Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 213 (2022) 523–527

Procedia Computer Science

www.elsevier.com/locate/procedia

2022 Annual International Conference on Brain-Inspired Cognitive Architectures for Artificial Intelligence: The 13th Annual Meeting of the BICA Society

Secure multi-party computations for privacy-preserving machine learning

Sergey Zapechnikov

Institute of Cyber Intelligence Systems, National Research Nuclear University (Moscow Engineering Physics Institute), Kashirskoye shosse 31, Moscow, 115409, Russia

Abstract

The paper is devoted to the analysis of privacy-preserving machine learning (PPML) systems based on secure multi-party computations. It reviews PPML systems, analyses the goals and objectives of its application. A generalized model of PPML architecture is proposed, reflecting the main functional blocks of such systems. The formulation of the problem of secure multi-party computation is considered. The descriptions of cryptographic primitives and protocols used to implement multi-party secure computation protocols, including garbled circuits, secret sharing schemes, and homomorphic encryption are given. The current PPML systems based on two-, three-, and four-party secure computations are analyzed. The main attention is paid to algorithmic aspects of systems, methods and protocols of securing information. Systems secure against semi-honest and active adversaries are considered, both based on universal modules for secure multi-party computations, and specialized ones designed to ensure the privacy of specific machine learning technologies, such as convolutional neural networks. We consider examples of implemented prototypes of several PPML systems. Based on the results of the analysis, conclusions are formulated about the features of the future PPML systems.

Keywords: machine learning; privacy; secure multi-party computations; secret sharing scheme; homomorphic encryption.

1. Introduction

One of the most significant trends in modern computer science and information technology is the rapid development of artificial intelligence (AI) technologies and systems. There is a lot of talk about AI forming the core of a new technological order. Like any complex of new technologies, it goes from theoretical development, experiments and prototypes to widespread implementation in many areas of business activity. One of the important

10.1016/j.procs.2022.11.100

criteria for the acceptance of new technologies by society is, of course, trust in them. One of the important aspects of trust in information technologies is information security.

AI technologies are a conglomerate of a wide variety of methods and algorithms: logical conclusions, search, optimal control, etc. However, no one doubts that the core of AI technologies are machine learning methods and models, primarily deep learning. In this regard, the mechanisms of ensuring the information security of machine learning systems are of primary interest.

2. Privacy-preserving machine learning

The PPML task can be defined as ensuring the privacy of each participant of a machine learning system in conditions when the persons providing the training sample at the stage of training the model or providing test sample at the stage of inference (we will call them clients), remotely interact with the owner of the model, who is able to perform computations using this model (we will call it the server). At the same time, clients are interested in non-disclosure of their data (training samples, requests to model, and responses to them) both to each other and to the owner of the model. At the same time, the owner of the model is interested in non-disclosure of the parameters of his model to the clients.

The specific situations in which such an interest arises may differ: for example, when processing his personal data, as well as data constituting medical, tax or banking secrecy. The interest of the model owner may arise when providing paid functionalities (for example, pattern search [1], face recognition, image processing, recommendations, OLAP techniques [2]) using the model. Such services have received the generalized name MLaaS – Machine Learning as a Service. In the future, sets of such functionalities may become platforms for various personalised information services [3].

In general, privacy includes such components as confidentiality, access control, anonymity, non-connectivity of actions or events, indistinguishability of the initiator of the event and a number of others, depending on the specific situation. Modern technologies and tools provide many ways to learn more information about a person than she desires: direct data leaks, recovery of information by indirect channels, data mining (identification of subtle and hidden patterns in data) and many others. In the broadest sense of the word, privacy implies protection from an overcurious adversary.

The configuration in which PPML is carried out can be represented by an architectural model proposed on Fig. 1.

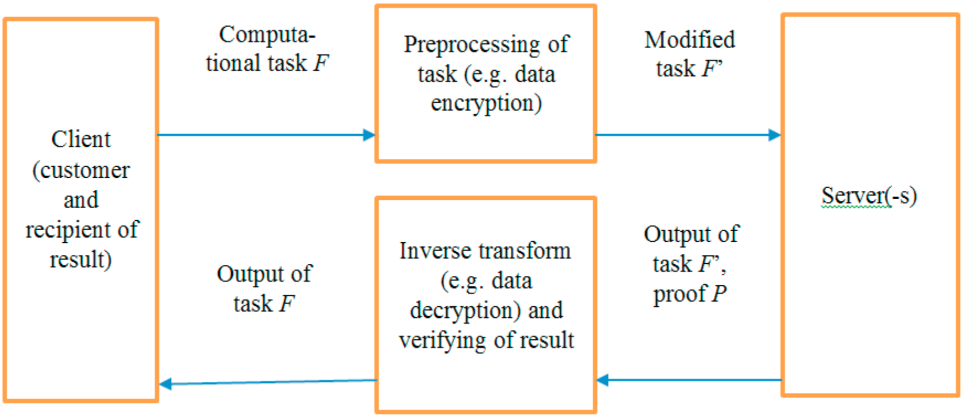
Client (customer and recipient of result)
Computa-tional task *F*
Preprocessing of task (e.g. data encryption)
Modified task *F*'
Server(-s)
Output of task *F*
Inverse transform (e.g. data decryption) and verifying of result
Output of task *F*', proof *P*

Fig. 1. A generalized model of PPML system architecture.

