functions were used to reduce the communication size and cost.

MOTIVATION

There are various methods of communication, garbling circuit is one amongst them. After reading the paper and all about the garbling circuit approach, the main thing that caught our attention was the way they transferred data and keys using the garbling scheme and the non-interactive commitment instead of the oblivious transfer. Also, they have implemented various techniques for reducing the communication cost and size, they were able to achieve the comparable performance efficiency to the existing information Three party protocol.

BACKGROUND MATERIAL

After reading the paper, there were a lot of key words which were hard to understand like garble circuit itself, Pseudorandom Generators, “Common random strings” [2], “random tape” [3], garbling scheme, non- commitment interaction.

RAVI EKAMBARAM
rekambaram@albany.edu

First, we started gathering information about the key topics via YouTube, Wikipedia, Reading other reference papers.

SCOPE

The garbled circuit approach is used for both Two-party and Three-Party Computation, we can extend this approach for multi-party computation using different techniques whose performance is comparable to the existing ones. The major advantage of three-party computation technique is the function used to encrypt is never revealed. This would help us achieve major advancements in the field of public-key cryptography.

LIMITATIONS

This approach was introduced to provide security against single, malicious party, but the approach would fail if both the parties are corrupted instead of one. Another limitation would be the method the server analyses the authenticity of the clients. Server should go through a series of trials to authenticate the client before a connection is being established which was a time-consuming process.

MAIN BODY

The paper is about 3PC (Three- party Computation) using the garbled circuit. The paper is taking reference from the previous protocol which was proposed by Yao. The "Yao's protocol" [4] had 2 Party Computation where in the sender who encrypts is known as Garbler and the receiver who decrypts is called Evaluator. They are using the oblivion transfer which is a public key operation to send the input. The garbled circuit was used in this approach which is a cryptographic protocol which is used by 2 parties to communicate with each other even when they are mistrusting and can make a function with their private inputs and evaluate it. The 3PC approach in the paper is also using the garbled circuit approach. In this approach, there are 3 parties out of which 2 parties are the garbler and 1 is the evaluator. This approach was proposed so that the communication can be protected from single malicious party. The oblivious transfer is not used because it is very expensive and public operation. They are

RAVI EKAMBARAM
rekambaram@albany.edu

using garbling scheme and a non-interactive commitment scheme. All the 3 parties have their own private inputs. The evaluator will generate a common random string and divide his input into 2 parts, one part is sent to first garbler and second part to another garbler. Either one of the garbler will generate a Pseudo-Random function and share it with another garbler. Both garblers will then use the following information (The Common Random String, Inputs and the Pseudo-Random Function) will create the garble inputs using the garble circuits and send those created inputs to the evaluator. The evaluator will check whether the inputs obtained are the ones which he sent and conclude whether the inputs are corrupted or not and decide whether to establish the connection or not. “In the implementation, they have used JustGarble library, MsgPack, open ssl libraries. “The evaluator will read the description from the file. The garblers will connect to each other to negotiate a shared seed and use it to generate garbled circuit.” [5] Multiple Gates are used to reduce the communication. Hash values are used by garblers, both garblers will send half hashed values to the evaluator and he will put the hashed values together to construct the garbled circuit and check the equality.

In our implementation, first as we have three party computation out which 2 parties are the garbler and one party is the evaluator, we have used socket programming to connect the garblers with the evaluator. Every party will have its own private input. The keys which are used are randomly generated. Both the garblers will have different keys. The clients will generate a garble table individually and send the table as well as the key used to generate the tables to the evaluator. The server choosing either from 0 or 1 will generate a garble circuit using the keys shared by the respective clients. The garbled values which are generated by the server(Evaluator) will be compared to the garble values send by the clients and if they match the connection will be established. We are also encrypting the garbled values using AES algorithm.

CONCLUSIONS

We implemented the Three-party computation using the garbled circuit using some benchmarking circuits and observed that the performance is comparable to Yao’s protocol and its efficiency is comparable to multi party computation techniques without the garbled circuits.

RAVI EKAMBARAM
rekambaram@albany.edu

REFERENCES

[1] https://en.wikipedia.org/wiki/Pseudorandom_generator
[2] https://en.wikipedia.org/wiki/Common_reference_string_model
[3] https://crypto.stackexchange.com/questions/35339/what-is-the-random-tape
[4] https://en.wikipedia.org/wiki/Garbled_circuit
[5]https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/43888.pdf