The model includes three main functional blocks:

- A block for generating a task outsourced and obtaining the result of solving the task;
- A block of task transformation and verification of the task solving result;

- A task solving block.

The physical correspondence of the functional blocks to the participants of the computational process can be ambiguous. The most preferable scheme is when the first two blocks are located on a trusted client system, the third block is on an untrusted server side, which can be represented by one or more physically dedicated servers or virtual machines. However, this configuration can lead to a high computational load on the client. The second possible configuration is to place the first block on a trusted client system, the second block on a trusted gateway system, and the third block on untrusted servers. This model allows to conveniently placing all the basic functionality that ensures the security of information in blocks, but is potentially associated with a large number of vulnerabilities. The server side can be represented by a single physical computer or a cluster of interconnected devices, which in this case must perform multi-party secure computations among themselves.

The task preparation blocks, as well as the reverse transformation and verification of the solution, may include different functionality depending on the method of solving the problem. For example, when using homomorphic encryption, the first of these blocks encrypts the input data, the second – decrypts the result of solving the problem. In the case of using secure multi–party computations based on secret sharing, the first block will divide input data into shares to subsequently forward them to the servers; the second block collects the result of solving the task from the shares returned by the servers. In some cases, verification of the correctness of the solution of the problem can be added to the inverse transformation, for example, by means of zero-knowledge proofs attached by servers to their solution shares. As an intermediate case, the client possibly may participate in the computations along with servers (if it has sufficient computing resources). This is typical for protocols based on secret sharing schemes.

The following functionality is required from untrusted components of the model:

- Execute data processing algorithms without disclosure of the plaintext of the data;
- Generate an evidence of correct execution of the required data processing algorithms.

The following functionality is required from the trusted components of the model:

- Prepare the input data of the task to be solved for transmission to untrusted components;
- Receive the results of solving the task from untrusted components and converting them into a format suitable for the customer;
- Verify the correctness of the task solution by an untrusted component (if necessary).

3. Secure multi-party computations

Recall the definition of secure multi-party computations [4]. A multi-party cryptographic protocol is being considered, in which each of the participants has its own individual secret. It is necessary to evaluate some function of these secrets, so that the result of the calculations is known to all members of the group, but the secrets themselves were not disclosed by the protocol participants to each other or to any third party. Formally: let the participants of the cryptographic protocol $P_1, P_2, \dots, P_n$ have confidential input data $x_1, x_2, \dots, x_n$, respectively. As a result of the protocol execution, they should jointly calculate a function of the form $y = f(x_1, x_2, \dots, x_n)$ when the following properties are met:

- Correctness: each of the participants $P_1, P_2, \dots, P_n$receives *y*;
- Privacy: no additional information is disclosed to any of the participants and outsiders, except for what they knew before the protocol was executed, including the intermediate results of calculations.

There are four main algorithmic tools to implement SMPC:

- Garbled circuits;
- Secret sharing schemes;
- Homomorphic encryption;

- Zero-knowledge proofs (in some cases for verification of task output).

4. PPML based on secure two-party computations

A special case of SMPC is secure two-party computations. Let the participants of the cryptographic protocol P_1, P_2 have private input data x_1, x_2, respectively. As a result of the protocol execution, they should jointly calculate a function of the form $y = f(x_1, x_2)$ when performing similar correctness and privacy properties.

Well-known examples of two-party PPML systems are ABY and ABY 2.0 modules. The ABY module [5] is a software backend component, based on the idea of combining arithmetic, Boolean and garbled circuits. To do this, three forms of secret sharing are used: arithmetic, Boolean and Yao accordingly. This is necessary in order to perform each operation on the shared secrets in the fastest way. For each form of secret sharing, its own semantics is used, so that its own set of standard operations and its own optimization techniques are available when executing protocols.

The ABY 2.0 module [6] is a significantly upgraded version of the ABY module, which made it possible to significantly speed up the multiplying of shared secrets and converting between different forms of their representation. Optimizations concern mainly the arithmetic form of representation. The Boolean form is treated as a special case of arithmetic one for the $\mathbb{Z}_2$ ring, and the Yao form has remained unchanged.

5. PPML based on secure three-party computations

An example of three-party PPML system is ABY^3 module [7]. It is conceived and implemented as a virtual processor performing a set of basic operations for three-party computations on integers. The general ideas of the ABY^3 module look similar to the ideas underlying the ABY module [5]: arithmetic, Boolean and Yao sharing. The appropriate schemes are used in computations. Switching between them is necessary for selection the most productive computation protocol. Among the participants of the protocol, no more than one adversary is allowed. However cryptographic protocols are very different due to the fact that the secret is shared among three parties here.

6. PPML based on secure two-party computations

The main motivation for increasing the number of participants in the SMPC protocols is the desire to achieve higher system performance and, if possible, provide users with stronger security guarantees for their data.

Four-party computations make it possible to implement distributed computing procedures that are resistant to stronger adversary models than two-party and three-party ones. Protocols that provide security against an active adversary are of primary interest. Such security is provided by two key properties of the protocol, known as fairness and robustness [4]. Let's give their informal definitions.

Fairness is a guarantee that at the end of the protocol either all participants receive the same output data, or none of the participants receives any result, i.e. the protocol will be interrupted. There is a weakened definition of fairness, which allows the adversary to interrupt the participation of honest participants, but not allows the adversary to get any additional information.

Robustness is a guarantee that honest participants will bring the protocol to the end and get the correct result when active adversary participates in protocol, even if she completely interrupts her participation in the protocol.

One example of a four-party PPML system with advanced functionality that most fully implements the properties of security against an active adversary is Tetrad [8].

7. Conclusions

In the course of the work, a search study was carried out and a review of PPML systems based on two-, three- and four-party secure computations was carried out. An informal formulation of the problem is highlighted and a generalized model of PPML systems is proposed.

One of the main algorithmic tool for secure computations in PPML systems are secret sharing schemes. In PPML systems, three types of secret sharing are used: arithmetic, Boolean and Yao sharing, each of which allows performing secure calculations with shared secrets according to the corresponding type of schemes: arithmetic, Boolean or garbled circuits. There is a tendency to gradually abandon garbled circuits in favour of the increasingly widespread use of arithmetic and Boolean schemes. The scope of application of garbled circuits remains mainly complicated nonlinear functions. The most likely reason for this is the great complexity of designing and implementing such protocols.

The analysis of SMPC-based PPML systems allows identifying two leading lines of their development:

- Systems with a core based on universal software modules with a set of basic operations that allow building a secure computations protocol that implements arbitrary functionality limited only by the complexity of the protocol;
- Specialized PPML with a set of cryptographic primitives optimized to achieve high throughput and performance for certain machine learning methods or adversary models.

Acknowledgements

This work was supported by the Ministry of Science and Higher Education of the Russian Federation (state assignment project No. 0723-2020-0036).

References

[1] Zapechnikov S. “Contemporary Trends in Privacy-Preserving Data Pattern Recognition”. In: *Procedia Computer Science* **190**: 838-844 (2021).

[2] Zapechnikov S. “Privacy-Preserving Machine Learning as a Tool for Secure Personalized Information Services”. In: *Procedia Computer Science* **169**: 393-399 (2020).

[3] Gorlatykh A., Zapechnikov S. “Challenges of privacy-preserving OLAP techniques”. (2017) Proceedings of the 2017 IEEE Russia Section Young Researchers in Electrical and Electronic Engineering Conference, ElConRus 2017, pp. 404-408.

[4] Evans D., Kolesnikov V., Rosulek M. “A pragmatic introduction to secure multi-party computation”. 182 pp. URL: https://securecomputation.org/docs/pragmaticmpc.pdf (accessed: 04.09.2022).

[5] Demmler, D. (2015) “ABY – a framework for efficient mixed-protocol secure two-party computation”. D. Demmler, T. Schneider, M. Zohner. 22nd Network and Distributed System Security Symposium (NDSS'15), Internet Society, San Diego, CA, USA, February 8-11, 2015. URL: https://encrypto.de/papers/DSZ15.pdf (accessed: 04.09.2022).

[6] Patra A., Schneieder T., Suresh A. et al. “ABY2.0: Improved mixed-protocol secure two-party computation”. URL: https://ia.cr/2020/1225 (accessed: 04.09.2022)

[7] Mohassel, P. “ABY^3: A mixed protocol framework for machine learning”. Cryptology ePrint Archive. URL: https://eprint.iacr.org/2018/403 (accessed: 04.09.2022).

[8] Nishat K., Arpita P., Rahul R., Ajith S. “Tetrad: Actively Secure 4PC for Secure Training and Inference”. URL: https://eprint.iacr.org/2021/755.pdf (accessed: 04.09.2022).